

Кафедра математики та інформатики
Matematika és Informatika Tanszék

ВЕБ-ПРОГРАМУВАННЯ / WEBPROGRAMOZÁS

МЕТОДИЧНІ ВКАЗІВКИ / MÓDSZERTANI ÚTMUTATÓ
до самостійної роботи / az önálló munkához

Перший (бакалаврський) / Alapképzés (BSc)
(ступінь вищої освіти / felsőoktatás szintje)

A Освіта / A Oktatás
(галузь знань / képzési ág)

A4 Середня Освіта / A4 Középfokú oktatás
(спеціальність / képzési program)

Інформатика / Informatika
(освітня програма / képzési program)



Веб-програмування: методичні рекомендації для самостійної роботи для здобувачів першого (бакалаврського) рівня вищої освіти галузі знань А "Освіта" спеціальності А4.09 «Середня освіта (Інформатика)» денної та заочної форми навчання / укл. Доровці А.Ф. Берегове: Закарпатський угорський університет ім. Ференца Ракоці II, 2025. 80 с.

Затверджено до використання у навчальному процесі
на засіданні кафедри математики та інформатики Університету Ракоці
(протокол № 3 від «15» жовтня 2025 року)

Укладачі:

Адам ДОРОВЦІ – доктор філософії, доцент кафедри математики та інформатики
Закарпатського угорського університету імені Ференца Ракоці II.

Навчально-методичне видання розроблене з метою забезпечення здобувачів вищої освіти методичними рекомендаціями щодо виконання самостійних робіт з дисципліни "Веб-програмування". Методичні рекомендації покликані допомогти здобувачам опанувати освітню компоненту та містять завдання для контрольних робіт. Методичні рекомендації розраховані на здобувачів денної та заочної форми навчання першого (бакалаврського) рівня вищої освіти, що навчаються за спеціальністю А4.09 «Середня освіта (Інформатика)». Методичні рекомендації включають тематики до самостійних робіт, перелік рекомендованих джерел та тестових завдань для самоаналізу.

Webprogramozás: módszertani ajánlások önálló munkához az első (alapképzés) szintű felsőoktatási hallgatók számára az A „Oktatás” tudományterület A4.09 „Középfokú oktatás (Informatika)” szak nappali és levelező tagozataihoz / összeállította: Daróci Á. Beregszász: II. Rákóczi Ferenc Kárpátaljai Magyar Egyetem, 2025. 80 oldal.

Az oktatási folyamatban történő felhasználását jóváhagyta
a II. Rákóczi Ferenc Kárpátaljai Magyar Egyetem Matematika és Informatika Tanszéke
(2025. október 15., 3. számú jegyzőkönyv).

A módszertani útmutatót összeállították:
DARÓCI Ádám – PhD, a II. Rákóczi Ferenc Kárpátaljai Magyar Egyetem Matematika és Informatika Tanszékének docense.

A kiadvány célja, hogy módszertani ajánlásokat nyújtson az „Webprogramozás” tantárgyból végzendő önálló munkákhoz. Az útmutató segít a hallgatóknak elsajátítani az adott tantárgyi komponens anyagát, és ellenőrző feladatokat is tartalmaz. A módszertani ajánlások a nappali és levelező tagozatos képzésben részt vevő hallgatóknak készültek, akik az „A Oktatás” tudományterület A4.09 „Középfokú oktatás (informatika)” szakján tanulnak alapképzésben. Az útmutató tartalmazza a különböző önálló munkák témaköreit, az ajánlott források jegyzékét, továbbá önellenőrzésre szolgáló tesztfeladatokat.

ЗМІСТ

ТЕМАТИЧНИЙ ПЛАН ДИСЦИПЛІНИ	5
ТЕМАТИКА ДИСЦИПЛІНИ:.....	6
МЕТОДИЧНІ ВКАЗІВКИ ДО САМОСТІЙНОГО ВИВЧЕННЯ ТЕМ ТА ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ	11
ТЕСТИ ДЛЯ ПРОВЕДЕННЯ САМОКОНТРОЛЮ	37
РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ.....	78

TARTALOM

A TANTÁRGY TEMATIKUS ÁTTEKINTÉSE.....	8
A TANTÁRGY TEMATIKÁJA	9
MÓDSZERTANI ÚTMUTATÓ A TÉMÁK ÖNÁLLÓ TANULMÁNYOZÁSÁHOZ ÉS ÖNELLENŐRZŐ KÉRDÉSEK.....	11
TESZTFELADATOK ÖNELLENŐRZÉSHEZ	61
AJÁNLOTT SZAKIRODALOM	78

ТЕМАТИЧНИЙ ПЛАН ДИСЦИПЛІНИ

«Веб-програмування»

Анотація

Програма вивчення навчальної дисципліни «Веб-програмування» складена відповідно до навчального плану підготовки бакалаврів напряму (спеціальності) «А4 Середня освіта. Інформатика». Предметом навчальної дисципліни «Веб-програмування» є процес і результат засвоєння студентами практичних навичок створення web-сайтів різного рівня складності, основних принципів та прийомів програмування в JavaScript, PHP та MySQL для використання отриманих ґрунтовних знань в подальшій професійній діяльності.

Мета:

Надати студентам систематизовані знання про мету, завдання та принципи мови гіперрозмітки HTML, таблиці каскадних стилів CSS, фреймворку Bootstrap, мов програмування PHP та JavaScript, JS-бібліотеки jQuery, JS-фреймворків Angular та ReactJS, технологій XML, AJAX, а саме знання про синтаксис мов та технологій, об'єктно-орієнтовані принципи, вміння працювати з рядками, текстовими файлами, базами даних, вміння створювати HTML-форми, обробляти їх та отримувати дані з них PHP- та JS-скриптами, після чого записувати дані в базу даних.

Завдання:

вивчення теоретичних відомостей, набуття студентами практичних навичок використання сучасних засобів інформаційних технологій для створення новітніх гіпермедійних систем та опанування фундаментальних знань з сучасних веб-технологій та основ веб-програмування.

Дисципліна формує такі компетентності за ОПП:

ФК18 Здатність до цифрового подання та обробки текстової, числової, графічної, звукової та відеоінформації.

Програмні результати навчання:

ПРН12 Визначає структуру предметної галузі інформатики, її місце в системі наук, пояснює перспективи розвитку інформатики та інформаційних технологій, їхнє суспільне значення.

ПРН17 Будувати інформаційну модель, реалізовувати її засобами цифрових технологій; проводити комп'ютерний експеримент, інтерпретувати, аналізувати та узагальнювати його результати.

ПРН22 Демонструвати знання з основних розділів математики та інформатики.

ПРН23 Розв'язувати задачі з інформатики різного рівня складності раціональними методами.

ПРН53 Застосовувати різні підходи до розв'язання проблем, генерувати нові ідеї, демонструвати відкритість до ідей та рішень учасників освітнього процесу.

ТЕМАТИКА ДИСЦИПЛІНИ:

Модуль 1. Мова розмітки даних HTML та каскадні таблиці стилів CSS

Змістовий модуль 1. Мова розмітки даних HTML

Тема 1. Принципи роботи Веб. Процес побудови веб-сторінок.
Налаштування середовища розробки для роботи з HTML.

Тема 2. Вступ до HTML. Базова структура веб-сторінки. Теги HTML та їхні властивості.

Тема 3. Основні елементи HTML5 на рівні блоків та на рівні тексту.

Тема 4. Веб-форми в документі HTML.

Тема 5. XML та XHTML.

Змістовий модуль 2. Каскадні таблиці стилів CSS

Тема 6. Вступ до використання веб-стилів. Форматування тексту, шрифту, кольору та фону.

Тема 7. Блокова модель CSS. Візуальна модель форматування.

Тема 8. Відображення елементів. Плаваючі елементи. Псевдо класи та елементи.

Тема 9. Елементи CSS3. FlexBox. Медіа запити. Сімейство рекомендацій для визначення перетворень і представлення XML: XSL.

Тема 10. Бібліотека стилів Bootstrap.

Модуль 2. Програмування на стороні клієнта

Змістовий модуль 3. Програмування на стороні клієнта

Тема 11. Основи мови JavaScript.

Тема 12. Використання JavaScript в HTML.

Тема 13. Обробка подій на веб-сторінці.

Тема 14. HTML DOM. Створення та видалення елементів HTML на JavaScript.

Тема 15. Основи деяких популярних фреймворків JavaScript: ReactJS та AngularJS.

Тема 16. Асинхронний JavaScript. AJAX.

Тема 17. Елементи jQuery.

Модуль 3. Програмування на стороні сервера

Змістовий модуль 4. Програмування на стороні сервера

Тема 18. Основи мови PHP.

Тема 19. Обробка веб-форм на мові PHP. Сесії. Ключі.

Тема 20. Асинхронний PHP – AJAX. Робота з XML в PHP.

Модуль 4. Робота з базами даних для веб MySQL

Змістовий модуль 5. Робота з базами даних для веб MySQL

Тема 21. Основи роботи з базами даних MySQL. Налаштування середовища MySQL. phpMyAdmin.

Тема 22. Основні команди роботи з базами MySQL. Агрегатні функції. Об'єднання.

Тема 23. Робота з базами даних із мови програмування PHP.

Модуль 5. Вебдоступність та інклюзивні технології

Змістовий модуль 6. Розробка доступних вебінтерфейсів

Тема 24. Поняття доступності та універсального дизайну. Групи користувачів з особливими потребами та їхні проблеми.

Тема 25. Стандарт WCAG 2.0 та основні принципи вебдоступності.

Тема 26. Спеціальні інструменти та технології для забезпечення доступності.

Тема 27. WAI-ARIA: покращення доступності вебзастосунків.

Тема 28. Методи перевірки доступності.

A TANTÁRGY TEMATIKUS ÁTTEKINTÉSE

„Webprogramozás”

Annotáció

A „Programozás (2. modul: Webprogramozás)” tantárgy a „014 Középfokú oktatás. Informatika” alapszak képzési tervéhez készült. A tantárgy célja, hogy a hallgatók elsajátítsák a különböző nehézségi szintű weboldalak készítésének gyakorlati alapjait, valamint megismerkedjenek a JavaScript, PHP és MySQL programozási nyelvek legfontosabb elveivel és módszereivel, amelyeket később a szakmai életben is hasznosíthatnak.

Cél:

A hallgatók megismerik a HTML jelölőnyelvet, a CSS stíluslapok, a Bootstrap keretrendszer, a PHP és JavaScript programozási nyelvek, a jQuery könyvtár, az Angular és ReactJS keretrendszerek, valamint az XML és AJAX technológiák alapjait. Megtanulják a nyelvek szintaxisát, az objektumorientált gondolkodás alapjait, valamint azt, hogyan kell szövegekkel, fájlokkal és adatbázisokkal dolgozni. Gyakorlati feladatokon keresztül sajátítják el HTML űrlapok készítését, azok feldolgozását PHP- és JS-szkriptekkel, és az adatok adatbázisba mentését.

Feladatok:

- alapvető elméleti ismeretek elsajátítása,
- gyakorlati készségek fejlesztése korszerű informatikai eszközökkel,
- egyszerű és összetettebb webes alkalmazások készítése,
- alapvető webprogramozási technológiák megismerése és használata.

Szakmai kompetenciák:

SKZ18 – Képes különböző típusú információkat — szöveget, számadatot, képet, hangot és videót — digitális formában megjeleníteni és feldolgozni.

Tervezett tanulási eredmények:

TTE12 – Meghatározza az informatika tudományterületének szerkezetét, annak helyét a tudományok rendszerében, magyarázza az informatika és az információs technológiák fejlődési kilátásait, társadalmi jelentőségét.

TTE17 – Képes információs modellt létrehozni és digitális technológiák segítségével megvalósítani; számítógépes kísérletet végezni, valamint annak eredményeit értelmezni, elemezni és összefoglalni.

TTE22 – Bemutatja a matematika és informatika alapvető területein szerzett ismereteit.

TTE23 – Megold különböző nehézségű informatikai feladatokat racionális módszerekkel.

TTE53 – Különböző megközelítéseket alkalmaz problémamegoldásra, új ötleteket generál, nyitottságot mutat az oktatási folyamat résztvevőinek ötletei és megoldásai iránt.

A TANTÁRGY TEMATIKÁJA

Modul 1.: HTML jelölőnyelv és a CSS

Tartalmi modul 1.: HTML jelölőnyelv

Téma 1. A web alapelvei. A weboldalak felépítésének folyamata. A HTML-el való munkához szükséges fejlesztőkörnyezet beállítása.

Téma 2. Bevezetés a HTML-be. Egy weboldal alapvető felépítése. HTML jelölők és tulajdonságaik.

Téma 3. A HTML5 alapvető elemei blokk- és szövegszinten.

Téma 4. Webes űrlapok egy HTML-dokumentumban.

Téma 5. XML és XHTML.

Tartalmi modul 2. CSS

Téma 6. Bevezetés a webes stílusok használatába. Szöveg, betűtípus, szín és háttér formázása.

Téma 7. A CSS blokkmodell. A formázás vizuális modellje.

Téma 8. Az elemek megjelenítése. Lebegő elemek. Pszeudo osztályok és elemek.

Téma 9. CSS3 elemek. FlexBox. Media.

Téma 10. Bootstrap stíluskönyvtár.

Modul 2. Kliensoldali programozás

Tartalmi modul 3. Kliensoldali programozás

Téma 11. JavaScript alapok.

Téma 12. JavaScript használata a HTML-ben.

Téma 13. Események kezelése egy weboldalon.

Téma 14. HTML DOM. HTML elemek létrehozása és törlése JavaScriptben.

Téma 15. Néhány népszerű JavaScript keretrendszer alapjai: ReactJS és AngularJS.

Téma 16. Aszinkron JavaScript. AJAX.

Téma 17. A jQuery elemei.

Modul 3. Szerveroldali programozás

Tartalmi modul 4. Szerveroldali programozás

Téma 18. PHP alapjai.

Téma 19. Webes űrlapok feldolgozása PHP-ben. Session.

Téma 20. Aszinkron PHP - AJAX. XML-el való munka PHP-ben.

Modul 4. Munka webes adatbázisokkal MySQL

Tartalmi modul 5. Adatbázisokkal való munka a webhez MySQL

Téma 21. A MySQL adatbázisokkal való munka alapjai. A MySQL környezet beállítása. phpMyAdmin.

Téma 22. A MySQL adatbázisokkal való munka alapvető parancsai. Összesítő függvények.

Téma 23. Adatbázisokkal való munka a PHP programozási nyelven.

Modul 5. Webes akadálymentesítés és inkluzív technológiák

Tartalmi modul 6. Hozzáférhető webes felületek fejlesztése

Téma 24. Az akadálymentesítés és az egyetemes tervezés fogalma. A speciális igényű felhasználók csoportjai és problémái.

Téma 25. WCAG 2.0 szabvány és a webes akadálymentesség alapelvei

Téma 26. Speciális eszközök és technológiák az akadálymentesítéshez

Téma 27. WAI-ARIA: Webes alkalmazások elérhetőségének javítása

Téma 28. Akadálymentesség ellenőrzésének módszere

МЕТОДИЧНІ ВКАЗІВКИ ДО САМОСТІЙНОГО ВИВЧЕННЯ ТЕМ ТА ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ / MÓDSZERTANI ÚTMUTATÓ A TÉMÁK ÖNÁLLÓ TANULMÁNYOZÁSÁHOZ ÉS ÖNELLENŐRZŐ KÉRDÉSEK

**Модуль 1. Мова розмітки даних HTML та каскадні таблиці стилів CSS /
Modul 1. HTML jelölőnyelv és a CSS**

**Змістовий модуль 1. Мова розмітки даних HTML / Tartalmi modul 1.
HTML jelölőnyelv**

**Тема 1. Принципи роботи Веб. Процес побудови веб-сторінок.
Налаштування середовища розробки для роботи з HTML. / A web
alapelvei. A weboldalak felépítésének folyamata. A HTML-el való munkához
szükséges fejlesztőkörnyezet beállítása.**

1. Опишіть повний життєвий цикл HTTP-запиту, починаючи від введення URL-адреси в браузері до відображення веб-сторінки. Яку роль у цьому процесі відіграють клієнт, сервер та система доменних імен (DNS)? / Írja le egy HTTP-kérés teljes életciklusát a böngészőben beírt URL-től a weboldal megjelenítéséig. Milyen szerepet játszik ebben a kliens, a szerver és a domainnév-rendszer (DNS)?
2. Назвіть та охарактеризуйте основні етапи процесу створення веб-сайту. / Sorolja fel és jellemezze a weboldal-készítés fő szakaszait.
3. Порівняйте щонайменше три сучасні текстові редактори коду. Які ключові функції роблять їх ефективними для веб-розробки? / Hasonlítsa össze legalább három modern kódszerkesztőt. Mely fő funkciók teszik őket hatékonyá a webfejlesztésben?
4. Що таке інструменти розробника в браузері (Developer Tools)? Опишіть призначення щонайменше трьох основних вкладок. / Mit jelentenek a böngésző fejlesztői eszközei? Mutassa be legalább három fő panel funkcióját.
5. Поясніть, як правильно організувати структуру файлів та папок для простого веб-проєкту. Чому важливо використовувати відносні шляхи до файлів (наприклад, для зображень та стилів) та дотримуватися єдиного стилю іменування файлів? / Magyarázza el, hogyan kell helyesen megszervezni a fájl- és mappastruktúrát egy egyszerű webprojektben. Miért fontos relatív útvonalakat használni (például képekhez és stíluslapokhoz), és egységes névadási konvenciót követni?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).

- Курінний С. Розробка веб-сайтів для початківців: HTML, CSS, JavaScript. 2022. (українською).
- Svekis L. Front-End Web Development with Modern HTML, CSS, and JavaScript. Packt Publishing, 2022. (англійською).
- Nagy G. Web programozás alapismeretek. Online tankönyv. 2011. (угорською).

Тема 2. Вступ до HTML. Базова структура веб-сторінки. Теги HTML та їхні властивості. / Bevezetés a HTML-be. Egy weboldal alapvető felépítése. HTML jelölők és tulajdonságaik.

