

# Reszponzív webdizájn

Jeszenszky Péter

2022. november 22.

# Webelérés asztali és mobil eszközökről

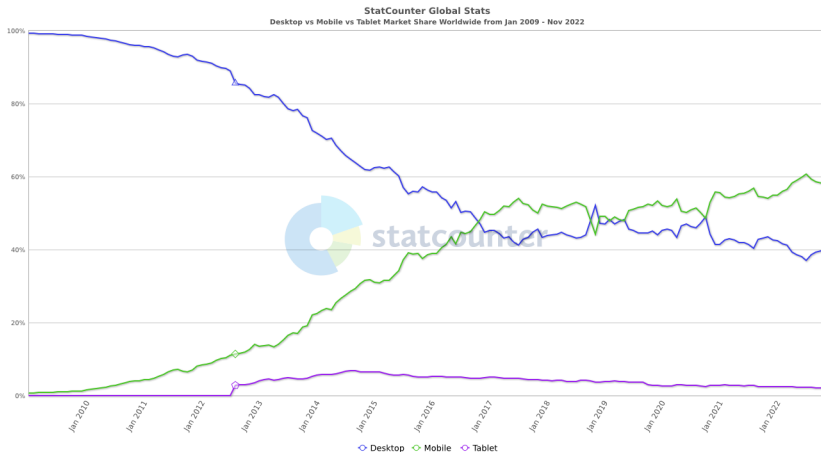


Figure 1: Forrás: StatCounter Global Stats – Desktop vs Mobile vs Tablet Market Share Worldwide

# Mobilegeddon

- A szó a webmesterek és webfejlesztők rémületét fejezi ki, melyet a Google arról szóló bejelentése váltott ki, hogy a keresési eredmények mobil eszközök számára történő rangsorolásánál előnyben részesítik a mobilbarát weboldalakat.
  - Lásd:
    - Chuck Price. [“Mobilegeddon” Is Coming on April 21 – Are You Ready?](#) March 9, 2015.
- Bejelentés:
  - Google Webmaster Central Blog. [Rolling out the mobile-friendly update.](#) April 21, 2015.

# Mobile-First Indexing (1)

- A kifejezés azt jelenti, hogy az indexelésnél és rangsorolásnál a Google túlnyomórészt a tartalmak mobil változatát fogja használni.
  - Lásd: <https://developers.google.com/search/mobile-sites/mobile-first-indexing>
- Bejelentés:
  - Google Webmaster Central Blog. [Rolling out mobile-first indexing](#). March 26, 2018.

# Mobile-First Indexing (2)

Eszközök:

- Google Search Console – Mobilbarát jelleget vizsgáló teszt

# Mobil tartalomszolgáltatás (1)

- A HTTP tartalomegyeztetési mechanizmusa felhasználható ahhoz, hogy ugyanazon erőforrás különböző reprezentációi kerüljenek kiszolgálásra különböző (például mobil és asztali) eszközök számára.
- Az adaptív és reszponzív webdizájn lehetővé teszi az eszközhez igazodó tartalom szolgáltatását.

## Mobil tartalomszolgáltatás (2)

- Még ma is megszokott gyakorlat a mobil webtartalom egy külön webhelyről történő szolgáltatása, mely egy (gyakran `m` nevű, mint például `m.example.com`) altartományon üzemel.
- Példák:
  - Facebook: <https://m.facebook.com/>
  - Twitter: <https://mobile.twitter.com/>
  - Wikipedia: [https://hu.m.wikipedia.org/wiki/Main\\_Page](https://hu.m.wikipedia.org/wiki/Main_Page)
  - YouTube: <https://m.youtube.com/>
  - hvg.hu: <https://m.hvg.hu/>

# Mi a reszponzív webdizájn? (1)

- A reszponzív webdizájn (RWD) egy webfejlesztési megközelítés olyan weblapok készítéséhez, melyek optimálisan néznek ki és viselkednek egy egész sor eszközön, a mobiltelefonoktól kezdve az asztali monitorokig, függetlenül attól, hogy mi az ablak-/képernyőméret vagy a tájolás.
- A reszponzív weboldalak dinamikusan ahhoz az eszközhöz igazodnak, melyen megjelenítésre kerülnek.
  - Megfelelően reagálnak a környezet változásaira, mint például az ablak átméretezése vagy álló tájolásról fekvőre történő váltás.



