

1. A CSS képpont (px) mértékegysége valóban, mindig fizikai képpontokban méri a távolságot? Ha nem, mondjon példát rá, hogy milyen esetben és miért nem!

- a. Például, egy Retina kijelzőn 1 CSS px gyakran 2 vagy 3 fizikai képpontnak felel meg, míg egy hagyományos kijelzőn 1 CSS px valóságos 1 fizikai képpontot jelenthet.
- b. Miért fontos ez?
 - i. *Ez lehetővé teszi, hogy a weboldalak reszponzív módon jelenjenek meg különböző eszközökön. Ha a px-t fizikai képpontokban mérnénk, akkor az egyes eszközök eltérő megjelenítése miatt a weboldal kinézete nagyon különböző lenne. A logikai képpontok használata biztosítja, hogy az oldalak a különböző kijelzőkön megfelelően jelenjenek meg, függetlenül azok fizikai felbontásától.*

2. Adjon meg három módszert, amivel JavaScript programban eseménykezelő függvényt lehet társítani egy eseményhez! Mik ezek előnyei-hátrányai?

- a. Inline eseménykezelés (HTML-ben)
 - i. `<button onclick="alert('Hello!')">Click me</button>`
 - ii. Előnye, gyors, és egyszerű, hátránya hogy nem rugalmas.
- b. DOM eseménykezelő (JavaScript-ben)
 - i. `document.querySelector('button').onclick = function() {`
 - ii. `alert('Hello!');`
 - iii. `};`
 - iv. Előnye, hogy JavaScriptben könnyen elérhető, hátránya, hogy egy eseményhez csak egy függvény írható.
- c. `addEventListener()` metódus
 - i. `document.querySelector('button').addEventListener('click', function() {`
 - ii. `alert('Hello!');`
 - iii. `});`
 - iv. A Javascript, és a html teljesen el van választva, több függvény is hozzáadható egy eseményhez, hátránya, hogy hamar bonyolulttá tud válni.

3. Mi az a három módszer, amivel valaminek a színét be lehet állítani CSS-ben? Adjon ezekre egy-egy mintát!

- a. Szín név (Color Name)
 - i. `color: red;`
- b. Hexadecimális érték (Hex Color)
 - i. `color: #3498db;`
- c. RGB (Red, Green, Blue)
 - i. `background-color: rgb(34, 193, 195);`

4. Miért lehet indokolt több adatforrást is megadni weboldalba ágyazott videó lejátszásakor?

- a. Különböző eszközök és böngészők eltérő videóformátumokat támogatnak, és felületenként változhat melyik formátum a legmegfelelőbb.

5. Adjon meg legalább hármat a HTML5 újdonságai közül!

- a. Az `<article>` elem egy önálló, elkülönített tartalomblokknak felel meg, amely függetlenül is megérthető és újrahasznosítható. Általában blogbejegyzéseknél, híreknél vagy egyéb hasonló tartalmaknál használják.
- b. A HTML5 bevezette a `<video>` elemet, amely lehetővé teszi videók közvetlen beágyazását a weboldalakra, natív lejátszással, anélkül, hogy külső plugineket (például Flash-t) kellene használni.
- c. A `<header>` elem egy dokumentum vagy szekció fejlécét jelöli, amely tartalmazhat például navigációs menüt, címeket, vagy logót. Könnyen megjegyezhető, mert a „header” angolul is „fejléct” jelent.
- d. `<section>`, `<article>`, `<details>`, `<summary>`, `<mark>`, `<progress>`, `<meter>`, `<output>`, `<figure>`, `<figcaption>`, `<time>`, `<address>`, `<dialog>`, `<template>`, `<wbr>`

6. Adja meg azt a reguláris kifejezést, ami illeszkedik a 9-nél nagyobb egész számokra!

- a. `^[1-9]\d*$`
 - i. `^`: A kifejezés elejét jelöli.
 - ii. `[1-9]`: Az első számjegy 1-től 9-ig terjed, biztosítva, hogy a szám nem lehet 0.
 - iii. `\d*`: Bármilyen számú további számjegyet engedélyez (vagyis a szám bármilyen hosszú lehet).
 - iv. `$`: A kifejezés végét jelöli.