1. Поясніть призначення декларації `<!DOCTYPE html>`. Чому вона є обов'язковою на початку кожного HTML-документа і що станеться, якщо її пропустити? / Magyarázza el a `<!DOCTYPE html>` deklaráció célját. Miért kötelező minden HTML-dokumentum elején, és mi történik, ha kihagyjuk?
2. Опишіть базову структуру HTML-документа, включаючи теги `<html>`, `<head>` та `<body>`. Який тип інформації розміщується в кожному з цих розділів? / Írja le egy HTML-dokumentum alapvető szerkezetét, beleértve a `<html>`, `<head>` és `<body>` tageket. Milyen információkat tartalmaz mindegyik rész?
3. Що таке HTML-елемент, тег та атрибут? Наведіть приклад елемента з кількома атрибутами та поясніть призначення кожного з них. / Mit jelent a HTML-elem, tag és attribútum? Hozzon példát olyan elemre, amelynek több attribútuma van, és magyarázza el mindegyik célját.
4. Назвіть щонайменше три ключові теги, що розміщуються всередині `<head>`, та поясніть їхню функцію. / Soroljon fel legalább három fontos taget, amelyet a `<head>` részben használunk, és ismertesse a funkciójukat.
5. Що таке глобальні атрибути в HTML? Наведіть приклади трьох глобальних атрибутів та опишіть, як вони використовуються. / Mit jelentenek a globális attribútumok a HTML-ben? Hozzon három példát globális attribútumokra, és írja le, hogyan használják őket.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівецьк. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Meloni J. C., Kyrnin J. HTML, CSS, and JavaScript All in One: Covering HTML5, CSS3, and ES6, Sams Teach Yourself. Sams Publishing, 2018. (англійською).
- Svekis L. HTML, CSS, and JavaScript for Beginners: A Web Design Course. Packt Publishing, 2021. (англійською).
- Infojegyzet.hu. Webszerkesztés: HTML alapok. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 3. Основні елементи HTML5 на рівні блоків та на рівні тексту. / A HTML5 alapvető elemei blokk- és szövegszinten.

1. Поясніть різницю між блоковими (block-level) та рядковими (inline-level) елементами в HTML. Наведіть по два приклади кожного типу та опишіть їхню поведінку на сторінці. / Magyarázza el a blokkszintű és soron belüli (inline) HTML-elemek közötti különbséget. Hozzon két-két példát mindkét típusra, és írja le viselkedésüket az oldalon.
2. Що таке семантична верстка? Порівняйте використання несемантичного тегу <div> та семантичних тегів HTML5 (<article>, <section>, <nav>, <header>). Чому семантика важлива для доступності (accessibility) та пошукової оптимізації (SEO)? / Mit jelent a szemantikus felépítés? Hasonlítsa össze a nem szemantikus <div> tag és a szemantikus HTML5 tagek (<article>, <section>, <nav>, <header>) használatát. Miért fontos a szemantika az akadálymentesség és a keresőoptimalizálás szempontjából?
3. Опишіть призначення та правильну структуру для створення нумерованих (), маркованих () та вкладених списків в HTML. / Ismertesse a számozott (), felsorolásos () és beágyazott listák célját és helyes struktúráját HTML-ben.
4. Які атрибути є обов'язковими, а які рекомендованими для тегу ? Поясніть важливість атрибута alt з точки зору веб-доступності. / Mely attribútumok kötelezőek, és melyek ajánlottak az tag esetén? Magyarázza el az alt attribútum fontosságát a webakadálymentesség szempontjából.
5. Як створити гіперпосилання на іншу сторінку того ж сайту, на зовнішній ресурс, на певну частину поточної сторінки (якір) та для відправки електронного листа? / Hogyan hozható létre hivatkozás ugyanazon oldal másik részére, külső forrásra, az aktuális oldal egy adott részére (horgony), illetve e-mail küldésére?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівец. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Svekis L. Front-End Web Development with Modern HTML, CSS, and JavaScript. Packt Publishing, 2022. (англійською).
- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- Infojegyzet.hu. HTML-CSS feladatok. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 4. Веб-форми в документі HTML. / Webes űrlapok egy HTML-dokumentumban.

1. Опишіть призначення тегу `<form>` та його ключових атрибутів `action` та `method`. Поясніть фундаментальну різницю між методами GET та POST для надсилання даних форми. / Ismertesse a `<form>` tag célját és fő attribútumait, az `action` és `method` paramétereit. Magyarázza el az alapvető különbséget a GET és POST módszerek között űrlapadatok küldésekor.
2. Перелічіть щонайменше п'ять різних типів елемента `<input>` та опишіть їхнє призначення та вигляд. / Soroljon fel legalább öt különböző `<input>` típust, és ismertesse rendeltetésüket és megjelenésüket.
3. Яка роль тегу `<label>` у веб-формах? Поясніть два способи асоціювати `<label>` з відповідним полем вводу та чому це критично важливо для доступності. / Mi a `<label>` tag szerepe a webűrlapokban? Ismertessen két módszert a `<label>` hozzárendelésére a megfelelő beviteli mezőhöz, és magyarázza el, miért kulcsfontosságú ez a hozzáférhetőség szempontjából.
4. Порівняйте елементи `<input type="submit">`, `<input type="button">` та `<button>`. У яких ситуаціях доцільно використовувати кожен з них? / Hasonlítsa össze az `<input type="submit">`, `<input type="button">` és `<button>` elemeket. Milyen esetekben célszerű az egyiket vagy a másikat használni?
5. Як реалізувати базову валідацію на стороні клієнта, використовуючи лише атрибути HTML5 (наприклад, `required`, `pattern`, `min`, `max`)? / Hogyan valósítható meg alapvető kliensoldali validáció kizárólag HTML5 attribútumok (például `required`, `pattern`, `min`, `max`) használatával?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Svekis L. HTML, CSS, and JavaScript for Beginners: A Web Design Course. Packt Publishing, 2021. (англійською).
- Infojegyzet.hu. HTML-CSS jegyzet: HTML űrlap elemek. Online tananyagok. Frissítve: 2024. (угорською).
- MDN Web Docs. HTML forms. Mozilla Foundation. Оновлюється постійно. (англійською).

Тема 5. XML та XHTML. / XML és XHTML.

1. Що таке XML (eXtensible Markup Language)? Поясніть його основне призначення та ключові відмінності від HTML, зокрема щодо синтаксису та гнучкості тегів. / Mi az XML (eXtensible Markup Language)? Mutassa be alapvető célját és fő különbségeit a HTML-hez képest, különösen a szintaxis és a tagok rugalmassága terén.
2. Наведіть приклад простого XML-документа для опису структури книги (наприклад, назва, автор, розділи). Які основні правила (наприклад, наявність кореневого елемента, закриття всіх тегів) потрібно виконати, щоб документ був синтаксично коректним? / Készítsen példát egy egyszerű XML-

dokumentumra, amely egy könyv struktúráját írja le (például cím, szerző, fejezetek). Milyen alapvető szabályokat (például gyökérelem megléte, minden tag lezárása) kell betartani ahhoz, hogy a dokumentum szintaktikailag helyes legyen?

3. Що таке XHTML (eXtensible HyperText Markup Language)? Поясніть, як він пов'язаний з HTML та XML, та назвіть три основні синтаксичні правила XHTML, які є суворішими, ніж у HTML4. / Mi az XHTML (eXtensible HyperText Markup Language)? Ismertesse kapcsolatát a HTML-lel és XML-lel, és soroljon fel három szigorúbb szintaktikai szabályt, amelyek az XHTML-ben érvényesek a HTML4-hez képest.

4. Обґрунтуйте, чому веб-спільнота перейшла від розробки XHTML 2.0 до створення HTML5. Які переваги HTML5 призвели до його домінування? / Indokolja meg, miért állt át a webes közösség az XHTML 2.0 fejlesztéséről a HTML5 kidolgozására. Milyen előnyök tették a HTML5-öt dominánssá?

5. Хоча XHTML більше не є стандартом для веб-сторінок, де технологія XML продовжує активно використовуватися в сучасних веб-технологіях? Наведіть приклади (наприклад, RSS-каналы, AJAX, конфігураційні файли). / Bár az XHTML már nem szabvány a weboldalak számára, hol használják továbbra is aktívan az XML technológiát a modern webfejlesztésben? Hozzon példákat (például RSS-csatornák, AJAX, konfigurációs fájlok).

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- W3C (World Wide Web Consortium). Extensible Markup Language (XML) Specification. Офіційна документація. (англійською).
- Infojegyzet.hu. PHP jegyzet: REST API és JSON. Online tananyagok. Frissítve: 2024. (угорською).

Змістовий модуль 2. Каскадні таблиці стилів CSS / Tartalmi modul 2. CSS

Тема 6. Вступ до використання веб-стилів. Форматування тексту, шрифту, кольору та фону. / Bevezetés a webes stílusok használatába. Szöveg, betűtípus, szín és háttér formázása.

1. Порівняйте три способи підключення CSS до HTML-документа: вбудований (inline), внутрішній (internal) та зовнішній (external). Опишіть переваги та недоліки кожного підходу та вкажіть, який є найкращою практикою для реальних проєктів. / Hasonlítsa össze a CSS három beillesztési módját HTML-dokumentumban: inline, belső és külső stíluslap. Ismertesse mindegyik előnyeit és hátrányait, és jelölje meg, melyik a legjobb gyakorlat valós projektekben.

2. Поясніть поняття "каскадності" та "специфічності" в CSS. Як браузер вирішує, яке правило застосувати, якщо до одного елемента застосовано кілька конфліктуючих стилів? / Magyarázza el a CSS „kaszkádolás” és „specificitás” fogalmait. Hogyan dönti el a böngésző, hogy melyik szabály érvényes egy elemre, ha több, egymással ütköző stílus vonatkozik rá?
3. Опишіть основні типи селекторів CSS: селектор тегу, селектор класу та селектор ID. Наведіть приклади синтаксису для кожного та поясніть, коли доцільно використовувати кожен з них. / Ismertesse a CSS szelektorok főbb típusait: elem-, osztály- és ID-szelektort. Adjon szintaktikai példát mindegyikre, és magyarázza el, mikor célszerű használni őket.
4. Назвіть та продемонструйте використання щонайменше п'яти CSS-властивостей для форматування тексту. / Soroljon fel és mutasson be legalább öt CSS-tulajdonságot a szöveg formázásához.
5. Які існують моделі представлення кольору в CSS? Опишіть синтаксис та основні відмінності між HEX, RGB та RGBA. / Milyen színmegjelenítési modellek léteznek a CSS-ben? Ismertesse a HEX, RGB és RGBA szintaxisát és főbb különbségeit.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Svekis L. Front-End Web Development with Modern HTML, CSS, and JavaScript. Packt Publishing, 2022. (англійською).
- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- Infojegyzet.hu. Webszerkesztés: CSS alapok. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 7. Блокова модель CSS. Візуальна модель форматування. / A CSS blokkmodell. A formázás vizuális modellje.

1. Опишіть чотири основні компоненти блокової моделі CSS: content, padding, border та margin. Намалуйте схему та поясніть призначення кожного компонента. / Ismertesse a CSS dobozmodell négy fő összetevőjét: a tartalom (content), a belső térköz (padding), a keret (border) és a külső margó (margin). Készítsen rajzot, és magyarázza el mindegyik elem szerepét.
2. Що таке властивість box-sizing? Поясніть різницю між значеннями content-box (за замовчуванням) та border-box. Чому border-box часто вважається більш інтуїтивним для розробки макетів? / Mi az a box-sizing tulajdonság? Magyarázza el a content-box (alapértelmezett) és a border-box érték közötti különbséget. Miért tekinthető a border-box sok esetben intuitívabbnak az elrendezések tervezésénél?

3. Припустимо, у вас є елемент з такими стилями: width: 200px;, height: 100px;, padding: 20px;, border: 5px solid black;, margin: 10px;. Розрахуйте повну ширину та висоту, яку цей елемент займе на сторінці, для обох значень box-sizing (content-box та border-box). / Tegyük fel, hogy van egy elem a következő stílusokkal: width: 200 px; height: 100 px; padding: 20 px; border: 5 px solid black; margin: 10 px. Számítsa ki a teljes szélességet és magasságot, amelyet ez az elem elfoglal az oldalon mindkét box-sizing érték (content-box és border-box) esetén.
4. Поясніть, як працює "схлопування відступів" (margin collapsing) для вертикальних margin. У яких випадках воно відбувається? / Magyarázza el, hogyan működik a függőleges margók „összeomlása” (margin collapsing). Milyen esetekben következik be?
5. Як можна встановити різні значення для padding або margin для кожної зі сторін блоку, використовуючи скорочений синтаксис (з одним, двома, трьома та чотирма значеннями)? / Hogyan adhatók meg különböző értékek a padding vagy a margin számára a blokk mindegyik oldalán, rövidített szintaxissal (egy, két, három vagy négy értékkel)?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Svekis L. Front-End Web Development with Modern HTML, CSS, and JavaScript. Packt Publishing, 2022. (англійською).
- MDN Web Docs. The box model. Mozilla Foundation. Оновлюється постійно. (англійською).
- Infojegyzet.hu. Webszerkesztés: CSS alapok. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 8. Відображення елементів. Плаваючі елементи. Псевдо класи та елементи. / Az elemek megjelenítése. Lebegő elemek. Pszeudo osztályok és elemek.

1. Опишіть основні значення властивості display: block, inline, inline-block та none. Які ключові відмінності в їхній поведінці та можливостях стилізації (наприклад, застосування width та height)? / Ismertesse a display tulajdonság fő értékeit: block, inline, inline-block és none. Miben különböznek ezek viselkedése és stílizálási lehetőségei (például width és height alkalmazása)?
2. Що таке властивість float? Поясніть її початкове призначення та чому вона вважається застарілим методом для побудови основних макетів сторінок. Що таке "очищення потоку" (clearing floats) і як його досягти? / Mi az a float tulajdonság? Magyarázza el eredeti rendeltetését, és miért tekintik elavult módszernek alapvető oldalelrendezések kialakítására. Mit jelent a lebegő elemek „törlése” (clearing floats), és hogyan valósítható meg?

3. Поясніть різницю між псевдокласами та псевдоелементами в CSS. Наведіть по два приклади кожного. / Magyarázza el a CSS pszeudoosztályai és pszeudoelemek közötti különbséget. Hozzon két-két példát mindegyikre.
4. Опишіть призначення псевдокласів :hover, :focus та :active. Як їх використання покращує інтерактивність та користувацький досвід? / Ismertesse a :hover, :focus és :active pszeudoosztályok funkcióját. Hogyan javítja ezek használata az interaktivitást és a felhasználói élményt?
5. Наведіть практичний приклад використання псевдоелементів ::before або ::after для додавання декоративного контенту до елемента без зміни HTML-розмітки. / Hozzon gyakorlati példát a ::before vagy ::after pszeudoelemek használatára dekoratív tartalom hozzáadására egy elemhez anélkül, hogy megváltoztatná a HTML-markupot.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Svekis L. Front-End Web Development with Modern HTML, CSS, and JavaScript. Packt Publishing, 2022. (англійською).
- Meloni J. C., Kyrnin J. HTML, CSS, and JavaScript All in One: Covering HTML5, CSS3, and ES6, Sams Teach Yourself. Sams Publishing, 2021. (англійською).
- Infojegyzet.hu. Webszerkesztés: CSS alapok. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 9. Елементи CSS3. FlexBox. Медіа запити. Сімейство рекомендацій для визначення перетворень і представлення XML: XSL. / CSS3 elemek. FlexBox. Media.

1. Що таке CSS Flexbox? Опишіть основні поняття: flex-контейнер, flex-елементи, головна вісь (main axis) та поперечна вісь (cross axis). / Mi az a CSS Flexbox? Ismertesse a fő fogalmakat: flex-konténer, flex-elem, fő tengely (main axis) és keresztirányú tengely (cross axis).
2. Назвіть та поясніть призначення трьох властивостей flex-контейнера (наприклад, display: flex, flex-direction, justify-content, align-items) для керування розташуванням елементів. / Soroljon fel és magyarázzon három flex-konténer tulajdonságot (például display: flex, flex-direction, justify-content, align-items), amelyek irányítják az elemek elhelyezését.
3. Що таке медіа-запити (@media) і яка їхня роль у створенні адаптивного веб-дизайну (Responsive Web Design)? Наведіть приклад медіа-запиту, який змінює колір фону сторінки для екранів шириною менше 600 пікселів. / Mit jelentenek a médialekérdezések (@media) és milyen szerepük van a reszponzív webtervezésben? Hozzon példát olyan médialekérdezésre, amely 600 pixelnél kisebb szélességű képernyők esetén módosítja az oldal háttérszínét.

4. Назвіть три нові можливості, що з'явилися в CSS3, окрім Flexbox (наприклад, border-radius, box-shadow, градієнти, трансформації, анімації), та коротко опишіть їх. / Soroljon fel három új lehetőséget a CSS3-ban a Flexbox mellett (például border-radius, box-shadow, gradiensek, transzformációk, animációk), és írja le röviden ezeket.

5. Що таке XSL (eXtensible Stylesheet Language)? Поясніть його основне призначення у трансформації XML-документів та його ключову відмінність від CSS. / Mi az XSL (eXtensible Stylesheet Language)? Magyarázza el fő célját az XML-dokumentumok átalakításában és legfontosabb különbségét a CSS-hez képest.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Svekis L. Front-End Web Development with Modern HTML, CSS, and JavaScript. Packt Publishing, 2022. (англійською).
- MDN Web Docs. Flexbox. Mozilla Foundation. Оновлюється постійно. (англійською).
- Infojegyzet.hu. Webszerkesztés: CSS alapok. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 10. Бібліотека стилів Bootstrap. / Bootstrap stíluskönyvtár.

1. Що таке CSS-фреймворк? Поясніть основну мету використання фреймворків, таких як Bootstrap, у веб-розробці. / Mi az a CSS keretrendszer? Ismertesse a keretrendszerek, például a Bootstrap használatának fő célját a webfejlesztésben.

