

<https://www.edureka.co/blog/interview-questions/css-interview-questions/>

Melyek a CSS főbb verziói?

- CSS 1
- CSS 2
- CSS 2.1
- CSS 3
- CSS 4

Hogyan lehet integrálni a CSS-t a HTML-oldalába?

Három módon lehet integrálni egy bizonyos CSS stílust:

- A stílusát integrálhatja a HTML-oldal fejrészában található stíluscímkék használatával.
- Integrálhatja stílusát az inline stílus használatával.
- A CSS-t külön fájlba írhatja, és a linkcímké segítségével hozzáadhatja a HTML-oldalához.

Mit jelent a kaszkádózás?

A CSS forradalmat hozott a webfejlesztésben, és abban, hogy az emberek hogyan érzékelik a weboldal építésének folyamatát. A CSS megléte előtt az elemeket in-line módon kellett megtervezni, vagy a stílust egy HTML-oldal fejrészában kellett megvalósítani. Ez a CSS lépcsőzetes jellege miatt megváltozott. A CSS három lépést mutat be:

- Elemek - Ugyanazt a CSS stílust lehet alkalmazni több elemre ugyanazon stílus elérése érdekében.
- Több stílusú egy elem - Egy egyedi stílus eléréséhez több stílus is alkalmazható egy adott HTML elemre.
- Ugyanaz a stílus, több oldal - Ugyanazt a stíluslapot lehet különféle HTML oldalakra alkalmazni, hogy a sablon stílusa nagyon gyors legyen.

Milyen előnyei vannak a CSS használatának?

- Több dokumentumnak lehet ugyanaz a stílusa.
- Több HTML elem sok dokumentumot tartalmazhat, ahol osztályok hozhatók létre.
- A stílusok összetett helyzetekben történő csoportosításához választók (selector) és csoportosítási módszereket alkalmaznak.

Milyen hátrányai vannak a CSS használatának?

- A szelektorok általi csökkenő sorrend
- A függőleges vezérlés korlátai

- Nincsenek kifejezések
- Nincs oszlop deklaráció
- A pseudo osztály dinamikus viselkedés által nem vezérelhető
- Szabályok, stílusok, konkrét szöveg célzása nem lehetséges

Milyen típusú médiatípusokat engedélyez a CSS?

A @media tulajdonságoknak négy típusa van (beleértve a képernyőt is):

- all - minden adathordozó típusú eszközhöz
- print - nyomtatók számára
- speech - a képernyőolvasók számára, akik hangosan „olvassák” az oldalt
- screen - számítógépes képernyőkhöz, táblagépekhez, okostelefonokhoz stb.

Íme egy példa a nyomtatott média típusának használatára:

```
@media print {
  h1 {
    background-color: yellow;
  }
}
```

Melyek a CSS-ben használt különféle egységek?

A CSS-nek kétféle hosszúsága van. Relatív hosszúság és abszolút hosszúság. Különböző egységeket használnak rájuk.

Relative Length

UNIT	DESCRIPTION
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

Absolute Length

UNIT	DESCRIPTION
CM	centimetres

MM	millimetres
IN	inches (1in = 96px = 2.54cm)
PX	pixels (1px = 1/96th of 1in)
PT	points (1pt = 1/72 of 1in)
PC	picas (1pc = 12 pt)

Melyek a CSS különböző betűtípus-attribútumai?

Az alábbiakban a CSS-ben elérhető különböző betűtípusokkal kapcsolatos attribútumok találhatók:

- Font-style
- Font-variant
- Font-weight
- Font-size/line-height
- Font-family
- Caption
- Icon

Mi a box-shadow használata a CSS-ben?

A box-shadow CSS tulajdonság árnyékhatásokat ad az elem kerete körül. Több effektust vesszővel elválasztva állíthat be. A box-shadow X és Y eltolások írják le az elemhez, color, blur and spread radii képest. Az alábbiakban bemutatjuk a box-shadow néhány megvalósítását

```
box-shadow: 10px 5px 5px red;
```