## Mi a reszponzív webdizájn? (2)

- A “reszponzív webdizájn” kifejezést Ethan Marcotte találta ki 2010-ben.
  - Lásd: Ethan Marcotte. [Responsive Web Design](#). May 25, 2010.
- További olvasnivaló:
  - <https://ethanmarcotte.com/>
  - Ethan Marcotte. [Responsive Web Design](#). 2 nd ed. A Book Apart, 2014.
  - Ethan Marcotte. [Responsive Design: Patterns & Principles](#). A Book Apart, 2015.

# Mi az adaptív webdizájn? (1)

- Az adaptív webdizájn (AWD) ugyanazon weboldal több – különböző eszközökre testreszabott – változatának elkészítését jelenti, és ezek közül a legalkalmasabb kiszolgálását egy adott felhasználói ágensnek.
- Miután egy weboldal betöltésre került, az elrendezése nem igazodik a környezet változásaira.

## Mi az adaptív webdizájn? (2)

Az “adaptív webdizájn” kifejezést Aaron Gustafson találta ki 2011-ben.

- Lásd: Aaron Gustafson. [Adaptive Web Design – Crafting Rich Experiences with Progressive Enhancement](#). Easy Readers, 2011.

# Weboldal elrendezések

Lásd: <https://web.archive.org/web/20190315030321/http://www.liquidapsive.com/>

- **Statikus/rögzített:** abszolút mértékegységben (többnyire pixelben) kifejezett rögzített szélességet használó elrendezés.
- **Folyékony (*fluid/liquid*):** relatív mértékegységben kifejezett szélességet használó elrendezés.
- **Adaptív:** statikus elrendezések egy sorozatként tekinthető.
  - Média lekérdezések révén különböző statikus elrendezéseket határoz meg különböző nézetablak szélességekhez.
- **Reszponzív:** folyékony elrendezések egy sorozatként tekinthető.
  - Média lekérdezések révén különböző folyékony elrendezéseket határoz meg különböző nézetablak szélességekhez.

# Reszponzív webhelyek

Példák:

- [Dropbox](#)
- [Facebook](#)
- [GitHub](#)
- [Microsoft](#)
- [Mozilla](#)
- [Wired](#)
- [YouTube](#)
- ...

# Mi egy pixel? (1)

- Vonatkozó CSS specifikáció:
  - [CSS Values and Units Module Level 3](#) (W3C Candidate Recommendation, 6 June 2019)
- Különbözőek a CSS pixelek és a fizikai pixelek!

# Mi egy pixel? (2)

## Referencia pixel:

- A referencia pixel az a látószög, mely alatt egy pixel látszik egy 96dpi pixelsűrűségű eszközön egy kar távolságról nézve.
  - 28 inch névleges karhossz esetén a látószög ezért nagyjából 0,0213 fok.
  - Egy karhossz távolságról nézve tehát 1px nagyjából 0,26 mm-nek (1/96 inchnek) felel meg.
- Lásd: <https://www.w3.org/TR/css3-values/#reference-pixel>

## Mi egy pixel? (3)

Referencia pixel (folytatás):

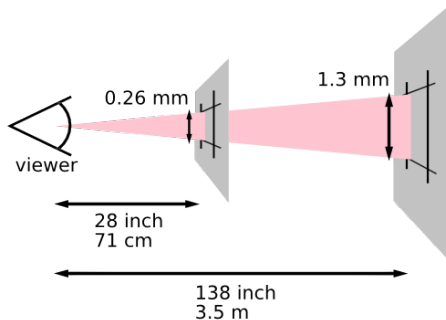


Figure 2: Az ábra a nézési távolság a referencia pixel méretére való hatását szemlélteti: 71 cm (28 inch) távolságról nézve a referencia pixel mérete 0,26 mm, míg 3,5 m (12 láb) távolságnál 1,3 mm