2. Опишіть, як працює система сіток (Grid System) в Bootstrap. Які основні компоненти (контейнер, рядок, колонка) використовуються для створення адаптивних макетів? / Mutassa be, hogyan működik a Bootstrap rácsrendszere (Grid System). Mely alapegységeket (konténer, sor, oszlop) használjuk az adaptív elrendezések elkészítéséhez?

3. Наведіть приклад HTML-розмітки для створення двоколонкового макету за допомогою Bootstrap, де на мобільних пристроях колонки розташовуються одна під одною, а на настільних — поруч. / Adjon példát HTML-markupra, amely kétszlopos elrendezést hoz létre Bootstrap segítségével, úgy hogy mobil eszközökön az oszlopok egymás alatt, asztali kijelzőn egymás mellett jelenjenek meg.

4. Окрім системи сіток, назвіть щонайменше чотири типи готових компонентів, які надає Bootstrap (наприклад, кнопки, навігаційні панелі, модальні вікна, картки), та опишіть їхні переваги. / A rácsrendszeren kívül soroljon fel legalább négy kész komponenst, amelyet a Bootstrap kínál (például gombok, navigációs sávok, modális ablakok, kártyák), és ismertesse előnyeiket.

5. Які переваги та недоліки використання Bootstrap у проєкті? У яких випадках його використання є доцільним, а коли краще писати CSS з нуля? / Milyen előnyei és hátrányai vannak a Bootstrap használatának egy projektben? Milyen esetekben célszerű alkalmazni, és mikor célszerű a CSS-t saját kézzel megírni?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівецьк. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- GetBootstrap.com. Офіційна документація Bootstrap 5. Оновлюється постійно. (англійською).
- Svekis L. HTML, CSS, and JavaScript for Beginners: A Web Design Course. Packt Publishing, 2021. (англійською).
- Infojegyzet.hu. Webszerkesztés: Bootstrap. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Модуль 2. Програмування на стороні клієнта / Modul 2. Kliensoldali programozás

Змістовий модуль 3. Програмування на стороні клієнта / Tartalmi modul 3. Kliensoldali programozás

Тема 11. Основи мови JavaScript. / JavaScript alapok

1. Поясніть різницю між var, let та const для оголошення змінних у JavaScript, зокрема щодо області видимості (scope) та можливості переприсвоєння. / Magyarázza el a var, let és const változók közötti különbséget a JavaScriptben, különösen a láthatóság (scope) és az újraértékelés lehetősége szempontjából.
2. Назвіть примітивні типи даних у JavaScript. Як перевірити тип змінної за допомогою оператора typeof? / Sorolja fel a JavaScriptben elérhető primitív adattípusokat. Hogyan ellenőrizhető egy változó típusa a typeof operátorral?
3. Що таке функція в JavaScript? Опишіть синтаксис оголошення функції (function declaration) та функціонального виразу (function expression). / Mit jelent a függvény fogalma a JavaScriptben? Ismertesse a function declaration és function expression szintaxisát.
4. Поясніть різницю між операторами суворого (===) та несуворого (==) порівняння. Наведіть приклад, де їх результати будуть відрізнятися. / Magyarázza el a szigorú (===) és nem szigorú (==) összehasonlító operátorok közötti különbséget. Hozzon példát, ahol eltérő eredményt adnak.
5. Напишіть функцію, яка приймає масив чисел та повертає їхню суму, використовуючи цикл for. / Írjon olyan függvényt, amely egy számokból álló tömböt vesz át, és for ciklus használatával visszaadja azok összegét.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Фрімен Е., Робсон Е. Head First. Програмування на JavaScript. Фабула, 2022. 672 с. (українською).
- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Flanagan D. JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, 2020. (англійською).
- Infojegyzet.hu. Webszerkesztés: JavaScript alapok. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 12. Використання JavaScript в HTML. / JavaScript használata a HTML-ben.

1. Опишіть два основні способи підключення JavaScript-коду до HTML-сторінки. Який спосіб є кращим для великих скриптів і чому? / Ismertesse a JavaScript-kód HTML-oldalhoz kapcsolásának két fő módját. Melyik módszer előnyösebb nagyméretű szkriptek esetén, és miért?
2. Поясніть, чому тег `<script>` часто розміщують перед закриваючим тегом `</body>`. Які проблеми можуть виникнути, якщо розмістити його в `<head>` без додаткових атрибутів? / Magyarázza el, miért helyezik gyakran a `<script>` taget a `</body>` zárótag elé. Milyen problémák adódhatnak, ha a `<head>` szakaszban tesszük elhelyezése nélkül további attribútumok?
3. Що таке атрибути `async` та `defer` для тегу `<script>`? Поясніть різницю в їхній поведінці щодо завантаження та виконання скрипта. / Mit jelent a `<script>` tag `async` és `defer` attribútuma? Magyarázza el, hogyan különböznek a betöltési és futtatási sorrend szempontjából.
4. Як можна вивести інформацію в консоль розробника? Для чого використовується `console.log()` у процесі розробки та налагодження коду? / Hogyan írhatunk információt a fejlesztői konzolra? Mire használjuk a `console.log()` függvényt fejlesztés és hibakeresés során?
5. Опишіть призначення функцій `alert()`, `prompt()` та `confirm()`. Чому їх використання не рекомендується в сучасних веб-додатках для створення користувацьких інтерфейсів? / Magyarázza el az `alert()`, `prompt()` és `confirm()` függvények szerepét. Miért nem ajánlott ezeket használni modern webalkalmazások felhasználói felületének létrehozásakor?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Фрімен Е., Робсон Е. Head First. Програмування на JavaScript. Фабула, 2022. 672 с. (українською).
- Курінний С. Розробка веб-сайтів для початківців: HTML, CSS, JavaScript. 2022. (українською).

- Flanagan D. JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, 2020. (англійською).
- Infojegyzet.hu. Webszerkesztés: JavaScript alapok. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 13. Обробка подій на веб-сторінці. / Események kezelése egy weboldalon.

1. Що таке подія (event) у контексті веб-сторінки? Наведіть приклади трьох різних типів подій. / Mit jelent az esemény (event) a weboldal kontextusában? Soroljon fel három különböző eseménytípust.
2. Порівняйте два способи призначення обробника подій: через HTML-атрибут (наприклад, `onclick="..."`) та через JavaScript за допомогою `element.addEventListener()`. Який спосіб є більш гнучким та рекомендованим? / Hasonlítsa össze a két módszert az eseménykezelők hozzárendelésére: HTML-attribútum használatával (például `onclick="..."`) és JavaScriptből az `element.addEventListener()` metódussal. Melyik módszer rugalmasabb és ajánlottabb?
3. Що таке об'єкт події (event object), який передається в функцію-обробник? Яку корисну інформацію він містить (наприклад, `event.target`, `event.preventDefault()`)? / Mi az az eseményobjektum, amelyet az eseménykezelő funkció kap? Milyen hasznos információkat tartalmaz (például `event.target`, `event.preventDefault()`)?
4. Поясніть механізми спливання (bubbling) та перехоплення (capturing) подій. Як можна зупинити спливання події? / Magyarázza el az események buborékosodásának (bubbling) és elfogásának (capturing) mechanizmusát. Hogyan lehet megállítani az esemény buborékolását?
5. Напишіть код, який при кліку на кнопку додає до лічильника на сторінці одиницю та оновлює його текстовий вміст. / Írjon kódot, amely kattintáskor egy gomb megnyomására növeli a számláló értékét az oldalon, és frissíti annak szöveges tartalmát.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Фрімен Е., Робсон Е. Head First. Програмування на JavaScript. Фабула, 2022. 672 с. (українською).
- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Flanagan D. JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, 2020. (англійською).
- Infojegyzet.hu. JS feladatok: eseménykezelés. Online tananyagok és feladatok. Frissítve: 2024. (угорською).

Тема 14. HTML DOM. Створення та видалення елементів HTML на JavaScript. / HTML DOM. HTML elemek létrehozása és törlése JavaScriptben

1. Що таке Document Object Model (DOM)? Поясніть, як браузер представляє HTML-документ у вигляді деревоподібної структури об'єктів. / Mi az a Document Object Model (DOM)? Magyarázza el, hogyan jeleníti meg a böngésző a HTML-dokumentumot objektumfaként.
2. Порівняйте методи пошуку елементів у DOM: getElementById(), getElementsByClassName(), querySelector() та querySelectorAll(). Які з них повертають "живу" колекцію, а які статичну? / Hasonlítsa össze a DOM elemeinek kiválasztási metódusait: getElementById(), getElementsByClassName(), querySelector() és querySelectorAll(). Melyek adnak élő gyűjteményt, és melyek statikusokat?
3. Як можна змінити вміст існуючого елемента? Поясніть різницю між властивостями innerHTML, innerText та textContent, зокрема з точки зору безпеки (XSS-атаки). / Hogyan módosítható egy meglévő elem tartalma? Magyarázza el az innerHTML, innerText és textContent közti különbséget, különösen a biztonság (XSS-támadás) szempontjából.
4. Опишіть процес створення нового HTML-елемента (наприклад, <p>), додавання до нього текстового вмісту та вставки його в кінець існуючого <div> на сторінці. / Ismertesse egy új HTML-elem (például <p>) létrehozásának, szöveg hozzáadásának, majd egy meglévő <div> végére történő beszúrásának folyamatát.
5. Як можна видалити елемент зі сторінки за допомогою JavaScript? Наведіть приклад коду. / Hogyan távolítható el egy elem az oldalról JavaScript segítségével? Hozzon példát kóddal.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Фрімен Е., Робсон Е. Head First. Програмування на JavaScript. Фабула, 2022. 672 с. (українською).
- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівецьк. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Svekis L. HTML, CSS, and JavaScript for Beginners: A Web Design Course. Packt Publishing, 2021. (англійською).
- Infojegyzet.hu. JS jegyzet: objektumok hierarchiája. Online tananyagok. Frissítve: 2024. (угорською).

Тема 15. Основи деяких популярних фреймворків JavaScript: ReactJS та AngularJS. / Néhány népszerű JavaScript keretrendszer alapjai: ReactJS és AngularJS.

1. Що таке компонентний підхід у розробці інтерфейсів? Поясніть, які переваги він надає порівняно з традиційною маніпуляцією DOM. / Mit jelent a

компонентів, які використовують виртуальний DOM? Мilyen előnyöket kínál a hagyományos DOM-manipulációval szemben?

2. Опишіть ключові поняття React: компонент, JSX, стан (state) та властивості (props). Як відбувається передача даних від батьківського компонента до дочірнього? / Ismertesse a React kulcsfogalmait: komponens, JSX, állapot (state) és tulajdonságok (props). Hogyan történik az adatátadás a szülőkomponensből a gyermekkomponensbe?

3. Що таке віртуальний DOM (Virtual DOM) і яку роль він відіграє в оптимізації продуктивності React-додатків? / Mi az a virtuális DOM (Virtual DOM) és milyen szerepet játszik a React-alkalmazások teljesítményének optimalizálásában?

4. Порівняйте React та Angular за їхньою архітектурою. Чому React називають бібліотекою, а Angular — фреймворком? / Hasonlítsa össze a React és az Angular architektúráját. Miért nevezik a Reactet könyvtárnak és az Angulart keretrendszernek?

5. Що таке хуки (Hooks) в React? Поясніть призначення хуків useState та useEffect на простих прикладах. / Mik a React hookjai? Magyarázza el a useState és a useEffect hookok célját egyszerű példákon keresztül.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, 6th Edition. O'Reilly Media, 2021. (включає React) (англійською).
- Stefanov S. React: Up & Running: Building Web Applications, 2nd Edition. O'Reilly Media, 2021. (англійською).
- React.dev. Офіційна документація React. Оновлюється постійно. (англійською).

Тема 16. Асинхронний JavaScript. AJAX. / Aszinkron JavaScript. AJAX.

1. Поясніть, що таке асинхронність у JavaScript. Як працює цикл подій (Event Loop) для обробки асинхронних операцій, не блокуючи основний потік виконання? / Magyarázza el, mi az aszinkronitás a JavaScriptben. Hogyan működik az eseményciklus (Event Loop) az aszinkron műveletek kezelésekor anélkül, hogy blokkolná a fő futási szálát?

2. Що таке AJAX (Asynchronous JavaScript and XML)? Опишіть, як ця технологія дозволяє оновлювати частини веб-сторінки без повного перезавантаження. / Mit jelent az AJAX (Asynchronous JavaScript and XML)? Ismertesse, hogyan teszi lehetővé a weboldal egyes részeinek frissítését teljes újratöltés nélkül.

3. Порівняйте три підходи до роботи з асинхронним кодом: callback-функції, проміси (Promise) та синтаксис async/await. Які переваги надає кожен

наступний підхід? / Hasonlítsa össze az aszinkron kód kezelésének három módját: callback függvények, ígéret (Promise) és az async/await szintaxis. Milyen előnyökkel jár mindegyik a következőhöz képest?

4. Що таке fetch API? Напишіть приклад коду, який робить GET-запит до публічного API та виводить отримані дані у консоль. / Mi az a fetch API? Írjon példakódot, amely egy nyilvános API-hoz GET kérést intéz, és az eredményt a konzolra írja.

5. Що таке JSON (JavaScript Object Notation)? Поясніть його синтаксис та роль у обміні даними між клієнтом та сервером. / Mit jelent a JSON (JavaScript Object Notation)? Ismertesse szintaxisát és szerepét az ügyfél és a szerver közötti adatcsereben.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Фрімен Е., Робсон Е. Head First. Програмування на JavaScript. Фабула, 2022. 672 с. (українською).
- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівецьк. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Flanagan D. JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, 2020. (англійською).
- Infojegyzet.hu. Ajax pofonegyszerűen. Online tananyagok. Frissítve: 2021. (угорською).

Тема 17. Елементи jQuery. / A jQuery elemei.

1. Яку історичну роль відіграла бібліотека jQuery у розвитку веб-розробки? Які проблеми "болючих точок" крос-браузерної сумісності вона вирішувала? / Milyen történelmi szerepet játszott a jQuery könyvtár a webfejlesztésben? Milyen „fájó pontokat” oldott meg a böngészők közötti kompatibilitás terén?

2. Порівняйте синтаксис вибору елемента за класом та додавання обробника події click за допомогою jQuery та нативного JavaScript (document.querySelector та addEventListener). / Hasonlítsa össze egy elem osztály szerinti kiválasztását és kattintási eseménykezelő hozzáadását jQuery-ben és natív JavaScriptben (document.querySelector és addEventListener használatával).

3. Що таке "ланцюжок методів" (method chaining) в jQuery? Наведіть приклад, що демонструє цю можливість. / Mit jelent a „metódusláncolás” (method chaining) a jQuery-ben? Mutasson be egy példát, amely szemlélteti ezt a lehetőséget.

4. Назвіть три приклади ефектів або анімацій, які легко реалізувати за допомогою jQuery (наприклад, fadeIn(), slideToggle(), animate()). / Nevezzen meg három olyan effektust vagy animációt, amelyet könnyen megvalósíthatunk jQuery-vel (például fadeIn(), slideToggle(), animate()).

5. Обґрунтуйте, чому популярність jQuery знизилася з розвитком сучасного JavaScript (ES6+) та CSS3. Чи існують сценарії, де її використання

може бути доцільним сьогодні? / Indokolja, miért csökkent a jQuery népszerűsége a modern JavaScript (ES6+) és a CSS3 fejlődésével. Van-e olyan eset, amikor ma is célszerű a használata?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек D. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- Svekis L. HTML, CSS, and JavaScript for Beginners: A Web Design Course. Packt Publishing, 2021. (включає jQuery) (англійською).
- jQuery.com. Офіційна документація jQuery. Оновлюється постійно. (англійською).

Модуль 3. Програмування на стороні сервера / Modul 3. Szerveroldali programozás

Змістовий модуль 4. Програмування на стороні сервера / Tartalmi modul 4. Szerveroldali programozás

Тема 18. Основи мови PHP. / PHP alapjai.

1. Як PHP-код вбудовується в HTML-документ? Наведіть приклад синтаксису для виведення значення змінної всередині HTML-тегу. / Hogyan épül be a PHP-kód egy HTML-dokumentumba? Hozzon példát szintaxisra, amellyel egy változó értékét írjuk ki egy HTML-tagben.
2. Поясніть різницю між індексними та асоціативними масивами в PHP. Наведіть приклади створення та доступу до елементів для обох типів. / Magyarázza el a PHP-ben az indexelt és asszociatív tömbök közötti különbséget. Hozzon példát mindkét típus létrehozására és elemhez való hozzáférésre.
3. Що таке суперглобальні масиви в PHP? Опишіть призначення масивів \$_GET, \$_POST та \$_SERVER. / Mit jelentenek a PHP-ben a szuperglobális tömbök? Írja le a \$_GET, \$_POST és \$_SERVER tömbök célját.
4. Напишіть PHP-функцію, яка приймає два числа як аргументи та повертає їх добуток. / Írjon olyan PHP-függvényt, amely két számot vesz argumentumként és visszaadja szorzatukat.
5. Порівняйте оператори конкатенації рядків (.) та додавання (+) в PHP. Що відбудеться, якщо спробувати "додати" два рядки? / Hasonlítsa össze a PHP-ben a karakterláncok összefűzésére szolgáló (.) operátort és az összeadás (+) operátort. Mi történik, ha két karakterláncot „összeadunk”?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек D. O. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- PHP.net/manual/uk. Офіційна документація PHP українською мовою. Оновлюється постійно. (українською).
- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- Infojegyzet.hu. PHP jegyzet: Szerveroldali programozás. Online tananyagok. Frissítve: 2024. (угорською).