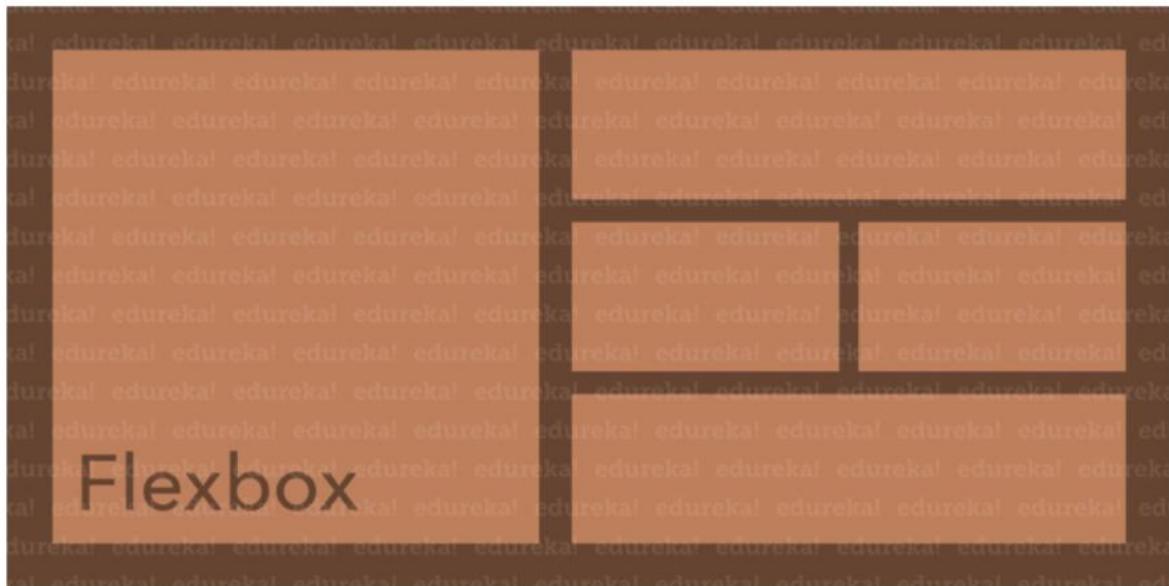
```
box-shadow: 60px -16px teal;
```

```
box-shadow: 12px 12px 2px 1px rgba(0, 0, 255, .2);
```

```
box-shadow: inset 5em 1em gold;
```

Mi az a CSS flexbox?

A flexbox elrendezés, amelyet hivatalosan CSS rugalmas elrendezési modulnak hívnak, egy új elrendezési modul a CSS3-ban. Arra szolgál, hogy javítsa az elemek összehangolását, irányát és sorrendjét a konténerben, még akkor is, ha dinamikus vagy akár ismeretlen méretűek. A flex konténer elsődleges jellemzője, hogy módosíthatja gyermekei szélességét vagy magasságát, hogy a rendelkezésre álló helyet a lehető legjobban kitöltse a különböző képernyőméreteken.



Sok tervező és fejlesztő úgy találja, hogy ez a flexbox elrendezés könnyebben használható, mivel az elemek pozicionálása egyszerűbb, így kevesebb kóddal bonyolultabb elrendezés érhető el, ami egyszerűbb fejlesztési folyamathoz vezet. A Flexbox elrendezés algoritmus irányalapú, ellentétben a függőleges- vagy vízszintes elrendezéssel, amelyek függőlegesen és vízszintesen vannak. Ezt a flexbox elrendezést kis alkalmazás-összetevőknél kell használni, míg az új CSS Grid Layout Module megjelenik a nagyméretű elrendezések kezelésére.

Hogyan állapítja meg a böngésző, hogy mely elemek felelnek meg a CSS-választóval?

A böngészők a jobb oldalról (key selector) balra nézik a selector-okat. A böngészők a kulcsválasztó szerint kiszűrik az elemeket a DOM-ban, és felfelé haladnak a szülőelemek között, hogy meghatározzák az egyezéseket. Minél rövidebb a választó lánc hossza, annál gyorsabban tudja a böngésző megállapítani, hogy az elem megfelel-e a választónak.

Például ezzel a választóval `p span`, a böngészők először megtalálják az összes `` elemet, és felfelé haladnak egészen a gyökérig, hogy megtalálják az `<p>` elemet. Egy adott `` esetében, amint talál egy `<p>`-ot, tudja, hogy a `` egyezik, és leállíthatja az egyezést.

Magyarázza el azt a forgatókönyvet, amelyet a `translate()` segítségével használna az abszolút pozicionálás helyett?

A `translate()` a CSS-transzformáció értéke. Az átalakítás vagy az átlátszatlanság megváltoztatása nem váltja ki a böngésző átfestését vagy újrafestését, de összetételeket vált ki; míg az abszolút pozicionálás megváltoztatása visszavezetést vált ki. Az átalakítás hatására a böngésző GPU-réteget hoz létre az elem számára, de az abszolút pozicionálási tulajdonságok megváltoztatása a CPU-t használja. Ezért a `translate()` hatékonyabb, és rövidebb festési időket eredményez a simább animációk számára.

A `translate()` használatakor az elem továbbra is elfoglalja eredeti terét (hasonló `position: relative` - hoz), ellentétben az abszolút helyzet megváltoztatásával.

Magyarázza meg a megközelítés különbségét, amikor egy reszponzív weboldalt tervezése és a mobil-first stratégia között?