7. Milyen szintaktikai szerepe van a `::`-nak?

- a. A `::` szintaktikai szerepe a CSS-ben **pseudo-elemek** (pszeudo-elemek) létrehozására szolgál. A `::` egy speciális jelölés, amely lehetővé teszi a stílusok alkalmazását olyan elemekre, amelyek nem tartoznak az alap HTML elemekhez, vagy a dokumentum normál struktúráján kívül esnek.
 - i. `::before` – Az elem tartalma elé beszúrt tartalom stílusozására használható.
 - ii. `::first-letter` – Az elem első karakterének stílusozására.
 - iii. `::first-line` – Az elem első sorának stílusozására.

8. A margó (margin), szegély (border) és kitöltés (padding) közül melyiknek/melyeknek lehet negatív méretet megadni?

- a. A margó értéke lehet negatív. A negatív margó azt jelenti, hogy az elem a szomszédos elemekhez közelebb kerül, vagy akár átfedheti őket.

9. Milyen űrlapvezérlőnél nélkülözhetetlen a value attribútum megadása, és miért?

- a. Az űrlapvezérlők közül a radio (rádiógomb) és a checkbox (jelölőnégyzet) típusú inputoknál kötelező megadni a value attribútumot

10. Hogyan lehet beállítani CSS-ben a kitöltések (padding) színét?

- a. Mivel a padding a tartalom (content) és a szegély (border) közötti távolságot határozza meg a padding terület színét az elem háttérszínével (background-color) lehet beállítani.

11. Írjon egy példát arra, amikor szükséges egy űrlapvezérlőnél (pl. küldés gomb) megadni a formaction attribútumot!

- a. A formaction attribútumot olyan esetekben használjuk, amikor egy adott űrlapvezérlőhöz (például egy küldés gombhoz) eltérő cselekvési útvonalat (action URL-t) szeretnénk megadni az űrlap beküldésekor. Ez lehetővé teszi, hogy ugyanazon az űrlapon belül több különböző célú gombot definiáljunk, amelyek eltérő műveleteket hajtanak végre.

12. Hol, milyen szoftverek szokták a webes tartalmakat gyorsítótárazni? Írjon legalább két példát!

- a. Böngészők: A böngészők, mint például a Google Chrome vagy Firefox, gyorsítótárazzák a weboldalak statikus elemeit (CSS, JavaScript, képek), hogy a visszatérő látogatók számára gyorsabb legyen a betöltés.
- b. Proxy szerverek és CDN-ek: Szolgáltatások, mint az NGINX, Varnish vagy Cloudflare, gyorsítótárazzák a tartalmakat a szerverek és a felhasználók között, csökkentve az eredeti szerver terhelését és javítva a válaszidőt.

13. Adjon meg legalább három relatív CSS mértékegységet!

- a. vw (viewport width): A nézetablak (viewport) szélességének 1%-a.
- b. vh (viewport height): A nézetablak magasságának 1%-a.
- c. em: Az aktuális elem betűméretének szorzata (pl. 2em kétszeres betűméret).
- d. rem: A gyökér (root) elem betűméretének szorzata (pl. 1.5rem másfélszeres gyökér betűméret).
- e. %: Az aktuális elem szülőeleméhez viszonyított százalékos arány (pl. szélesség 50%).

14. Mi a SonarQube, mire használható?

- a. A SonarQube egy nyílt forráskódú statikus kódelemző eszköz, amely a szoftverfejlesztés során a kód minőségének javítását szolgálja. A SonarQube képes az alkalmazott kódban található hibákat, biztonsági réseket, kódminőségi problémákat és potenciális hibákat azonosítani, valamint mérni a kód karbantartási könnyűségét és olvashatóságát.

15. Mi az a reszponzivitás, reszponzív webdesign?

- a. A reszponzivitás és a reszponzív webdesign a webfejlesztésben olyan tervezési megközelítéseket jelentenek, amelyek célja, hogy a weboldalak jól jelenjenek meg különböző eszközökön és képernyőméreteken, például asztali számítógépeken, tableteken és okostelefonokon.