Тема 19. Обробка веб-форм на мові PHP. Сесії. Ключі. / Webes űrlapok feldolgozása PHP-ben. Session

1. Опишіть процес отримання даних, надісланих з HTML-форми методом POST, у PHP-скрипті. Чому перед використанням цих даних їх необхідно валідувати та очищувати? / Ismertesse, hogyan kapja meg a PHP-skript a POST metódussal küldött adatokat egy HTML-űrlapról. Miért kell ezeket az adatokat érvényesíteni és tisztítani felhasználás előtt?
2. Що таке сесії в PHP? Поясніть, як за допомогою session_start() та суперглобального масиву \$_SESSION можна зберігати інформацію про користувача між різними запитами. / Mit jelentenek a PHP-ben a sessionök? Magyarázza el, hogyan tárolhatók felhasználói információk a session_start() és a szuperglobális \$_SESSION tömb segítségével a különböző kérések között.
3. Порівняйте механізми сесій та cookies. Яка між ними основна різниця щодо місця зберігання даних та безпеки? / Hasonlítsa össze a munkamenetek és a süтик működését. Mi a fő különbség az adattárolás helye és a biztonság szempontjából?
4. Напишіть PHP-скрипт, який перевіряє, чи було надіслано дані з форми, та виводить повідомлення про успіх або помилку, якщо певне поле порожнє. / Írjon olyan PHP-skriptet, amely ellenőrzi, hogy érkezett-e adat az űrlapról, és siker-vagy hibaüzenetet jelenít meg, ha egy mező üres.
5. Які загрози безпеці, такі як Cross-Site Scripting (XSS) та Cross-Site Request Forgery (CSRF), пов'язані з обробкою форм, і які базові методи захисту існують в PHP? / Milyen biztonsági kockázatok, például Cross-Site Scripting (XSS) és Cross-Site Request Forgery (CSRF) kapcsolódnak az űrlapfeldolgozáshoz, és milyen alapvető védelmi módszerek vannak PHP-ben?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек D. O. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Welling L., Thomson L. PHP and MySQL Web Development, 5th Edition. Addison-Wesley Professional, 2016. (англійською).

- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- Infojegyzet.hu. PHP-MySQL jegyzet: Regisztráció megvalósítása. Online tananyagok. Frissítve: 2020. (угорською).

Тема 20. Асинхронний PHP – AJAX. Робота з XML в PHP. / Aszinkron PHP - AJAX. XML-el való munka PHP-ben.

1. Опишіть архітектуру взаємодії клієнта та сервера при використанні AJAX. Яку роль відіграє PHP-скрипт у цьому процесі? / Ismertesse az ügyfél és szerver közötti kommunikáció architektúráját AJAX használatakor. Milyen szerepe van a PHP-skriptnek ebben a folyamatban?
2. Як PHP-скрипт може повернути дані у форматі JSON, щоб їх було легко обробити на стороні клієнта за допомогою JavaScript? Наведіть приклад використання функції `json_encode()`. / Hogyan adhat vissza egy PHP-skript adatokat JSON formátumban, hogy a JavaScript oldal könnyen feldolgozhassa azokat? Hozzon példát a `json_encode()` függvény használatára.
3. Що таке розширення SimpleXML в PHP? Поясніть, як за його допомогою можна прочитати дані з XML-файлу та отримати доступ до його елементів та атрибутів. / Mit jelent a SimpleXML kiterjesztés a PHP-ben? Magyarázza el, hogyan lehet vele XML-fájlból adatot olvasni, és hozzáférni az elemekhez és attribútumokhoz.
4. Напишіть простий PHP-скрипт, який на AJAX-запит повертає масив з кількох імен у форматі JSON. / Írjon egyszerű PHP-skriptet, amely AJAX-kérésre több nevet tartalmazó tömböt ad vissza JSON formátumban.
5. Порівняйте формати JSON та XML для обміну даними в AJAX-запитах. Які переваги має JSON, що зробили його більш популярним у сучасній веб-розробці? / Hasonlítsa össze a JSON és XML formátumot az AJAX adatsereben. Milyen előnyei tették a JSON-t népszerűbbé a modern webfejlesztésben?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Welling L., Thomson L. PHP and MySQL Web Development, 5th Edition. Addison-Wesley Professional, 2016. (англійською).
- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- PHP.net/manual/uk. Офіційна документація PHP по SimpleXML та JSON. Оновлюється постійно. (українською).

Модуль 4. Робота з базами даних для веб MySQL / Modul 4. Munka webes adatbázisokkal MySQL

Змістовий модуль 5. Робота з базами даних для веб MySQL / Tartalmi modul 5. Adatbázisokkal való munka a webhez MySQL

Тема 21. Основи роботи з базами даних MySQL. Налаштування середовища MySQL. phpMyAdmin. / A MySQL adatbázisokkal való munka alapjai. A MySQL környezet beállítása. phpMyAdmin.

1. Що таке реляційна база даних? Поясніть основні поняття: таблиця, запис (рядок), поле (стовпець), первинний ключ (primary key) та зовнішній ключ (foreign key). / Mi az a relációs adatbázis? Ismertesse az alapfogalmakat: tábla, rekord (sor), mező (oszlop), elsődleges kulcs (primary key) és idegen kulcs (foreign key).
2. Опишіть призначення інструменту phpMyAdmin. Які основні операції з базами даних та таблицями (створення, редагування, видалення) можна виконати через його графічний інтерфейс? / Mutassa be a phpMyAdmin eszköz célját. Milyen alapszerveletek végezhetők el grafikus felületén adatbázisokkal és táblákkal (létrehozás, szerkesztés, törlés)?
3. Спроектуйте структуру двох пов'язаних таблиць для простого інтернет-магазину: products (з полями id, name, price) та categories (з полями id, name). Який зв'язок (один-до-багатьох, багато-до-багатьох) буде між ними і як його реалізувати? / Tervezzon két kapcsolódó táblát egy egyszerű webáruház számára: products (mezők: id, name, price) és categories (mezők: id, name). Milyen kapcsolat lesz közöttük (egy-a-sokhoz, sok-a-sokhoz), és hogyan valósítaná meg?
4. Назвіть та опишіть щонайменше чотири типи даних, що використовуються в MySQL для стовпців таблиць (наприклад, INT, VARCHAR, TEXT, DATE). / Soroljon fel és ismertessen legalább négy adattípust, amelyet a MySQL használ oszlopok számára (például INT, VARCHAR, TEXT, DATE).
5. Що таке індекс у базі даних і для чого він використовується? Чому первинні ключі автоматично індексуються? / Mi az index az adatbázisban, és mire használjuk? Miért indexeli a rendszer automatikusan az elsődleges kulcsokat?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Гайна Г.А.. Основи проектування баз даних: навчальний посібник. Навчально-методичне видання, 2018. (українською).
- Онлайн ресурс. Організація Баз Даних і Знань: Лабораторні роботи (MySQL, MongoDB). Навчальний посібник, 2022. (українською).
- Welling L., Thomson L. PHP and MySQL Web Development, 5th Edition. Addison-Wesley Professional, 2016. (англійською).
- Infojegyzet.hu. MySQL adatbázisok és lekérdezési feladatok. Online tananyagok. Frissítve: 2024. (угорською).

Тема 22. Основні команди роботи з базами MySQL. Агрегатні функції. Об'єднання. / A MySQL adatbázisokkal való munka alapvető parancsai. Összesítő függvények.

1. Напишіть SQL-запити для виконання чотирьох базових CRUD-операцій: створення нового запису (INSERT), читання записів (SELECT), оновлення існуючого запису (UPDATE) та видалення запису (DELETE). / Írjon SQL-lekérdezéseket a négy alapvető CRUD műveletre: új bejegyzés (INSERT), bejegyzések lekérdezése (SELECT), meglévő bejegyzés módosítása (UPDATE) és bejegyzés törlése (DELETE).
2. Поясніть призначення ключових слів WHERE, ORDER BY та LIMIT у SELECT-запиті. Наведіть приклад запиту, що використовує всі три. / Magyarázza el a WHERE, ORDER BY és LIMIT kulcsszavak szerepét a SELECT lekérdezésben. Hozzon példát egy olyan lekérdezésre, amely mindhármát használja.
3. Що таке агрегатні функції в SQL? Опишіть призначення функцій COUNT(), SUM(), AVG(), MAX() та MIN() та наведіть приклад запиту з використанням GROUP BY. / Mit jelentenek az aggregációs függvények az SQL-ben? Ismertesse a COUNT(), SUM(), AVG(), MAX() és MIN() függvények célját, és hozzon példát lekérdezésre GROUP BY használatával.
4. Поясніть різницю між INNER JOIN та LEFT JOIN. Наведіть приклад, коли використання LEFT JOIN є необхідним. / Magyarázza el a különbséget az INNER JOIN és a LEFT JOIN között. Hozzon példát, amikor a LEFT JOIN használata szükséges.
5. Маючи таблиці authors (id, name) та books (id, title, author_id), напишіть SQL-запит, який виведе назви всіх книг разом з іменами їхніх авторів. / Adott authors (id, name) és books (id, title, author_id) táblák esetén írjon SQL-lekérdezést, amely az összes könyv címét kiírja a szerző neveivel együtt.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Гайна Г.А.. Основи проектування баз даних: навчальний посібник. Навчально-методичне видання, 2018. (українською).
- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- Welling L., Thomson L. PHP and MySQL Web Development, 5th Edition. Addison-Wesley Professional, 2016. (англійською).
- Infojegyzet.hu. MySQL lekérdezések: Reptér adatbázis. Online feladatok. Frissítve: 2024. (угорською).

Тема 23. Робота з базами даних із мови програмування PHP. / Adatbázisokkal való munka a PHP programozási nyelven.

1. Що таке PDO (PHP Data Objects)? Поясніть, які переваги надає цей інтерфейс для роботи з базами даних у PHP порівняно зі старими розширеннями, такими як mysql_. / Mi az a PDO (PHP Data Objects)?

Magyarázza el, milyen előnyöket nyújt ez az adatbázis-kezelő interfész a régi mysql_ kiterjesztésekhez képest.

2. Опишіть послідовність кроків для підключення до бази даних MySQL з PHP за допомогою PDO та виконання простого SELECT-запиту. / Írja le a lépéseket a MySQL adatbázishoz való kapcsolódáshoz PHP-ben PDO használatával, és egy egyszerű SELECT lekérdezés végrehajtását.

3. Що таке SQL-ін'єкція? Поясніть на прикладі, як зловмисник може використати вразливість у коді для маніпуляції базою даних. / Mi az SQL-injekció? Ismertesse egy példán keresztül, hogyan használhatja ki egy támadó a kódbeli sérülékenységet az adatbázis manipulálására.

4. Що таке підготовлені вирази (prepared statements) і як вони допомагають запобігти SQL-ін'єкціям? Наведіть приклад використання підготовленого виразу в PDO. / Mit jelent a prepared statement, és hogyan segít megelőzni az SQL-injekciót? Hozzon példát prepared statement alkalmazására PDO-val.

5. Напишіть PHP-скрипт, який отримує дані з POST-запиту (ім'я та email) та вставляє їх у таблицю users бази даних, використовуючи безпечний підхід з підготовленими виразами. / Írjon PHP-skriptet, amely a POST kérésből kapott adatokat (név és e-mail) a users táblába illeszti be biztonságos módon, prepared statement használatával.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською).
- Welling L., Thomson L. PHP and MySQL Web Development, 5th Edition. Addison-Wesley Professional, 2016. (англійською).
- Nixon R. Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською).
- Infojegyzet.hu. PHP-MySQL mintapéllda: Európa országai. Online tananyagok. Frissítve: 2020. (угорською).

Модуль 5. Вебдоступність та інклюзивні технології / Modul 5. Webes akadálymentesítés és inkluzív technológiák

Змістовий модуль 6. Розробка доступних вебінтерфейсів / Tartalmi modul 6. Hozzáférhető webes felületek fejlesztése

Тема 24. Поняття доступності та універсального дизайну. Групи користувачів з особливими потребами та їхні проблеми. / Az akadálymentesítés és az egyetemes tervezés fogalma. A speciális igényű felhasználók csoportjai és problémái.

1. Дайте визначення веб-доступності. Поясніть, чому це важливо не лише для людей з інвалідністю, але й для ширшого кола користувачів (наприклад,

люди похилого віку, користувачі з повільним інтернетом). / Adjon definíciót a webes hozzáférhetőségre. Magyarázza el, miért fontos ez nem csak a fogyatékkal élőknek, hanem szélesebb körű felhasználók számára is (például idősek, lassú internettel rendelkezők).

2. Що таке універсальний дизайн? Наведіть приклад з реального світу (наприклад, пандус) та поясніть, як цей принцип застосовується у веб-розробці. / Mi az univerzális tervezés? Mondjon egy valós példát (például rámpa), és ismertesse, hogyan alkalmazható ez az elv a webfejlesztésben.

3. Назвіть чотири основні категорії порушень, які впливають на взаємодію з вебом (зорові, слухові, моторні, когнітивні). / Soroljon fel négy fő fogyatékosági kategóriát, amely hatással van a webhasználatra (látási, hallási, motoros és kognitív).

4. Опишіть щонайменше три конкретні бар'єри, з якими може зіткнутися незрячий користувач, що використовує скрінрідер, на погано розробленому сайті. / Írjon le legalább három konkrét akadályt, amellyel egy képernyőolvasót használó vak felhasználó találkozhat egy rosszul tervezett weboldalon.

5. Поясніть, чому покладання виключно на колір для передачі важливої інформації (наприклад, позначення помилки червоним кольором) є проблемою доступності. / Magyarázza el, miért jelent problémát a hozzáférhetőség szempontjából, ha kizárólag színnel közölnek fontos információt (például hibát pirossal jelölnek).

Рекомендовані джерела інформації / Ajánlott szakirodalom

- ПРООН в Україні. Короткий посібник з цифрової доступності. 2023. (українською).
- W3C Web Accessibility Initiative (WAI). Introduction to Web Accessibility. Оновлюється постійно. (англійською).
- European Commission. Proposal for a Directive on the approximation of the laws, regulations and administrative provisions of the Member States as regards the accessibility requirements for products and services. 2022. (англійською).

Тема 25. Стандарт WCAG 2.0 та основні принципи вебдоступності. / WCAG 2.0 szabvány és a webes akadálymentesség alapelvei.

1. Що таке WCAG (Web Content Accessibility Guidelines)? Яка організація розробляє та підтримує цей стандарт? / Mi az a WCAG (Web Content Accessibility Guidelines)? Mely szervezet fejlesztí és tartja karban ezt a szabványt?

2. Назвіть та коротко опишіть чотири фундаментальні принципи WCAG (POUR: Perceivable, Operable, Understandable, Robust). / Nevezze meg és röviden ismertesse a WCAG négy alapelvét (POUR: érzékelhető, kezelhető, érthető, robusztus).

3. Поясніть, що означають рівні відповідності WCAG: А, АА та ААА. Який рівень є загальноприйнятим стандартом для більшості веб-сайтів? /

Магyарázza el a WCAG megfelelőségi szinteket: A, AA és AAA. Melyik szint tekinthető általánosan elfogadott célkitűzésnek a legtöbb weboldal számára?

4. Наведіть приклад критерію успішності для кожного з перших двох принципів (Сприйнятний та Керований). Наприклад, для «Сприйнятного» — надання текстових альтернатив, для «Керованого» — доступність з клавіатури. / Hozzon példát egy-egy sikerességi kritériumra az első két alapelvhez (Érzékelhető és Kezelhető). Például az „Érzékelhető” esetében szöveges alternatívák, a „Kezelhető” esetében billentyűzetes navigálhatóság.

5. Чому стандарт WCAG є технологічно-нейтральним? Як це допомагає йому залишатися актуальним з розвитком нових веб-технологій? / Miért technológiasemleges a WCAG standard? Hogyan segít ez abban, hogy naprakész maradjon az új webes technológiák fejlődésével?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- W3C. Настанови з доступності вебвмісту (WCAG) 2.1. Авторизований переклад W3C. 2021. (українською).
- ПРООН та Міністерство цифрової трансформації України. Моніторинг базової вебдоступності офіційних вебсайтів органів державної влади. 2023. (посилається на ДСТУ EN 301 549:2022, що гармонізований з WCAG) (українською).
- W3C. Web Content Accessibility Guidelines (WCAG) 2.2. 2023. (англійською).
- W3C. Web Content Accessibility Guidelines (WCAG) 2.0. 2008. (угорський переклад доступний на сайті W3C).

Тема 26. Спеціальні інструменти та технології для забезпечення доступності. / Speciális eszközök és technológiák az akadálymentesítéshez

1. Що таке асистивні (допоміжні) технології? Наведіть приклади для різних типів порушень: скрінрідер, екранна лупа, альтернативна клавіатура. / Mit jelent az asszisztív (segítő) technológia? Hozzon példákat különböző fogyatékosági típusokra: képernyőolvasó, képernyőnagyító, alternatív billentyűzet.

2. Поясніть призначення та важливість атрибута alt для тегу . Яким має бути хороший альтернативний текст? / Magyarázza el az tag alt attribútumának célját és fontosságát. Milyen a jó alternatív szöveg?

3. Що означає «доступність з клавіатури»? Опишіть, як користувач повинен мати можливість взаємодіяти з усіма інтерактивними елементами сайту (посилання, кнопки, форми), використовуючи лише клавіатуру. / Mit jelent az „billentyűzetes elérhetőség”? Írja le, hogyan kell a felhasználónak képesnek lennie az összes interaktív elem (hivatkozások, gombok, űrlapok) használatára kizárólag billentyűzettel.

4. Чому достатній коефіцієнт контрастності між текстом та фоном є критично важливим для доступності? Який мінімальний коефіцієнт рекомендується стандартом WCAG AA? / Miért kritikus az elegendő

kontrasztarány a szöveg és a háttér között az elérhetőség érdekében? Mekkora minimális kontrasztarányt ajánl a WCAG AA szint?

5. Як правильно реалізувати доступність для відеоконтенту? Опишіть роль субтитрів (captions) та аудіоопису (audio description). / Hogyan biztosítható a videótartalom akadálymentessége? Ismertesse a feliratok (captions) és az audionarráció (audio description) szerepét.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- ПРООН та Міністерство цифрової трансформації України. Моніторинг базової вебдоступності офіційних вебсайтів органів державної влади. 2023. (містить чек-лист основних технік) (українською).
- W3C. Настанови з доступності вебвмісту (WCAG) 2.1. Авторизований переклад W3C. 2021. (українською).
- The A11Y Project. Checklist. Оновлюється постійно. (англійською). • WCAG2.com. eAccessibility Library. David Berman Communications. Оновлюється постійно. (англійською).