Ez a két megközelítés nem kizárólagos. Ha egy weboldalt reszponzívvá tesz, az azt jelenti, hogy egyes elemek reagálnak a méretének vagy más funkcióinak az eszköz képernyőméretéhez, jellemzően a nézetablak szélességéhez igazításával, CSS média lekérdezések révén.

Például kisebb betűk méretének csökkentése kisebb eszközökön.

```
@media (min-width: 601px) {  
  
  .my-class {  
  
    font-size: 24px;  
  
  }  
  
}  
  
@media (max-width: 600px) {  
  
  .my-class {  
  
    font-size: 12px;  
  
  }  
  
}
```

A mobil-first stratégia szintén reszponzív, azonban alapértelmezés szerint meg kell határoznunk a mobileszközök összes stílusát, és csak később adjunk hozzá speciális reagálási szabályokat más eszközökhöz. Az előző példát követve:

```
.my-class {  
  
  font-size: 12px;  
  
}  
  
@media (min-width: 600px) {  
  
  .my-class {  
  
    font-size: 24px;  
  
  }  
  
}
```

A mobil-first stratégiának két fő előnye van:

- A mobileszközökön jobban teljesít, mivel a rájuk vonatkozó összes szabályt nem kell semmilyen média lekérdezéssel érvényesíteni
- Tisztább kód megírására kényszerül az reszponzív CSS-szabályok vonatkozásában.

Mondjon példát arra, hogyan használná a számláló növelést (counter-increment) és a visszaállítást (counter-reset) a CSS-ben az automatikus számozás létrehozásához egy weboldalon belül.

Az counter-reset és counter-increment tulajdonságok lehetővé teszik a fejlesztő számára, hogy automatikusan számozza a CSS-elemeket, például egy rendezett listát (). A counter-reset tulajdonság visszaállítja a CSS számlálót egy adott értékre. Az counter-increment tulajdonság ezután egy vagy több számláló értéket növel. A weboldalon belüli automatikus számozás gyakran hasznos, hasonlóan a cikk fejlécéhez. Az alábbiakban látható egy példa a számlálók CSS-ben történő használatára.

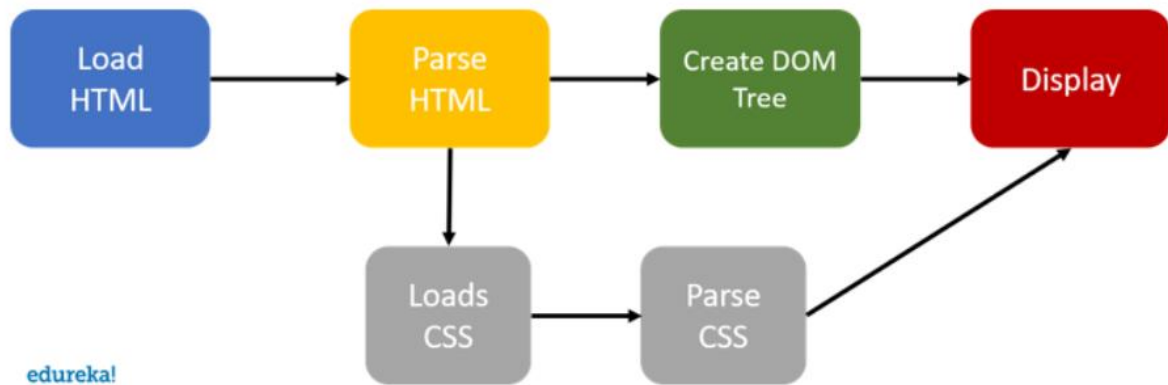
```
body {  
  
  counter-reset: foo;  
  
}  
  
h1 {  
  
  counter-reset: bar;  
  
}  
  
h1:before {  
  
  counter-increment: foo;  
  
  content: "Section " counter(foo) ". ";  
  
}  
  
h2:before {  
  
  counter-increment: bar;  
  
  content: counter(foo) "." counter(bar) " ";  
  
}
```

Hogyan működik a CSS a háttérben?

Amikor a böngésző megjeleníti a dokumentumot, akkor kombinálnia kell a dokumentum tartalmát a stílusinformációival. A dokumentumot két szakaszban dolgozza fel:

A böngésző a HTML-t és a CSS-t DOM-má (Document Object Model) alakítja. A DOM képviseli a dokumentumot a számítógép memóriájában. Kombinálja a dokumentum tartalmát a stílusával.

A böngésző megjeleníti a DOM tartalmát.



Weblap optimalizálása nyomtatáshoz?

- Hozzon létre egy stíluslapot a nyomtatáshoz
- Kerülje a felesleges HTML táblákat
- Használjon oldaltöréseket
- Méretezze az oldalt a nyomtatáshoz - max. magasság stb