16. Sorolja fel a Memex legalább három tulajdonságát, jellemzőjét!

- a. Személyre szabott tárolás: A Memex olyan módon tárolta az információkat, hogy az egyén személyes igényei szerint rendszerezhesse azokat, hasonlóan ahhoz, ahogy ma a számítógépes fájlrendszerek működnek.
- b. Keresés és indexelés: A Memex rendszerében az információk gyorsan kereshetők és indexelhetők voltak, ami azt jelentette, hogy a felhasználók könnyen megtalálhatták a kívánt adatokat.
- c. Linkelés és asszociációk: lehetővé tette volna az információk közvetlen összekapcsolását, azaz linkek létrehozását a különböző adatpontok között.
- d. Széleskörű tárolás: A Memex eszköz képes lett volna különféle típusú adatokat tárolni, beleértve a szöveget, képeket és egyéb dokumentumokat.

17. Mi az az npm?

- a. Az npm (Node Package Manager) egy csomagkezelő rendszer, amely a Node.js környezethez készült, és a JavaScript fejlesztésében széles körben használt. Az npm lehetővé teszi a fejlesztők számára, hogy könnyedén telepítsenek, frissítsenek, kezeljenek és osszanak meg csomagokat (modulokat), amelyek különféle JavaScript funkciókat és könyvtárakat tartalmaznak.

18. Milyen design pattern-t követ a Microsoft ASP.NET?

- a. Az egyik legfontosabb és legszélesebb körben használt minta az MVC (Model-View-Controller) minta. Ezen kívül az ASP.NET alkalmazások más design pattern-eket is követhetnek, például a Singleton, Factory és Repository mintákat.

19. Nevezzen meg legalább két okot, amiért érdemes lehet jQuery-t is használni egy weboldalon!

- a. Egyszerű animációk hozzáadása: A jQuery lehetővé teszi egyszerű animációk (pl. elhalványulás, mozgás, méretezés) hozzáadását a DOM elemekhez egyetlen sor kóddal, anélkül, hogy bonyolult CSS vagy JavaScript kódot kellene írni.
- b. Kiváló eseménykezelés: A jQuery könnyen kezelhető eseményekkel (pl. kattintás, hover, billentyűzetnyomás) segíti a felhasználói interakciók kezelését, és támogatja az események delegálását, így egyszerűsíti a dinamikus alkalmazások fejlesztését.
- c. DOM manipulációk egyszerűsége: jQuery rövid és tiszta szintaxist biztosít HTML elemek kiválasztásához és módosításához, például egy elem tartalmának vagy stílusának megváltoztatásához.
- d. Komplex formák kezelése: jQuery egyszerűsíti a formák validálását és a bejegyzett adatokat, lehetővé téve az egyszerű űrlap-ellenőrzést és a hibák vizuális jelzését, mielőtt az adatokat elküldenénk a szervernek.
- e. Keresztböngészős kompatibilitás: jQuery automatikusan gondoskodik a különböző böngészők közötti inkompatibilitások kezeléséről.

20. A TCP/IP protokollstruktúrájának melyik rétegében található a HTTP protokoll?

- a. A HTTP (HyperText Transfer Protocol) protokoll a TCP/IP protokollstruktúra alkalmazási rétegében található.

21. Mik azok a süтик, és mire használhatóak?

- a. A süтик (cookies) kis méretű adatcsomagok, amelyeket a weboldalak tárolnak a felhasználó böngészőjében, és amelyeket a weboldalak később hozzáférhetnek. Ezek az adatok lehetnek egyszerű információk, mint például felhasználói preferenciák, bejelentkezési állapotok vagy nyomon követési adatok.

22. Miért előnyösebb a W3C eseménykezelési módszerét alkalmazni a „klasszikus” eseménykezelés helyett?

- a. Mert biztosítja a böngészőkompatibilitást, lehetővé teszi több eseménykezelő hozzáadását ugyanahhoz az elemhez, és támogatja az eseménybuborékolást és elkapást, így rugalmasabb eseménykezelést kínál.
- b. Ezen kívül tisztább, karbantarthatóbb kódot eredményez, mivel elválasztja a HTML-t a JavaScript-től, és egyszerűsíti az események eltávolítását is a `removeEventListener()` metódussal.