Тема 27. WAI-ARIA: покращення доступності вебзастосунків. / WAI-ARIA: Webes alkalmazások elérhetőségének javítása

1. Що таке WAI-ARIA (Accessible Rich Internet Applications)? Поясніть, яку проблему доступності, що виникає в динамічних веб-додатках, вона вирішує. / Mi az a WAI-ARIA (Accessible Rich Internet Applications)? Ismertesse, milyen hozzáférhetőségi problémát old meg a dinamikus webalkalmazásokban.

2. Назвіть три основні складові ARIA: ролі, властивості та стани. Наведіть по одному прикладу для кожної складової (наприклад, `role="button"`, `aria-label="..."`, `aria-expanded="true"`). / Nevezze meg az ARIA három fő összetevőjét: szerepek, tulajdonságok és állapotok. Hozzon egy-egy példát mindegyikre (például `role="button"`, `aria-label="..."`, `aria-expanded="true"`).

3. Поясніть перше правило використання ARIA: "Якщо ви можете використати нативний HTML-елемент або атрибут, який має вбудовану семантику та поведінку, робіть це". Наведіть приклад. / Ismertesse az ARIA első szabályát: „Ha használható natív HTML-elem vagy attribútum beépített szemantikával és viselkedéssel, akkor azt használja”. Hozzon példát.

4. Уявіть, що ви створюєте кастомний віджет "акордеон" за допомогою `<div>` та JavaScript. Які ARIA-атрибути потрібно додати до заголовків та панелей, щоб зробити його доступним для користувачів скрінрідерів? / Tegyük fel, hogy egy egyedi „harmonika” widgetet készít `<div>` és JavaScript segítségével. Milyen ARIA attribútumokat kell hozzáadni a címsorokhoz és panelekhez, hogy a képernyőolvasót használó felhasználók számára hozzáférhető legyen?

5. Як ARIA-атрибути, такі як `aria-live`, допомагають повідомляти користувачів асистивних технологій про динамічні зміни на сторінці (наприклад, про появу повідомлення про помилку)? / Hogyan segítik az olyan ARIA attribútumok, mint az `aria-live`, a segítő technológiák felhasználóit abban,

якщо повідомляти про динамічні зміни на сторінці (наприклад повідомлення про помилку)?

Рекомендовані джерела інформації / Ajánlott szakirodalom

- MDN Web Docs. WAI-ARIA Basics. Mozilla Foundation. Оновлюється постійно. (англійською, є частковий український переклад).
- W3C. WAI-ARIA Authoring Practices 1.2. Оновлюється постійно. (англійською).
- The A11Y Project. ARIA. Оновлюється постійно. (англійською).

Тема 28. Методи перевірки доступності. / Akadálymentesség ellenőrzésének módszere

1. Порівняйте автоматизоване та ручне тестування доступності. Які типи проблем може виявити кожен з методів, і які є їхні обмеження? / Hasonlítsa össze az automatikus és manuális hozzáférhetőségi tesztelést. Milyen problémákat tud feltárni mindegyik módszer, és milyen korlátaik vannak?
2. Назвіть щонайменше два інструменти для автоматизованої перевірки доступності (наприклад, Lighthouse в Chrome DevTools, axe DevTools, WAVE). Які звіти вони генерують? / Nevezzen meg legalább két eszközt az automatizált akadálymentességi ellenőrzéshez (például Lighthouse a Chrome DevTools-ben, axe DevTools, WAVE). Milyen jelentéseket készítenek?
3. Опишіть основні кроки ручної перевірки доступності сайту за допомогою клавіатури. На що слід звертати увагу (порядок фокусу, видимість фокусу, доступність всіх елементів)? / Ismertesse a weboldal billentyűzetes ellenőrzésének fő lépéseit. Mire kell figyelni (fókusz sorrendje, fókusz láthatósága, az összes elem hozzáférhetősége)?
4. Що таке тестування за допомогою скрінрідера? Чому важливо не просто слухати, що читає програма, а й намагатися виконати ключові завдання на сайті? / Mi az a képernyőolvasós tesztelés? Miért fontos nemcsak meghallgatni, mit mond a program, hanem megpróbálni végrehajtani kulcsfeladatokat a weboldalon?
5. Складіть короткий чек-лист з п'яти ключових пунктів для базової перевірки доступності веб-сторінки перед її публікацією. / Állítson össze egy rövid ellenőrzőlistát öt fő ponttal egy weboldal alapvető hozzáférhetőségének ellenőrzésére publikálás előtt.

Рекомендовані джерела інформації / Ajánlott szakirodalom

- ПРООН та Міністерство цифрової трансформації України. Моніторинг базової вебдоступності офіційних вебсайтів органів державної влади. 2023. (фактично є методикою перевірки) (українською).
- W3C Web Accessibility Initiative (WAI). Easy Checks – A First Review of Web Accessibility. Оновлюється постійно. (англійською).
- The A11Y Project. How-tos. Оновлюється постійно. (англійською).

- WCAG2.com. eAccessibility Library: Testing tools. David Berman Communications. Оновлюється постійно. (англійською).

ТЕСТИ ДЛЯ ПРОВЕДЕННЯ САМОКОНТРОЛЮ

Одним зі способів контролю знань студентів, що дозволяє оцінити успішність навчання та заохотити їх до регулярної роботи протягом семестру, є комп'ютерне тестування. Для самоперевірки підготовлено десять варіантів тестів: у кожному – десять запитань, до яких пропонується по три/чотири відповіді, тільки одна з яких правильна.

Тести по HTML

1. Елемент `<p>` – це тег, який ...

- а) містить наповнення, розташовується з нового рядка, і фактично, визначає новий абзац
- б) створює блок, який, за замовчуванням, розтягується по всій ширині браузера, а наступний після нього елемент переноситься на новий рядок
- в) виводить блок попередньо відформатованого тексту; такий текст відображається зазвичай моноширинним шрифтом і з усіма пробілами між словами
- г) призначений, переважно, для стилізації вкладеного в нього тексту, а саме дозволяє виділити частину інформації всередині інших тегів і встановити для неї свій стиль

2. Елемент `<div>` – це тег, який ...

- а) містить наповнення, розташовується з нового рядка, і фактично, визначає новий абзац
- б) створює блок, який, за замовчуванням, розтягується по всій ширині браузера, а наступний після нього елемент переноситься на новий рядок
- в) виводить блок попередньо відформатованого тексту; такий текст відображається зазвичай моноширинним шрифтом і з усіма пробілами між словами
- г) призначений, переважно, для стилізації вкладеного в нього тексту, а саме дозволяє виділити частину інформації всередині інших тегів і встановити для неї свій стиль

3. Елемент <pre> – це тег, який ...

- а) містить наповнення, розташовується з нового рядка, і фактично, визначає новий абзац
- б) створює блок, який, за замовчуванням, розтягується по всій ширині браузера, а наступний після нього елемент переноситься на новий рядок
- в) виводить блок попередньо відформатованого тексту; такий текст відображається зазвичай моноширинним шрифтом і з усіма пробілами між словами
- г) призначений, переважно, для стилізації вкладеного в нього тексту, а саме дозволяє виділити частину інформації всередині інших тегів і встановити для неї свій стиль

4. Елемент – це тег, який ...

- а) містить наповнення, розташовується з нового рядка, і фактично, визначає новий абзац
- б) створює блок, який, за замовчуванням, розтягується по всій ширині браузера, а наступний після нього елемент переноситься на новий рядок
- в) виводить блок попередньо відформатованого тексту; такий текст відображається зазвичай моноширинним шрифтом і з усіма пробілами між словами
- г) призначений, переважно, для стилізації вкладеного в нього тексту, а саме дозволяє виділити частину інформації всередині інших тегів і встановити для неї свій стиль

5. Для створення маркованих списків використовуються теги ...

- а) ,
- б) ,
- в) ,
- г) , ,

6. Для створення нумерованих списків використовуються теги ...

- а) ,
- б) ,

в) ,

г) , ,

7. Значення атрибута посилань (<a ...> ...) target відповідає за ...

а) відкриття html-документа в новому вікні або вкладці браузера

б) відкриття html-документа в тому ж фреймі (або вікні)

в) відкриття документа у батьківському фреймі, якщо посилання розташовано у внутрішньому фреймі

г) відкриття html-документа на все вікно браузера

8. Значення атрибута посилань (<a ...> ...) target відповідає за ...

а) відкриття html-документа в новому вікні або вкладці браузера

б) відкриття html-документа в тому ж фреймі (або вікні)

в) відкриття документа у батьківському фреймі, якщо посилання розташовано у внутрішньому фреймі

г) відкриття html-документа на все вікно браузера

9. Елемент footer ...

а) являє собою вступний елемент, що передує основному вмісту, де можуть бути заголовки, елементи навігації або будь-які інші допоміжні елементи, наприклад, логотип, форма пошуку і т.п.

б) задає нижній колонтитул документу або розділу; зазвичай, містить інформацію про те, хто автор контенту на веб-сторінці, копірайт, дату публікації, блок посилань на схожі ресурси і т.д.

в) об'єднує пов'язані між собою частини htmlдокумента, виконуючи їх групування

г) містить набір посилань, як правило, у вигляді нумерованого списку, що є навігацією по сайту

10. Елемент header ...

- а) являє собою вступний елемент, що передує основному вмісту, де можуть бути заголовки, елементи навігації або будь-які інші допоміжні елементи, наприклад, логотип, форма пошуку і т.п.
- б) задає нижній колонтитул документу або розділу; зазвичай, містить інформацію про те, хто автор контенту на веб-сторінці, копірайт, дату публікації, блок посилань на схожі ресурси і т.д.
- в) об'єднує пов'язані між собою частини htmlдокумента, виконуючи їх групування
- г) містить набір посилань, як правило, у вигляді нумерованого списку, що є навігацією по сайту

11. Значення radio атрибуту type поля input для вводу даних HTML-форми задає ...

- а) радіокнопку або перемикач (з групи радіокнопок можна вибрати тільки одну)
- б) елемент прапорець, який може знаходитися у відміченому або невідміченому стані
- в) кнопку відправки даних з форми
- г) немає такого значення атрибуту type

12. Значення checkbox атрибуту type поля input для вводу даних HTML-форми задає ...

- а) радіокнопку або перемикач (з групи радіокнопок можна вибрати тільки одну)
- б) елемент прапорець, який може знаходитися у відміченому або невідміченому стані
- в) кнопку відправки даних з форми
- г) немає такого значення атрибуту type

13. Крім елемента input у різних модифікаціях на формі також можна використовувати такі елементи:

- а) button, textbox, label, select
- б) button, textarea, label, listbox

в) button, textbox, label, listbox

г) button, textarea, label, select

14. Атрибут pattern елемента input (type=text) задає ...

а) напрямок тексту

б) максимально допустиму кількість символів у текстовому полі

в) шаблон, якому повинен відповідати текст, що вводиться

г) текст-підказку для вводу

15. Атрибут placeholder елемента input (type=text) задає ...

а) напрямок тексту

б) максимально допустиму кількість символів у текстовому полі

в) шаблон, якому повинен відповідати текст, що вводиться

г) текст-підказку для вводу

16. Які значення за допомогою нижченаведеного input'а можна відправити на сервер? <input type="number" list="priceList" step="10" min="30" max="70" value="50" id="price" name="price"/>

а) 10, 20, 30

б) 30, 40, 50, 60, 70

в) 30, 40, 50

г) будь-які значення

17. Які значення за допомогою нижченаведеного input'а можна відправити на сервер? <input type="text" placeholder="+X-XXX-XXX-XXX" pattern="+\d-\d{3}-\d{3}-\d{4}" id="phone" name="phone"/>

а) +X-XXX-XXX-XXX, +X-XXX-XXX-XXXX, де X – цифра

б) +X-XXX-XXX-XXX, де X –цифра

в) будь-які значення

г) +X-XXX-XXX-XXXX, де X –цифра

18. Які значення за допомогою нижченаведених input'ів можна відправити на сервер? <input type="radio" value="HTML" checked name="tech"/>HTML <input type="radio" value="JS" name="tech2"/>JavaScript

- а) тільки HTML
- б) HTML, JS
- в) HTML, JavaScript
- г) тільки JS

19. Які значення за допомогою нижченаведених input'ів можна відправити на сервер? `<input type="checkbox" value="Csharp" name="tech"/>C#` `<input type="checkbox" value="Java" name="tech"/>Java`

- а) . тільки Csharp
- б) C#, Java
- в) Csharp, Java
- г) тільки Java

20. Значення CSS-властивості background-color для визначення кольору фону HTML-елемента можна задати ...

- а) значеннями #RRGGBB, де RR, GG, BB – шістнадцяткові числа
- б) назвою кольору англійською мовою для певної сукупності найпоширеніших кольорів
- в) значеннями rgb(X, X, X), rgba(X, X, X, D), де X є 0..255, D є 0..1
- г) будь-якими варіантами, переліченими у всіх відповідях

21. Найвищий пріоритет мають ...

- а) вбудовані стилі (inline-стилі)
- б) стилі, визначені в елементі style в тому ж файлі
- в) стилі з підключеного зовнішнього файлу
- г) стилі, визначені в елементі script

22. CSS-стилі HTML-елементів можна задати з використанням ...

- а) селектора класу, ідентифікатора елемента, об'єкта класу
- б) селектора тегу, універсального селектора, селектора класу, атрибутів HTML-тегів
- в) селектора тегу, універсального селектора, селектора класу, ідентифікатора елемента

г) поля класу, універсального селектора, селектора класу, ідентифікатора елемента

23. Задати стилі для всіх елементів span, які розміщені після div на одному рівні вкладеності, можна, використовуючи селектор ...

- а) div span
- б) div>span
- в) div~span
- г) div+span

24. Задати стилі для елементів span, які розміщені в елементі div на першому рівні вкладеності, можна, використовуючи селектор ...

- а) div span
- б) div>span
- в) div~span
- г) div+span

25. Задати стилі для елемента span, який розміщений безпосередньо після div, можна, використовуючи селектор ...

- а) div span
- б) div>span
- в) div~span
- г) div+span

26. Задати стилі для посилання в момент переходу по ньому можна з використанням псевдокласу ...

- а) :active
- б) :focus
- в) :visited
- г) :hover

27. Задати стилі для елемента в момент наведення на нього курсору миші можна з використанням псевдокласу ...

- а) :active
- б) :focus

в) :visited

г) :hover

28. Для додавання повідомлення після HTML-елемента використовується псевдоелемент ...

а) ::before

б) ::after

в) ::first-line

г) ::selection

29. Для застосування стилів до виділеного користувачем фрагмента тексту використовується псевдоелемент ...

а) ::before

б) ::after

в) ::first-line

г) ::selection

30. Наслідування стилів в CSS застосовується ...

а) до всіх елементів

б) до всіх елементів, крім властивостей: margin, padding, border

в) до всіх елементів, крім властивостей: margin, color, border

г) до всіх елементів, крім властивостей: position, padding, border

31. Який колір для тексту `<div id="myId"><span class="myClass">Текст</span></div>` зададуть такі стилі: `#myId span {color: green;} /* зелений колір тексту / div .myClass {color: red;} / червоний колір тексту / div span {color: gray;} / сірий колір тексту */`

а) зелений

б) червоний

в) сірий

г) чорний

32. Скасувати пріоритет стилів можна за допомогою ключового слова...

а) importantly

- б) fatefully
- в) important
- г) big

33. Розмір шрифту можна задати за допомогою ...

- а) 7 ключових слів (small, large, medium, ...)
- б) задання розміру в пікселях або відсотках
- в) задання розміру в одиницях em
- г) всіма переліченими способами

34. Задати для заголовків h2 горизонтальне зміщення тіні тексту на 5 пікселів вправо, вертикальне зміщення вниз – 4 пікселі, ступінь розмитості – 3 пікселі та колір для тіні #999 можна за допомогою стилю ...

- а) `h2 { text-shadow: 5px 4px 3px #999; }`
- б) `h2 { text-shadow: 4px 5px 3px #999; }` в) `h2 { text-shadow: 3px 4px 5px #999; }`
- г) `h2 { text-shadow: 5px 3px 4px #999; }`

35. CSS-властивості margin та padding елементів можуть бути ...

- а) тільки додатні значення
- б) і додатні, і від'ємні значення
- в) для властивості padding тільки додатні, для margin допускаються і від'ємні
- г) для властивості margin тільки додатні, для padding допускаються і від'ємні

36. Значення solid властивості border-style задає границю у вигляді ...

- а) суцільної лінії
- б) послідовності точок
- в) штрих-лінії
- г) двох паралельних ліній

37. Значення double властивості border-style задає границю у вигляді...

- а) суцільної лінії
- б) послідовності точок
- в) штрих-лінії
- г) двох паралельних ліній

38. Значення `hidden` властивості `overflow` дозволяє налаштувати прокрутку блоку наступним чином:

- а) якщо контент виходить за межі блоку, то створюється прокрутка; в інших випадках смуги прокрутки не відображаються
- б) відображається тільки видима частина контенту; контент, який виходить за межі блоку, не відображається, а смуги прокрутки не створюються
- в) в блоці відображаються смуги прокрутки, навіть якщо весь контент поміщається в межах блоку, і смуг прокрутки не потрібно
- г) контент відображається, навіть якщо він виходить за межі блоку, смуги прокрутки при цьому не створюються

39. Значення `visible` властивості `overflow` дозволяє налаштувати прокрутку блоку наступним чином:

- а) якщо контент виходить за межі блоку, то створюється прокрутка; в інших випадках смуги прокрутки не відображаються
- б) відображається тільки видима частина контенту; контент, який виходить за межі блоку не відображається, а смуги прокрутки не створюються
- в) в блоці відображаються смуги прокрутки, навіть якщо весь контент поміщається в межах блоку, і смуг прокрутки не потрібно
- г) контент відображається, навіть якщо він виходить за межі блоку, смуги прокрутки при цьому не створюються

40. Який колір для тексту `<div id="myId"><span class="myClass">Текст</span></div>` зададуть такі стилі: `#myId {color: green;}` /* зелений колір тексту / `div.myClass {color: red;}` / червоний колір тексту / `div span{color: gray;}` / сірий колір тексту */

- а) зелений
- б) червоний
- в) сірий
- г) чорний

Тести по JavaScript

1. Коректні імена змінних у JavaScript:

- a) \$ds, s_h, s1, \$gs_
- б) asd, vasia, with, fgh
- в) r#h, df, as, er
- г) \$ds, s_%h, s1, gs

2. Типи даних JavaScript:

- a) float, double, int, string, object, char, boolean, null, undefined
- б) float, number, string, object, char, boolean, null, undefined
- в) number, string, object, boolean, null, undefined
- г) float, double, int, string, char, boolean, null

3. Після виконання коду `var isAlive; console.log(isAlive);` в консоль буде виведено ...