## Mi egy pixel? (4)

- DPI (*dots per inch*), DPCM (*dots per centimeter*), PPI (*pixels per inch*), PPCM (*pixels per centimeter*):
  - A nyomtatók/kijelzők pont-/pixelsűrűségét (felbontását) mérik.
- A DPI (DPCM) és PPI (PPCM) kifejezéseket gyakran felcserélhetőként használják.
- Online eszközök a pixelsűrűség meghatározásához:
  - [DPI love](#)
  - [mydevice.io](#)

# Mi egy pixel? (5)

A CSS abszolút hosszúság mértékegységei:

- Egymáshoz képest rögzítettek és valamilyen fizikai méréshez vannak kötve.

Egység	Név	Ekvivalencia
cm	centiméter	$1\text{cm} = 96\text{px}/2,54$
mm	milliméter	$1\text{mm} = 1\text{cm}/10$
Q	negyvened milliméter	$1\text{Q} = 1\text{cm}/40$
in	inch	$1\text{in} = 2,54\text{cm} = 96\text{px}$
pc	pica	$1\text{pc} = 1\text{in}/6$
pt	pont	$1\text{pt} = 1\text{in}/72$
px	pixel	$1\text{px} = 1\text{in}/96$

## Mi egy pixel? (6)

- Az abszolút hossz mértékegységek kötése az alábbiak szerint történik:
  - a fizikai mértékegységek (úgy mint cm, mm, Q, in, pc, pt) a megfelelő fizikai mérésekhez való kötésével, vagy
  - a px mértékegység a referencia pixelhez való kötésével.
- A px mértékegységet **látószög mértékegységnek** (***visual angle unit***) is nevezik.

# Mi egy pixel? (7)

- Képernyő médiáknál, kis pixelsűrűségű eszközöknél és szokatlan távolságról nézett eszközöknél a px mértékegységhez ajánlott kötni a többi mértékegységet.
  - A px mértékegység azt az egész eszköz pixelszámot jelenti, mely a legjobban közelíti a referencia pixelt.
  - Az ilyen eszközökön a fizikai mértékegységek (például cm, in, mm) lehet, hogy nem egyeznek a fizikai megfelelőikkel!
- Tipikus távolságról nézett nyomtatott médiáknál a szabványos fizikai mértékegységek egyikéhez (például cm, in, mm) ajánlott kötni a többi mértékegységet.
  - Az ilyen eszközökön a px mértékegység lehet, hogy nem egész számú eszközpixelnek felel meg.

## Mi egy pixel? (8)

A `Window` interfész `devicePixelRatio` attribútuma a CSS pixel méretének az eszközpixel méretéhez viszonyított arányát adja vissza az aktuális megjelenítő eszközhöz.

- Lásd: [MDN Web Docs – Window.devicePixelRatio](#)

# Mi egy pixel? (9)

Demonstráció: <https://pixel-example.surge.sh/>

# Mi egy pixel? (10)

További ajánlott olvasnivaló:

- Peter-Paul Koch. [A pixel is not a pixel is not a pixel](#). 20 April 2010.
- Elad Shechter. [Understanding the Difference Between CSS Resolution and Device Resolution](#). 17 November 2019.

# A CSS relatív hosszúság mértékegységei

- `em`: a `font-size` tulajdonság az elemhez tartozó számított értékét jelenti.
- `rem`: az `em` egység a gyökérelemhez tartozó számított értékét jelenti.
- `vw`, `vh`, `vmin`, `vmax`: lásd később.