23. Hogyan lehet megakadályozni, hogy a böngésző végrehajtsa egy eseményhez rendelt alapértelmezett tevékenységeit? Minden módszert írjon le, amit csak ismer!

- a. `event.preventDefault()`: Megakadályozza az esemény alapértelmezett viselkedését.
- b. `return false`: Régebbi módszer, amely szintén megakadályozza az alapértelmezett viselkedést.
- c. `event.stopPropagation()`: Megakadályozza, hogy az esemény továbbterjedjen a DOM fában, ami indirekt módon hatással lehet az alapértelmezett viselkedésre.

24. Mire jó a HTTP feltételes GET metódusa?

- a. A HTTP feltételes GET metódusa lehetővé teszi, hogy a böngésző csak akkor kérje le az erőforrást a szervertől, ha az változott az utolsó lekérés óta. Ezt **If-Modified-Since** vagy **If-None-Match** fejlécekkel használja, így csökkentve a hálózati forgalmat és javítva a teljesítményt, mivel nem kell újra letölteni a változatlan tartalmakat.

25. Miért nem kedvelik a webfejlesztők az alert, confirm, prompt JavaScript függvényekkel megjelenített ablakokat? Melyik mire szolgál?

- a. A webfejlesztők gyakran nem kedvelik az `alert`, `confirm` és `prompt` függvényekkel megjelenített ablakokat, mert ezek blokkolják a felhasználói interakciót, zavarnak a felhasználói élményben, és nem testreszabhatóak.
- b. `alert()`: Egy egyszerű figyelmeztető üzenetet jelenít meg, és a felhasználónak el kell azt fogadnia, hogy tovább lépjen.
- c. `confirm()`: Egy igaz/hamis kérdést jelenít meg, és a felhasználónak választania kell (OK/Cancel).
- d. `prompt()`: Egy szöveges mezőt biztosít a felhasználónak, hogy adatot adjon meg (pl. név), majd választ adjon.

26. Hogyan lehet HTML-ben megadni az oldal készítőjének nevét?

- a. `<meta name="author" content="Készítő Neve">`
- b. TML-ben az oldal készítőjének nevét a `<meta>` tag segítségével adhatjuk meg a dokumentum fejlécében.

27. Milyen típusú felsorolásokat lehet készíteni HTML-ben?

- a. Rendezetlen felsorolás (``): Elemek listája, amelynek nincs sorrendje, és alapértelmezés szerint pontokkal jelöli a listát.
- b. Rendezett felsorolás (``): Sorszámozott lista, ahol az elemek sorrendje fontos, és a lista automatikusan sorszámozza őket.
- c. Lejegyzett lista (`<dl>`): Kérdések és válaszok típusú lista, ahol az egyes elemek kulcs-érték párban jelennek meg (`<dt>` a kulcs, `<dd>` az érték).

**28. Soroljon fel legalább három blokk szintű elemet, a rendeltetésükkel együtt!
A blokk szintű (block-level) elemek mindig új sorban kezdődnek, és vízszintesen kitöltik a rendelkezésre álló helyet.**

- a. <div> – Általános tárolóelem, amelyet a weboldalon különböző tartalmi egységek csoportosítására használunk. Jellemzően stílusok és elrendezések alkalmazására használják.
- b. <p> – Bekezdés, szövegek csoportosítására szolgál. Az egyes bekezdések közötti térközt is meghatározza.
- c. <h1>, <h2>, <h3>, stb. – Fejléc elemek, amelyek a címeket jelölik, és hierarchikus struktúrát adnak az oldalnak (h1 a legmagasabb szintű, h6 a legalacsonyabb).
- d. – Rendezetlen felsorolás, amely listát jelenít meg a elemek segítségével. Az elemek pontokkal jelennek meg.
- e. – Rendezett felsorolás, ahol a listaelemek számozottak, és az elemek sorszáma a tagek segítségével van megadva.