- a) не буде нічого
- б) null
- в) undefined
- г) 0

4. Після виконання коду `x="45"; var z=x + 5;` змінна z міститиме значення ...

- a) 50
- б) 455
- в) undefined
- г) null

**5. Після виконання нижченаведеного коду в консоль буде записано ...
`var income = 45.8; console.log(typeof income);`**

- a) float
- б) double
- в) number
- г) 45.8

**6. Після виконання нижченаведеного коду в консоль буде записано ...
`var income = 100;`
`var strIncome = "100";`**

```
var result = income == strIncome;  
console.log(result);
```

- а) 100
- б) "100"
- в) true
- г) false

7. Після виконання нижченаведеного коду в консоль буде записано ...

```
var car = {}; if (car) console.log(false); else console.log(true);
```

- а) true
- б) false

8. Результатом виконання нижченаведеного коду буде ... let z = 20; { let

```
z = 30; console.log(z); } console.log(z);
```

- а) помилка, оскільки змінна Z два рази оголошена
- б) запис в консоль 30 та 20

9. Результатом виконання нижченаведеного коду буде ... var z = 20; {

```
var z = 30; console.log(z); } console.log(z);
```

- а) запис в консоль 30 та 30
- б) запис в консоль 30 та 20

10. Після виконання нижченаведеного коду в консоль буде записано ...

```
function bar() { foo = "25"; } bar(); console.log(foo);
```

- а) 25
- б) undefined
- в) буде помилка

11. Після виконання нижченаведеного коду в консоль буде записано ...

```
function outer() { let x = 5; function inner() { x++; console.log(x); }; return  
inner; } let fn = outer(); fn();
```

- а) 6
- б) undefined
- в) нічого не буде, бо буде помилка

12. Після виконання нижченаведеного коду в консоль буде записано ...
- ```
function display(){ console.log("0"); display = function(){ console.log("1"); } }
var displayMessage = display; display(); display(); displayMessage();
displayMessage();
```
- a) 1 1 1 1  
б) 0 1 1 1  
в) 0 1 0 0  
г) 1 1 0 0
13. Після виконання нижченаведеного коду в консоль буде записано ...
- ```
console.log(foo); var foo = "Tom";
```
- a) undefined
б) Tom
в) помилка на першому рядку коду ReferenceError: foo is not defined
14. Після виконання нижченаведеного коду в консоль буде записано ...
- ```
function change(x) { x = 2 * x; console.log(x); }
var n = 10; console.log(n);
change(n);
console.log(n);
```
- a) 10 10 10  
б) 10 20 10  
в) 10 20 20
15. Після виконання нижченаведеного коду в консоль буде записано ...
- ```
function change(user){ user.name = "Tom"; }  
var bob={ name: "Bob" };  
console.log(bob.name);  
change(bob);  
console.log(bob.name);
```
- a) Bob Tom
б) Bob Bob

16. Після виконання нижченаведеного коду в консоль буде записано ...

```
function change(array) { array[0] = 8; } var numbers = [1, 2, 3];  
change(numbers);  
console.log(numbers);
```

а) [8, 2, 3]

б) [1, 2, 3]

17. Для отримання значення року столиці об'єкту країни

```
var country={ name: "Германия", capital:{ name:"Берлин", year:1237}};  
можна використати рядок коду ...
```

а) country["capital"]["year"]

б) country.capital.year

в) country.capital["year"]

г) відповіді “а” та “б” правильні д) всі відповіді правильні

18. У мові програмування JavaScript ...

а) конструктор – це звичайна функція

б) конструктор – це функція, в якій можна встановити властивості та методи з використанням this

в) немає конструкторів

19. Властивість prototype дає можливість ...

а) визначити спільні властивості для об'єктів одного типу

б) створити статичний об'єкт

в) створити екземпляр класу

20. За замовчуванням, усі властивості об'єкту ...

а) загальнодоступні

б) закриті

21. Після виконання нижченаведеного коду в консоль буде виведено...

```
function User (name) { this.name = name; var _age = 1; }  
var tom = new User("Том");  
console.log(tom._age);
```

а) undefined

б) 1

в) помилка буде

г) null

22. Чи дозволяє JS принцип успадкування?

а) так

б) ні

23. Чи є в JS можливість створення класів?

а) так

б) ні

24. Оператор super дає можливість

а) викликати функціонал конструктора батьківського класу

б) позначити клас із суперможливостями

в) створити помилку, оскільки такого оператора немає

25. Конструктор класу може мати такий синтаксис:

а) `class Person{ constructor(name, age){ this.name = name; this.age = age; }`

б) `class Person{ Person(name, age){ this.name = name; this.age = age; }`

в) немає конструкторів у класах г) немає ні класів, ні конструкторів

26. У регулярних виразах символ /D відповідає ...

а) алфавітно-цифровому символу

б) не алфавітно-цифровому символу

в) десятковій цифрі

г) символу, який не є десятковою цифрою

27. У регулярних виразах символ /W відповідає ...

а) алфавітно-цифровому символу

б) не алфавітно-цифровому символу

в) десятковій цифрі

г) символу, який не є десятковою цифрою

28. Після виконання нижченаведеного коду nodes міститиме ... `var`

`articleDiv = document.querySelector("div.article"); var nodes = articleDiv.childNodes;`

- а) текст, атрибути та самі теги дочірніх елементів елемента div.article
- б) дочірні елементи елемента div.article
- в) атрибути та самі теги дочірніх елементів div.article

29. Наступний код дозволяє вивести ...

```
function getChildren(elem){
  for(var i in elem.childNodes){
    if(elem.childNodes[i].nodeType===1){
      console.log(elem.childNodes[i].tagName); getChildren(elem.childNodes[i]); } }
}

var root = document.documentElement;
console.log(root.tagName);
getChildren(root);
```

- а) всі елементи html-документа
- б) всі атрибути
- в) вмістиме кожного елемента сторінки
- г) все перераховане

30. Подія mousedown виникає при ...

- а) знаходженні покажчика миші на елементі, коли кнопка миші знаходиться в натиснутому стані
- б) знаходженні покажчика миші на елементі під час відпускання кнопки миші
- в) входженні покажчика миші в межі елемента
- г) проходженні покажчика миші над елементом

31. Подія mouseup виникає при ...

- а) знаходженні покажчика миші на елементі, коли кнопка миші знаходиться в натиснутому стані
- б) знаходженні покажчика миші на елементі під час відпускання кнопки миші
- в) входженні покажчика миші в межі елемента
- г) проходженні покажчика миші над елементом

32. Подія mouseover виникає при ...

- а) знаходженні покажчика миші на елементі, коли кнопка миші знаходиться в натиснутому стані

- б) знаходженні покажчика миші на елементі під час відпускання кнопки миші
- в) входженні покажчика миші в межі елемента
- г) проходженні покажчика миші над елементом

33. Подія `mousemove` виникає при ...

- а) знаходженні покажчика миші на елементі, коли кнопка миші знаходиться в натиснутому стані
- б) знаходженні покажчика миші на елементі під час відпускання кнопки миші
- в) входженні покажчика миші в межі елемента
- г) проходженні покажчика миші над елементом

34. Правильний запис `json`-об'єкта:

- а) { name: "Tom", married: true, age: 30 }
- б) { "name": "Tom", "married": true, "age": 30 }
- в) { name: "Tom", married: true, "my age": 30 }
- г) всі перераховані

35. За замовчуванням, час життя `cookie` триває до ...

- а) 24 годин
- б) до закриття браузера
- в) до зачищення вручну куків
- г) немає часу, за замовчуванням, обов'язково треба визначати

36. Яка з типів браузерної пам'яті передається по `http`?

- а) `session storage`
- б) `local storage`
- в) `cookie`
- г) всі перераховані

37. `Session storage` ...

- а) реалізує тимчасове зберігання інформації, яка видаляється після закриття браузера
- б) являє собою сховище даних на постійній основі
- в) нічого не означає, оскільки немає такого типу пам'яті

38. `Local storage` ...

- а) реалізує тимчасове зберігання інформації, яка видаляється після закриття браузера
- б) являє собою сховище даних на постійній основі
- в) нічого не означає, оскільки немає такого типу пам'яті

Тести по PHP

1. Найпоширенішими веб-сервером та СКБД, які працюють у зв'язці з мовою PHP, є ...

- а) IIS, MsSQL Server
- б) Nginx, PostgreSQL
- в) Apache, MySQL
- г) Nginx, MsSQL Server

2. Відправлені дані з HTML-форми на мові PHP можна отримати з глобальних масивів ...

- а) \$_GET, \$_POST
- б) \$_GET, \$_ECHO
- в) \$_POST, \$_ECHO
- г) \$_ECHO, \$_FORM

3. Оператором конкатенації (об'єднання) рядків на мові PHP є ...

- а) ++
- б) +
- в) .
- г) concat()

4. Змінними на PHP є ...

- а) \$_for, \$&gh, \$gh
- б) \$_4for, \$h_dsg3, \$while
- в) \$4for, \$h_dsg3, \$ghf
- г) _gh4, \$for, \$f3g4

5. Нижченаведений php-код виведе на веб-сторінку ...
`$i=4; echo "квадрат числа $i дорівнює ". $i * $i;`

- а) квадрат числа 4 дорівнює 16

- б) квадрат числа 4 дорівнює 44
- в) квадрат числа \$i\$ дорівнює 16
- г) квадрат числа \$i\$ дорівнює 44

6. Мова PHP підтримує такі типи даних:

- а) boolean, integer, float, string, array, object, resource, NULL
- б) boolean, integer, double, float, string, array, object, NULL
- в) boolean, int, double, string, array, object, resource, NULL
- г) boolean, integer, double, string, array, object, resource, NULL

7. Для передачі змінної а за посиланням потрібно записати ...

- а) function get(ref \$a){ тіло функції }
- б) function get(&\$a){ тіло функції }
- в) function get(a){ тіло функції }
- г) function get(out \$a){ тіло функції }

8. Для доступу до змінної, яка визначена у деякій функції, з будь-якого місця php-коду, її потрібно визначити з ключовим словом ...

- а) static
- б) public
- в) global
- г) var

9. Для підключення php-файлу до іншого php-файлу використовують...

- а) include
- б) require
- в) include_once
- г) всі перелічені

10. Для підключення php-файлу до іншого php-файлу з виведенням помилки у разі відсутності файлу та забезпеченням тільки разового підключення використовується ...

- а) include
- б) require_once

в) include_once

г) всі перелічені

11. Для виводу розміру масиву \$a в php використовується ...

а) count(\$a), sizeof(\$a)

б) length(\$a), sizeof(\$a)

в) count(\$a), length(\$a)

г) count(\$a), length(\$a), sizeof(\$a)

12. Для сортування елементів масиву рядків по алфавіту або масиву чисел за зростанням в php використовується ...

а) asort()

б) arsort()

в) ksort()

г) krsort()

13. Для сортування елементів масиву чисел за спаданням в php використовується ...

а) asort()

б) arsort()

в) ksort()

г) krsort()

14. Для сортування елементів масиву за ключами в php використовується ...

а) asort()

б) arsort()

в) ksort()

г) kksort()

15. Для отримання cookie в php можна використовувати глобальний асоціативний масив ...

а) _COOKIE

б) COOKIE

в) \$_COOKIE_START

г) \$_COOKIE

16. Коректне визначення cookie:

а) setcookie("language", "китайський", time() + 3600)

б) setcookie("language", "китайський")

в) setcookie("language", "китайський", time() + 3600, "/")

г) всі перелічені

17. Вираз setcookie("language","китайський", time() + 3600) визначає cookie на ...

а) годину

б) хвилину

в) 36 секунд

г) 36 хвилин

18. Для запуску сесії потрібно викликати функцію ...

а) session_enter()

б) session_start()

в) session_push()

г) session_begin()

19. Для запису або читання змінної сесії використовується глобальний асоціативний масив ...

а) _SESSION

б) \$_SESSION

в) \$SESSION

г) \$_SESSION_START

20. Параметри-значення набору в запиті get відокремлюються знаком...

а) &

б) \$

в) +

г) ;

21. Результатом виконання коду `echo 11 . 22`; буде вивід на веб-сторінці...

- а) 33
- б) 1122
- в) 11.22
- г) 11

22. Для знищення змінної `$a` використовується оператор ...

- а) `unset($a)`
- б) `isset($a)`
- в) `delete($a)`
- г) `drop($a)`

23. Для перевірки ініціалізованості змінної `$a` застосовується оператор...

- а) `unset($a)`
- б) `isset($a)`
- в) `isvariable($a)`
- г) `isvar($a)`

24. Результатом виконання коду `$a=NULL; echo $a`; на веб-сторінці буде ...

- а) вивідення 0
- б) відсутність якогось виведення
- в) виведення NULL
- г) помилка

25. Результатом виконання коду `$a=4;echo -$a--`; буде вивід на веб-сторінці числа ...

- а) 4
- б) 3
- в) -3
- г) -4

26. Результатом виконання нижченаведеного коду буде вивід на веб-сторінці...

\$a = 10; \$b = 5; echo "\$a + \$b " . '\$a + \$b';

- а) 10 + 5 10 + 5
- б) \$a + \$b \$a + \$b
- в) 15 \$a + \$b
- г) 10 + 5 \$a + \$b

27. Для визначення деякої константи NUMBER потрібно записати ...

- а) var NUMBER=22
- б) final NUMBER=22
- в) const NUMBER=22
- г) define("NUMBER", 22)

28. Результатом виконання нижченаведеного коду буде вивід на веб-сторінці...

\$a = 10=="10"; \$b = 10!="10"; echo \$a . \$b;

- а) порожніх символів
- б) 11
- в) 10
- г) 01

29. Результатом виконання нижченаведеного коду буде вивід на веб-сторінці...

echo 7 > 6 && 8 > 8 or 9 > 8; echo 1 || 0 xor 0;

- а) порожніх символів
- б) 11
- в) помилки
- г) 1

30. Результатом виконання нижченаведеного коду буде вивід на веб-сторінці ...

\$n=10;

```
for($i=0;$i<$n;$i++,$n--);
```

```
echo $i;
```

a) 5

б) 0

в) 0123456789

г) 01234

TESZTFELADATOK ÖNELLENŐRZÉSHEZ

A hallgatók tudásának ellenőrzésének egyik módja, amely lehetővé teszi a tanulmányi eredményesség értékelését és ösztönzi őket a rendszeres munkára a szemeszter során, a számítógépes tesztelés. ÖnelLENŐRZÉS céljából tíz tesztváltozat készült: mindegyikben tíz kérdés található, amelyekhez három/négy válaszlehetőség tartozik, és közülük csak egy helyes.

HTML és CSS3 tesztek

1. A **<p>** elem olyan címke, amely ...

- a) tartalmaz szöveget, új sorban kezdődik, és tulajdonképpen egy új bekezdést jelöl;
- b) olyan blokkot hoz létre, amely alapértelmezésben a böngésző teljes szélességére nyúlik, és az utána következő elem új sorba kerül;
- c) előre formázott szövegblokkot jelenít meg; az ilyen szöveg általában fix szélességű betűtípussal és az összes szóköz megtartásával jelenik meg;
- d) főként a benne található szövegrészlet stílusozására szolgál, vagyis lehetővé teszi a más tágékben lévő információ egy részének kiemelését és saját stílusának beállítását.

2. A **<div>** elem olyan címke, amely ...

- a) tartalmaz szöveget, új sorban kezdődik, és tulajdonképpen egy új bekezdést jelöl;
- b) olyan blokkot hoz létre, amely alapértelmezésben a böngésző teljes szélességére nyúlik, és az utána következő elem új sorba kerül;
- c) előre formázott szövegblokkot jelenít meg; az ilyen szöveg általában fix szélességű betűtípussal és az összes szóköz megtartásával jelenik meg;
- d) főként a benne található szövegrészlet stílusozására szolgál, vagyis lehetővé teszi a más tágékben lévő információ egy részének kiemelését és saját stílusának beállítását.

3. A **<pre>** elem olyan címke, amely ...

- a) tartalmaz szöveget, új sorban kezdődik, és tulajdonképpen egy új bekezdést jelöl;
- b) olyan blokkot hoz létre, amely alapértelmezésben a böngésző teljes szélességére nyúlik, és az utána következő elem új sorba kerül;
- c) előre formázott szövegblokkot jelenít meg; az ilyen szöveg általában fix szélességű betűtípussal és az összes szóköz megtartásával jelenik meg;
- d) főként a benne található szövegrészlet stílusozására szolgál, vagyis lehetővé teszi a más tágékben lévő információ egy részének kiemelését és saját stílusának beállítását.

4. A **** elem olyan címke, amely ...

- a) tartalmaz szöveget, új sorban kezdődik, és tulajdonképpen egy új bekezdést jelöl;
- b) olyan blokkot hoz létre, amely alapértelmezésben a böngésző teljes szélességére nyúlik, és az utána következő elem új sorba kerül;
- c) előre formázott szövegblokkot jelenít meg; az ilyen szöveg általában fix szélességű betűtípussal és az összes szóköz megtartásával jelenik meg;
- d) főként a benne található szövegrészlet stílusozására szolgál, vagyis lehetővé teszi a más tágékben lévő információ egy részének kiemelését és saját stílusának beállítását.

5. **A számozatlan (márkázatlan) listák létrehozásához az alábbi tagek használhatók ...**
- a) ,
 - b) ,
 - c) ,
 - d) , ,
6. **A számozott listák létrehozásához az alábbi tagek használhatók ...**
- a) ,
 - b) ,
 - c) ,
 - d) , ,
7. **Az <a ...> hivatkozás target attribútumának _blank értéke ...**
- a) új böngészőablakban vagy fülön nyitja meg a dokumentumot;
 - b) ugyanabban a keretben (vagy ablakban) nyitja meg a dokumentumot;
 - c) a szülő keretben nyitja meg a dokumentumot, ha a link egy belső keretben helyezkedik el;
 - d) a dokumentumot teljes böngészőablakra nagyítja.
8. **Az <a ...> hivatkozás target attribútumának _self értéke ...**
- a) új böngészőablakban vagy fülön nyitja meg a dokumentumot;
 - b) ugyanabban a keretben (vagy ablakban) nyitja meg a dokumentumot;
 - c) a szülő keretben nyitja meg a dokumentumot, ha a link egy belső keretben helyezkedik el;
 - d) a dokumentumot teljes böngészőablakra nagyítja.
9. **A <footer> elem ...**
- a) bevezető elem, amely a fő tartalom előtt áll, és tartalmazhat címsorokat, navigációs elemeket vagy más segédelemeket, például logót, keresőmezőt stb.;
 - b) a dokumentum vagy egy rész láblécét határozza meg; általában információkat tartalmaz a szerzőről, szerzői jogról, publikálási dátumról, hasonló forrásokra mutató hivatkozásokról stb.;
 - c) összefüggő részeket csoportosít a HTML-dokumentumban;
 - d) hivatkozások halmazát tartalmazza, általában számozatlan lista formájában, és a webhely navigációját valósítja meg.
10. **A <header> elem ...**
- a) bevezető elem, amely a fő tartalom előtt áll, és tartalmazhat címsorokat, navigációs elemeket vagy más segédelemeket, például logót, keresőmezőt stb.;
 - b) a dokumentum vagy egy rész láblécét határozza meg; általában információkat tartalmaz a szerzőről, szerzői jogról, publikálási dátumról, hasonló forrásokra mutató hivatkozásokról stb.;
 - c) összefüggő részeket csoportosít a HTML-dokumentumban;
 - d) hivatkozások halmazát tartalmazza, általában számozatlan lista formájában, és a webhely navigációját valósítja meg.
11. **Az input mező type attribútumának radio értéke ...**
- a) rádiógombot hoz létre (egy rádiógomb-csoportból egyszerre csak egy választható);
 - b) jelölőnégyzetet hoz létre, amely bejelölt vagy nem bejelölt állapotú lehet;
 - c) egy űrlap adatainak elküldésére szolgáló gombot hoz létre;
 - d) ilyen type-érték nincs.
12. **Az input mező type attribútumának checkbox értéke ...**
- a) rádiógombot hoz létre (egy rádiógomb-csoportból egyszerre csak egy választható);
 - b) jelölőnégyzetet hoz létre, amely bejelölt vagy nem bejelölt állapotú lehet;

- c) egy űrlap adatainak elküldésére szolgáló gombot hoz létre;
- d) ilyen type-érték nincs.

13. Az input elem különböző típusain kívül az űrlapon az alábbi elemek is használhatók:

- a) <button>, textbox, <label>, <select>
- b) <button>, <textarea>, <label>, listbox
- c) <button>, textbox, <label>, listbox
- d) <button>, <textarea>, <label>, <select>

14. Az input (type="text") elem pattern attribútuma ...

- a) a beírt szöveg irányát határozza meg;
- b) a megengedett maximális karakterek számát adja meg a szövegmezőben;
- c) azt a mintát adja meg, amelynek a beírt szövegnek meg kell felelnie;
- d) a mezőben megjelenő súgó szöveget határozza meg.

15. Az input (type="text") elem placeholder attribútuma ...

- a) a beírt szöveg irányát határozza meg;
- b) a megengedett maximális karakterek számát adja meg a szövegmezőben;
- c) azt a mintát adja meg, amelynek a beírt szövegnek meg kell felelnie;
- d) a mezőben megjelenő súgó szöveget határozza meg.

16. Milyen értékeket lehet a következő bemeneti mezővel elküldeni a szervernek?

```
<input type="number" list="priceList" step="10" min="30" max="70" value="50" id="price" name="price"/>
```

- a) 10, 20, 30
- b) 30, 40, 50, 60, 70
- c) 30, 40, 50
- d) bármilyen érték

17. Milyen értékeket lehet a következő mezővel elküldeni a szervernek?

```
<input type="text" placeholder="+X-XXX-XXX-XXX" pattern="+\d-\d{3}-\d{3}-\d{4}" id="phone" name="phone"/>
```

- a) +X-XXX-XXX-XXX, +X-XXX-XXX-XXXX, ahol X számjegy
- b) +X-XXX-XXX-XXX, ahol X számjegy
- c) bármilyen érték
- d) +X-XXX-XXX-XXXX, ahol X számjegy

18. Milyen értékeket lehet a következő inputokkal elküldeni a szervernek?

```
<input type="radio" value="HTML" checked name="tech"/>HTML  
<input type="radio" value="JS" name="tech2"/>JavaScript
```

- a) csak HTML
- b) HTML, JS
- c) HTML, JavaScript
- d) csak JS

19. Milyen értékeket lehet a következő inputokkal elküldeni a szervernek?

```
<input type="checkbox" value="Csharp" name="tech"/>C#  
<input type="checkbox" value="Java" name="tech"/>Java
```

- a) csak Csharp
- b) C#, Java

- c) Csharp, Java
- d) csak Java

20. **A háttérszín (background-color) CSS-tulajdonság értékét megadhatjuk ...**

- a) #RRGGBB formátumban, ahol RR, GG, BB hexadecimális számok;
- b) bizonyos meghatározott színek angol nevével;
- c) rgb(X, X, X) vagy rgba(X, X, X, D) formátumban, ahol X 0..255, D pedig 0..1 közötti;
- d) bármelyik felsorolt módon.

21. **A legmagasabb prioritású CSS-stílusok ...**

- a) a beágyazott (inline) stílusok;
- b) az ugyanabban a fájlban lévő <style> elemben definiált stílusok;
- c) a külső stíluslapból (linkelt fájlból) származó stílusok;
- d) a <script> elemben definiált stílusok.

22. **A HTML-elemekre vonatkozó CSS-stílusokat megadhatjuk ...**

- a) osztályszelektorral, elem-azonosítóval (id), osztály-objektummal;
- b) tag-szelektorral, univerzális szelektorral, osztályszelektorral, HTML-tulajdonságokra vonatkozó szelektorral;
- c) tag-szelektorral, univerzális szelektorral, osztályszelektorral, elem-azonosítóval;
- d) az osztály mezőjével, univerzális szelektorral, osztályszelektorral, elem-azonosítóval.

23. **A CSS-ben az összes elemre, amelyek ugyanazon hierarchiaszinten a <div> után helyezkednek el, a következő szelektorral adhatunk stílust:**

- a) div span
- b) div > span
- c) div~span
- d) div+span

24. **A <div> elemben közvetlenül (első szinten) elhelyezkedő elemek stílusának megadásához használható szelektor:**

- a) div span
- b) div > span
- c) div~span
- d) div+span

25. **A <div> elemet közvetlenül követő elem stílusát megadhatjuk a következő szelektorral:**

- a) div span
- b) div > span
- c) div~span
- d) div+span

26. **Egy hivatkozásra kattintás pillanatában alkalmazott stílus megadásához a következő pszeudoosztály használható:**

- a) :active
- b) :focus
- c) :visited
- d) :hover

27. Egy elemre történő egérmutató föléhúzáskor alkalmazott stílus megadásához a következő pszeudoosztály használható:

- a) :active
- b) :focus
- c) :visited
- d) :hover

28. HTML-elem mögé szöveget vagy tartalmat hozzáfűzni a következő pszeudoelem segítségével lehet:

- a) ::before
- b) ::after
- c) ::first-line
- d) ::selection

29. A felhasználó által kijelölt (kijelöléssel kiválasztott) szövegrész stílusának megadásához a következő pszeudoelem használható:

- a) ::before
- b) ::after
- c) ::first-line
- d) ::selection

30. A CSS-stílusok öröklődése alkalmazható ...

- a) minden elemre;
- b) minden elemre, kivéve a margin, padding és border tulajdonságokat;
- c) minden elemre, kivéve a margin, color és border tulajdonságokat;
- d) minden elemre, kivéve a position, padding és border tulajdonságokat.

31. Milyen szint fog kapni a következő HTML-kódrészben a szöveg?

```
<div id="myId"><span class="myClass">Szöveg</span></div>
```

A stílusok:

```
#myId span {color: green;} /* zöld szöveg */  
div .myClass {color: red;} /* piros szöveg */  
div span {color: gray;} /* szürke szöveg */
```

- a) zöld
- b) piros
- c) szürke
- d) fekete

32. Egy stílus prioritásának felülírásához használható kulcsszó ...

- a) importantly
- b) fatefully
- c) important
- d) big

33. A betűméretet megadhatjuk ...

- a) hét kulcsszóval (small, large, medium, stb.);
- b) pixelben vagy százalékban kifejezve;

- c) em egységekben kifejezve;
- d) mindegyik felsorolt módon.

34. Az alábbi stílus közül melyik adja meg a h2 címsorok szövegének árnyékát 5 pixel jobbra, 4 pixel lefelé, 3 pixel elmosódással és #999 színnel?

- a) h2 { text-shadow: 5px 4px 3px #999; }
- b) h2 { text-shadow: 4px 5px 3px #999; }
- c) h2 { text-shadow: 3px 4px 5px #999; }
- d) h2 { text-shadow: 5px 3px 4px #999; }

35. A margin és padding CSS-tulajdonságok ...

- a) csak pozitív értéket vehetnek fel;
- b) lehetnek pozitív vagy negatív értékek is;
- c) a padding csak pozitív, a margin lehet negatív is;
- d) a margin csak pozitív, a padding lehet negatív is.

36. A border-style tulajdonság solid értéke a határvonalat ...

- a) folytonos vonalként;
- b) pontok sorozataként;
- c) szaggatott vonalként;
- d) két párhuzamos vonalként jeleníti meg.

37. A border-style tulajdonság double értéke a határvonalat ...

- a) folytonos vonalként;
- b) pontok sorozataként;
- c) szaggatott vonalként;
- d) két párhuzamos vonalként jeleníti meg.

38. Az overflow tulajdonság hidden értéke a blokk görgetését úgy állítja be, hogy ...

- a) ha a tartalom túlnyúlik a blokken, akkor megjelennek a görgetősávok, egyébként nem látszanak;
- b) csak a látható tartalomrész jelenik meg; az a tartalom, amely túlnyúlik a blokken, nem jelenik meg, és görgetősávok sem jönnek létre;
- c) a blokken belül akkor is láthatók görgetősávok, ha a tartalom elfér, és nincs rájuk szükség;
- d) a tartalom akkor is megjelenik, ha túlnyúlik a blokken, görgetősávok nélkül.

39. Az overflow tulajdonság visible értéke a blokk görgetését úgy állítja be, hogy

- ...
a) ha a tartalom túlnyúlik a blokken, akkor megjelennek a görgetősávok, egyébként nem látszanak;
- b) csak a látható tartalomrész jelenik meg; az a tartalom, amely túlnyúlik a blokken, nem jelenik meg, és görgetősávok sem jönnek létre;
- c) a blokken belül akkor is láthatók görgetősávok, ha a tartalom elfér, és nincs rájuk szükség;
- d) a tartalom akkor is megjelenik, ha túlnyúlik a blokken, görgetősávok nélkül.

40. Milyen szint fog kapni a következő HTML-kódrészben a szöveg?

```
<div id="myId"><span class="myClass">Szöveg</span></div>
```

A stílusok:

```
#myId {color: green;}          /* zöld szöveg */
```

```
div .myClass {color: red;} /* piros szöveg */  
div span {color: gray;} /* szürke szöveg */
```

- a) zöld
- b) piros
- c) szürke
- d) fekete

JavaScript-tesztek

1. Helyes JavaScript-változónevek:

- a) \$ds, s_h, s1, \$gs_
- b) asd, vasia, with, fgh
- c) r#h, df, as, er
- d) \$ds, s_%h, s1, gs

2. A JavaScript adattípusai:

- a) float, double, int, string, object, char, boolean, null, undefined
- b) float, number, string, object, char, boolean, null, undefined
- c) number, string, object, boolean, null, undefined
- d) float, double, int, string, char, boolean, null

3. A következő kód futtatása után mi jelenik meg a konzolon?

```
var isAlive;  
console.log(isAlive);
```

- a) nem jelenik meg semmi
- b) null
- c) undefined
- d) 0

4. A következő kód végrehajtása után a z változó tartalma:

```
x = "45";  
var z = x + 5;
```

- a) 50
- b) 455
- c) undefined
- d) null

5. Mi íródik a konzolra a következő kód végrehajtása után?

```
var income = 45.8;  
console.log(typeof income);
```

- a) float
- b) double
- c) number
- d) 45.8

6. Mi íródik a konzolra a következő kód végrehajtása után?

```
var income = 100;
var strIncome = "100";
var result = income == strIncome;
console.log(result);
```

- a) 100
- b) "100"
- c) true
- d) false

7. Mi íródik a konzolra a következő kód futtatása után?

```
var car = {};  
if (car) console.log(false);  
else console.log(true);
```

- a) true
- b) false

8. Mi lesz az eredménye a következő kódnak?

```
let z = 20;  
{  
  let z = 30;  
  console.log(z);  
}  
console.log(z);
```

- a) hiba, mert a z változót kétszer deklaráljuk
- b) a konzolba 30 és 20 íródik

9. Mi lesz az eredménye a következő kódnak?

```
var z = 20;  
{  
  var z = 30;  
  console.log(z);  
}  
console.log(z);
```

- a) a konzolba 30 és 30 íródik
- b) a konzolba 30 és 20 íródik

10. Mi íródik a konzolra a következő kód végrehajtása után?

```
function bar() {  
  foo = "25";  
}  
bar();  
console.log(foo);
```

- a) 25
- b) undefined
- c) hiba lép fel

11. Mi íródik a konzolra a következő kód futtatása után?

```
function outer() {  
  let x = 5;  
  function inner() {  
    x++;  
    console.log(x);  
  };  
  return inner;  
}  
let fn = outer();  
fn();
```

- a) 6
- b) undefined
- c) hiba történik, ezért semmi sem íródik ki

12. Mi íródik a konzolra a következő kód futtatása után?

```
function display() {  
  console.log("0");  
  display = function () {  
    console.log("1");  
  }  
}  
var displayMessage = display;  
display();  
display();  
displayMessage();  
displayMessage();
```

- a) 1 1 1 1
- b) 0 1 1 1
- c) 0 1 0 0
- d) 1 1 0 0

13. Mi íródik a konzolra a következő kód futtatása után?

```
console.log(foo);  
var foo = "Tom";
```

- a) undefined
- b) Tom
- c) ReferenceError, mert a foo nincs definiálva

14. Mi íródik a konzolra a következő kód futtatása után?

```
function change(x) {  
  x = 2 * x;  
  console.log(x);  
}  
var n = 10;  
console.log(n);  
change(n);
```

```
console.log(n);
```

- a) 10 10 10
- b) 10 20 10
- c) 10 20 20

15. Mi íródik a konzolra a következő kód futtatása után?

```
function change(user) {  
    user.name = "Tom";  
}  
var bob = { name: "Bob" };  
console.log(bob.name);  
change(bob);  
console.log(bob.name);
```

- a) Bob Tom
- b) Bob Bob

16. Mi íródik a konzolra a következő kód futtatása után?

```
function change(array) {  
    array[0] = 8;  
}  
var numbers = [1, 2, 3];  
change(numbers);  
console.log(numbers);
```

- a) [8, 2, 3]
- b) [1, 2, 3]

17. A következő objektum esetén

```
var country = { name: "Németország", capital: { name: "Berlin",  
year: 1237 } };
```

milyen kóddal szerezhető meg a főváros alapításának éve?

- a) country["capital"]["year"]
- b) country.capital.year
- c) country.capital["year"]
- d) az „a” és „b” válasz is helyes
- e) minden felsorolt válasz helyes

18. A JavaScript nyelven ...

- a) a konstruktor egy közönséges függvény;
- b) a konstruktor olyan függvény, amelyben a `this` segítségével beállíthatók tulajdonságok és metódusok;
- c) nincsenek konstruktorok.

19. A `prototype` tulajdonság lehetővé teszi ...

- a) hogy az azonos típusú objektumok számára közös tulajdonságokat határozzunk meg;
- b) statikus objektum létrehozását;
- c) egy osztálypéldány létrehozását.

20. Alapértelmezés szerint az objektum összes tulajdonsága ...

- a) nyilvános (mindenhol elérhető);
- b) rejtett (privát).

21. Mi íródik a konzolra a következő kód futtatása után?

```
function User(name) {  
    this.name = name;  
    var _age = 1;  
}  
var tom = new User("Tom");  
console.log(tom._age);
```

- a) undefined
- b) 1
- c) hiba lép fel
- d) null

22. Támogatja-e a JavaScript az öröklés elvét?

- a) igen
- b) nem

23. Lehet-e JavaScriptben osztályokat létrehozni?

- a) igen
- b) nem

24. A super operátor lehetővé teszi ...

- a) a szülő osztály konstruktorának funkcióinak meghívását;
- b) egy „szuperképességekkel” bíró osztály jelölését;
- c) hiba létrehozását, mert ilyen operátor nincs.

25. Egy osztály konstruktorának szintaxisa lehet például:

- a) `class Person { constructor(name, age) { this.name = name; this.age = age; } }`
- b) `class Person { Person(name, age) { this.name = name; this.age = age; } }`
- c) nincsenek konstruktorok a klasszokban
- d) nincsenek sem osztályok, sem konstruktorok

26. Reguláris kifejezésekben a /D jel ...

- a) alfanumerikus karaktert jelöl;
- b) nem alfanumerikus karaktert jelöl;
- c) tizedes számjegyet jelöl;
- d) olyan karaktert jelöl, amely nem tizedes számjegy.

27. Reguláris kifejezésekben a /w jel ...

- a) alfanumerikus karaktert jelöl;
- b) nem alfanumerikus karaktert jelöl;
- c) tizedes számjegyet jelöl;
- d) olyan karaktert jelöl, amely nem tizedes számjegy.

28. Mi lesz a `nodes` tartalma a következő kód végrehajtása után?

```
var articleDiv = document.querySelector("div.article");
```

```
var nodes = articleDiv.childNodes;
```

- a) az articleDiv elem gyermekeinek szövege, attribútumai és maguk a tagek;
- b) az articleDiv elem gyermekelemei;
- c) az articleDiv elem gyermekelemeinek attribútumai és tagei.

29. A következő kód mit ír ki?

```
function getChildren(elem) {  
    for (var i in elem.childNodes) {  
        if (elem.childNodes[i].nodeType === 1) {  
            console.log(elem.childNodes[i].tagName);  
            getChildren(elem.childNodes[i]);  
        }  
    }  
}  
  
var root = document.documentElement;  
console.log(root.tagName);  
getChildren(root);
```

- a) az összes HTML-dokumentumelemet;
- b) az összes attribútumot;
- c) minden elem tartalmát;
- d) mindent, ami felsorolásra került.

30. A mousedown esemény akkor jön létre, amikor ...

- a) az egérkurzor az elem felett van, és az egérgomb le van nyomva;
- b) az egérkurzor az elem felett van, a gomb felengedésekor;
- c) az egérkurzor belép az elem határain belülre;
- d) az egérkurzor az elem felett mozog.

31. A mouseup esemény akkor jön létre, amikor ...

- a) az egérkurzor az elem felett van, és az egérgomb le van nyomva;
- b) az egérkurzor az elem felett van, a gomb felengedésekor;
- c) az egérkurzor belép az elem határain belülre;
- d) az egérkurzor az elem felett mozog.

32. A mouseover esemény akkor jön létre, amikor ...

- a) az egérkurzor az elem felett van, és az egérgomb le van nyomva;
- b) az egérkurzor az elem felett van, a gomb felengedésekor;
- c) az egérkurzor belép az elem határain belülre;
- d) az egérkurzor az elem felett mozog.

33. A mousemove esemény akkor jön létre, amikor ...

- a) az egérkurzor az elem felett van, és az egérgomb le van nyomva;
- b) az egérkurzor az elem felett van, a gomb felengedésekor;
- c) az egérkurzor belép az elem határain belülre;
- d) az egérkurzor az elem felett mozog.

34. A JSON-objektum helyes felírása:

- a) { name: "Tom", married: true, age: 30 }
- b) { "name": "Tom", "married": true, "age": 30 }

- c) { name: "Tom", married: true, "my age": 30 }
- d) mindhárom felírás helyes

35. Alapértelmezés szerint a süti (cookie) élettartama ...

- a) 24 óráig tart;
- b) a böngésző bezárásáig tart;
- c) addig tart, amíg kézzel nem törlik a sütiket;
- d) nincs alapértelmezett élettartam, kötelező megadni.

36. A böngésző memóriatípusai közül melyik továbbítódik HTTP-n keresztül?

- a) session storage
- b) local storage
- c) cookie
- d) mindhárom

37. A sessionStorage ...

- a) ideiglenes adatmegőrzést valósít meg, amely a böngésző bezárásakor törlődik;
- b) tartós adatmegőrzésre szolgáló tároló;
- c) nem jelent semmit, mert ilyen típusú tároló nincs.

38. A localStorage ...

- a) ideiglenes adatmegőrzést valósít meg, amely a böngésző bezárásakor törlődik;
- b) tartós adatmegőrzésre szolgáló tároló;
- c) nem jelent semmit, mert ilyen típusú tároló nincs.

PHP-tesztek

1. A PHP-vel leggyakrabban együtt használt webszerver és adatbázis-kezelő ...

- a) IIS, MsSQL Server
- b) Nginx, PostgreSQL
- c) Apache, MySQL
- d) Nginx, MsSQL Server

2. A HTML-ürlapról elküldött adatok PHP-ben az alábbi globális tömbökből érhetők el ...

- a) \$_GET, \$_POST
- b) \$_GET, \$_ECHO
- c) \$_POST, \$_ECHO
- d) \$_ECHO, \$_FORM

3. A PHP-ben a karakterláncok összefűzésének operátora ...

- a) ++
- b) +
- c) . (pont)
- d) concat()

4. A PHP-változók például ...

- a) \$_for, \$&gh, \$gh
- b) \$_4for, \$h_dsg3, \$while
- c) \$4for, \$h_dsg3, \$ghf
- d) _gh4, \$for, \$f3g4

5. Mit ír a következő PHP-kód a weboldalra?

```
$i = 4;  
echo "két szám négyzete $i egyenlő " . $i * $i;
```

- a) „két szám négyzete 4 egyenlő 16”
- b) „két szám négyzete 4 egyenlő 44”
- c) „két szám négyzete \$i egyenlő 16”
- d) „két szám négyzete \$i egyenlő 44”

6. A PHP nyelv az alábbi adattípusokat támogatja:

- a) boolean, integer, float, string, array, object, resource, NULL
- b) boolean, integer, double, float, string, array, object, NULL
- c) boolean, int, double, string, array, object, resource, NULL
- d) boolean, integer, double, string, array, object, resource, NULL

7. Egy változó a hivatkozás (referencia) szerinti átadásához így kell definiálni a függvényt:

- a) `function get(ref $a){ /*...*/ }`
- b) `function get(&$a){ /*...*/ }`
- c) `function get(a){ /*...*/ }`
- d) `function get(out $a){ /*...*/ }`

8. Egy függvényen belül definiált változó globális eléréséhez a következő kulcsszót kell használnunk:

- a) static
- b) public
- c) global
- d) var

9. Egy PHP-fájl másik PHP-fájlba történő beillesztésére használható ...

- a) include
- b) require
- c) include_once
- d) mindhárom

10. Egy PHP-fájl másik PHP-fájlba való beillesztésére, úgy hogy hiány esetén hibaüzenetet kapjunk és csak egyszer történjen meg a beillesztés, használható ...

- a) include
- b) require_once
- c) include_once
- d) mindhárom

11. A \$a tömb elem számának lekérdezésére PHP-ben ...

- a) `count($a), sizeof($a)`
- b) `length($a), sizeof($a)`
- c) `count($a), length($a)`
- d) `count($a), length($a), sizeof($a)`

12. Egy karakterlánc-tömb betűrend szerinti, illetve egy számtömb növekvő sorrend szerinti rendezésére PHP-ben ...

- a) `asort()`

- b) `arsort()`
- c) `ksort()`
- d) `krsort()`

13. Egy számtömb elemeinek csökkenő sorrendbe rendezésére PHP-ben ...

- a) `asort()`
- b) `arsort()`
- c) `ksort()`
- d) `krsort()`

14. Egy tömb elemeinek kulcs szerinti rendezésére PHP-ben ...

- a) `asort()`
- b) `arsort()`
- c) `ksort()`
- d) `kksort()`

15. A cookie-k olvasására PHP-ben a következő globális asszociatív tömb használható:

- a) `_COOKIE`
- b) `COOKIE`
- c) `$_COOKIE_START`
- d) `$_COOKIE`

16. Melyik meghívás definiál helyesen cookie-t?

- a) `setcookie("language", "kínai", time() + 3600)`
- b) `setcookie("language", "kínai")`
- c) `setcookie("language", "kínai", time() + 3600, "/")`
- d) mindhárom helyes

17. A kifejezés `setcookie("language", "kínai", time() + 3600)` egy cookie-t ...

- a) egy órára hoz létre;
- b) egy percre hoz létre;
- c) 36 másodpercre hoz létre;
- d) 36 percre hoz létre.

18. A munkamenet (session) indításához meg kell hívni a ... függvényt.

- a) `session_enter()`
- b) `session_start()`
- c) `session_push()`
- d) `session_begin()`

19. A munkamenet-változó írásához vagy olvasásához a következő globális asszociatív tömb használható:

- a) `_SESSION`
- b) `$_SESSION`
- c) `$SESSION`
- d) `$_SESSION_START`

20. A GET kérésekben a paraméter-érték párok elválasztó jele ...

- a) `&`
- b) `$`

- c) +
- d) ;

21. Mi jelenik meg a weboldalon a következő kód futtatásakor?

```
echo 11 . 22;
```

- a) 33
- b) 1122
- c) 11.22
- d) 11

22. Egy változó \$a megsemmisítésére használható operátor ...

- a) unset(\$a)
- b) isset(\$a)
- c) delete(\$a)
- d) drop(\$a)

23. Egy változó \$a inicializáltságának ellenőrzésére használható operátor ...

- a) unset(\$a)
- b) isset(\$a)
- c) isvariable(\$a)
- d) isvar(\$a)

24. Mi jelenik meg a weboldalon a következő kód futtatásakor?

```
$a = NULL;  
echo $a;
```

- a) 0 jelenik meg
- b) nem jelenik meg semmi
- c) NULL jelenik meg
- d) hiba lép fel

25. Mi jelenik meg a weboldalon a következő kód futtatásakor?

```
$a = 4;  
echo -$a--;
```

- a) 4
- b) 3
- c) -3
- d) -4

26. Mit ír a következő kód a weboldalra?

```
$a = 10;  
$b = 5;  
echo "$a + $b " . '$a + $b';
```

- a) 10 + 5 10 + 5
- b) \$a + \$b \$a + \$b

- c) 15 \$a + \$b
- d) 10 + 5 \$a + \$b

27. Egy NUMBER nevű konstans definiálásához írjuk ...

- a) `var NUMBER=22`
- b) `final NUMBER=22`
- c) `const NUMBER=22`
- d) `define("NUMBER", 22)`

28. Mit ír a következő kód a weboldalra?

```
$a = 10=="10";  
$b = 10!="10";  
echo $a . $b;
```

- a) üres karaktereket
- b) 11
- c) 10
- d) 01

29. Mit ír a következő kód a weboldalra?

```
echo 7 > 6 && 8 > 8 or 9 > 8;  
echo 1 || 0 xor 0;
```

- a) üres karaktereket
- b) 11
- c) hiba lép fel
- d) 1

30. Mit ír a következő kód a weboldalra?

```
$n = 10;  
for($i = 0; $i < $n; $i++, $n--);  
echo $i;
```

- a) 5
- b) 0
- c) 0123456789
- d) 01234

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ / AJÁNLOTT SZAKIRODALOM

1. **Двірничук К. В., Вацек D. О.** Веб-програмування та веб-дизайн: навч. посіб. Чернівці: Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с. (українською)
2. **European Commission.** Proposal for a Directive on the approximation of the laws, regulations and administrative provisions of the Member States as regards the accessibility requirements for products and services. 2022. (англійською)
3. **Flanagan D.** JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, 2020. (англійською)
4. **Фрімен Е., Робсон Е.** Head First. Програмування на JavaScript. Фабула, 2022. 672 с. (українською)
5. **GetBootstrap.com.** Офіційна документація Bootstrap 5. Оновлюється постійно. (англійською)
6. **Гайна Г. А.** Основи проектування баз даних: навчальний посібник. Навчально-методичне видання, 2018. (українською)
7. **Infojegyzet.hu.** Ajax pofonegyszerűen. Online tananyagok. Frissítve: 2021. (угорською)
8. **Infojegyzet.hu.** HTML CSS feladatok. Online tananyagok i feladatok. Frissítve: 2024. (угорською)
9. **Infojegyzet.hu.** HTML CSS jegyzet: HTML űrlap elemek. Online tananyagok. Frissítve: 2024. (угорською)
10. **Infojegyzet.hu.** JS feladatok: eseménykezelés. Online tananyagok i feladatok. Frissítve: 2024. (угорською)
11. **Infojegyzet.hu.** JS jegyzet: objektumok hierarchiája. Online tananyagok. Frissítve: 2024. (угорською)
12. **Infojegyzet.hu.** MySQL adatbázisok és lekérdezési feladatok. Online tananyagok. Frissítve: 2024. (угорською)
13. **Infojegyzet.hu.** MySQL lekérdezések: Reptér adatbázis. Online feladatok. Frissítve: 2024. (угорською)
14. **Infojegyzet.hu.** PHP jegyzet: REST API és JSON. Online tananyagok. Frissítve: 2024. (угорською)
15. **Infojegyzet.hu.** PHP MySQL jegyzet: Regisztráció megvalósítása. Online tananyagok. Frissítve: 2020. (угорською)
16. **Infojegyzet.hu.** PHP-MySQL mintapélda: Európa országai. Online tananyagok. Frissítve: 2020. (угорською)
17. **Infojegyzet.hu.** PHP jegyzet: Szerveroldali programozás. Online tananyagok. Frissítve: 2024. (угорською)
18. **Infojegyzet.hu.** Webszerkesztés: Bootstrap. Online tananyagok i feladatok. Frissítve: 2024. (угорською)
19. **Infojegyzet.hu.** Webszerkesztés: CSS alapok. Online tananyagok i feladatok. Frissítve: 2024. (угорською)
20. **Infojegyzet.hu.** Webszerkesztés: JavaScript alapok. Online tananyagok i feladatok. Frissítve: 2024. (угорською)
21. **jQuery.com.** Офіційна документація jQuery. Оновлюється постійно. (англійською)
22. **Курінний С.** Розробка веб-сайтів для початківців: HTML, CSS, JavaScript. 2022. (українською)
23. **MDN Web Docs.** Flexbox. Mozilla Foundation. Оновлюється постійно. (англійською)
24. **MDN Web Docs.** HTML forms. Mozilla Foundation. Оновлюється постійно. (англійською)

25. **MDN Web Docs.** The box model. Mozilla Foundation. Оновлюється постійно. (англійською)
26. **MDN Web Docs.** WAI ARIA Basics. Mozilla Foundation. Оновлюється постійно. (англійською, є частковий український переклад)
27. **Meloni J. C., Kyrnin J.** HTML, CSS, and JavaScript All in One: Covering HTML5, CSS3, and ES6, Sams Teach Yourself. Sams Publishing, 2021. (англійською)
28. **Nagy G.** Web programozás alapismeretek. Online tankönyv. 2011. (угорською)
29. **Nixon R.** Learning PHP, MySQL & JavaScript, 6th Edition. O'Reilly Media, 2021. (англійською)
30. **Nixon R.** Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, 6th Edition. O'Reilly Media, 2021. (англійською)
31. **Онлайн ресурс.** Організація Баз Даних і Знань: Лабораторні роботи (MySQL, MongoDB). Навчальний посібник, 2022. (українською)
32. **PHP.net/manual/uk.** Офіційна документація PHP по SimpleXML та JSON. Оновлюється постійно. (українською)
33. **ПРООН в Україні.** Короткий посібник з цифрової доступності. 2023. (українською)
34. **ПРООН та Міністерство цифрової трансформації України.** Моніторинг базової вебдоступності офіційних вебсайтів органів державної влади. 2023. (українською)
35. **React.dev.** Офіційна документація React. Оновлюється постійно. (англійською)
36. **Stefanov S.** React: Up & Running: Building Web Applications, 2nd Edition. O'Reilly Media, 2021. (англійською)
37. **Svekis L.** Front End Web Development with Modern HTML, CSS, and JavaScript. Packt Publishing, 2022. (англійською)
38. **Svekis L.** HTML, CSS, and JavaScript for Beginners: A Web Design Course. Packt Publishing, 2021. (англійською)
39. **The A11Y Project.** ARIA. Оновлюється постійно. (англійською)
40. **The A11Y Project.** Checklist. Оновлюється постійно. (англійською)
41. **The A11Y Project.** How tos. Оновлюється постійно. (англійською)
42. **W3C (World Wide Web Consortium).** Extensible Markup Language (XML) Specification. Офіційна документація. (англійською)
43. **W3C Web Accessibility Initiative (WAI).** Easy Checks – A First Review of Web Accessibility. Оновлюється постійно. (англійською)
44. **W3C Web Accessibility Initiative (WAI).** Introduction to Web Accessibility. Оновлюється постійно. (англійською)
45. **W3C.** Настанови з доступності вебвмісту (WCAG) 2.1. Авторизований переклад W3C. 2021. (українською)
46. **W3C.** Web Content Accessibility Guidelines (WCAG) 2.0. 2008. (угорський переклад доступний на сайті W3C)
47. **W3C.** Web Content Accessibility Guidelines (WCAG) 2.2. 2023. (англійською)
48. **W3C.** WAI ARIA Authoring Practices 1.2. Оновлюється постійно. (англійською)
49. **WCAG2.com.** eAccessibility Library. David Berman Communications. Оновлюється постійно. (англійською)
50. **WCAG2.com.** eAccessibility Library: Testing tools. David Berman Communications. Оновлюється постійно. (англійською)
51. **Welling L., Thomson L.** PHP and MySQL Web Development, 5th Edition. Addison Wesley Professional, 2016. (англійською)

НАВЧАЛЬНО-МЕТОДИЧНЕ ВИДАННЯ

«Вебпрограмування»
методичні рекомендації для самостійної роботи
2025 р.

Затверджено до використання у навчальному процесі
на засіданні кафедри математики та інформатики Університету Ракоці
(протокол № 3 від «15» жовтня 2025 року)

Укладачі:

Адам ДОРОВЦІ – PhD, доцент кафедри математики та інформатики Закарпатського угорського університету імені Ференца Ракоці II.

За зміст методичних вказівок відповідальність несуть розробники.

Видавництво: Закарпатський угорський університету імені Ференца Ракоці II (адреса: пл. Кошута 6, м. Берегове, 90202. Електронна пошта: foiskola@kmf.uz.ua) *Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції Серія ДК ____ від ____ 2025 року*

Шрифт «Times new roman» та «Courier New». Розмір сторінок методичних вказівок: А4 (210x297мм)