# A reszponzív webdizájn alkotóelemei

- A reszponzív webdizájn fő komponensei:
  - Média lekérdezések
  - Nézetablak
  - Folyékony rácsok
  - Rugalmas helyettesített elemek
  - Adaptív/reszponzív képek
- Lásd: [MDN Web Docs – The building blocks of responsive design](#)

# Nézetablak (*viewport*)

- Folytonos médiákhoz a felhasználói ágensek általában egy nézetablakot (egy ablakot vagy egyéb területet a kijelzőn) nyújtanak, melyen keresztül a felhasználó nézheti a dokumentumot.
  - A felhasználói ágensek megváltoztathatják a dokumentum elrendezését a nézetablak átméretezésekor.
  - Amikor a nézetablak kisebb a vászon azon területénél, melyen a dokumentum megjelenítésre kerül, a felhasználói ágensnek egy görgetési mechanizmust kellene felajánlania.
  - Vásznanként legfeljebb egy nézetablak van, de a felhasználói ágens egyenél több vásznat is használhat a megjelenítéshez (azaz különböző nézeteket szolgáltatathat ugyanarról a dokumentumról).
- Lásd: [CSS 2.1 Specification – Visual formatting model – The viewport](#)

# CSS nézetablak mértékegységek

- Vonatkozó specifikáció:
  - [CSS Values and Units Module Level 3](#) (W3C Candidate Recommendation, 6 June 2019)
- Nézetablak-arány mértékegységek:
  - vw: a nézetablak szélességének 1%-ával egyenlő.
  - vh: a nézetablak magasságának 1%-ával egyenlő.
  - vmin: a vw és vh közül a kisebbel egyenlő.
  - vmax: a vw és vh a nagyobbbal egyenlő.

# Mik a média lekérdezések? (1)

- Egy média lekérdezés egy módszer a felhasználói ágens vagy eszköz bizonyos jellemzőinek vizsgálatára, melyen a dokumentum megjelenítésre kerül.
- Stíluslapok feltételes alkalmazására szolgálnak.

## Mik a média lekérdezések? (2)

- A média lekérdezések (majdnem) mindig függetlenek a megjelenített dokumentumtól.
- Csak “külső” információktól függenek, hacsak nem írja elő egy másik lehetőség kifejezetten azt, hogy hatással van a feloldásukra.
  - Lásd például a `@viewport at`-szabályt.

# Média lekérdezés specifikáció

- Az aktuális szabvány:
  - [Media Queries Level 3](#) (W3C Recommendation, 5 April 2022)
- A szabvány következő verziója:
  - [Media Queries Level 4](#) (W3C Candidate Recommendation Draft, 25 December 2021)

# Média lekérdezés támogatás

Lásd: <https://caniuse.com/css-mediaqueries>

# Média lekérdezések rendelkezésre állása

Média lekérdezések az alábbi nyelvekben használhatók:

- CSS
- XML
- HTML
- JavaScript



# Média lekérdezések CSS-ben (1)

@import:

- Vonatkozó specifikáció:
  - [CSS Cascading and Inheritance Level 3](#) (W3C Recommendation, 11 February 2021)
    - Lásd: [Importing Style Sheets: the @import rule](#)
- Példa:

```
@import "screen.css" screen;  
@import "print.css" print;
```

## Média lekérdezések CSS-ben (2)

### @import: (folytatás)

- Az opcionális média lekérdezés lista a feltételeket fejezi ki, melyek teljesülése esetén alkalmazható az @import.
- Ha a média lekérdezés nem illeszkedik, akkor az importált stíluslap szabályai nem lesznek alkalmazva, pontosan úgy, mintha a stíluslap az adott média lekérdezést tartalmazó @media blokkba lenne csomagolva.

# Média lekérdezések CSS-ben (3)

@media:

- Vonatkozó specifikáció:
  - [CSS Conditional Rules Module Level 3](#) (W3C Candidate Recommendation Snapshot, 13 January 2022)
    - Lásd: [Media-specific style sheets: the @media rule](#)
- Példa:

```
@media screen {  
  * {  
    font-family: sans-serif;  
  }  
}
```

# Média lekérdezések CSS-ben (4)

## @media: (folytatás)

- A @media szabály egy olyan feltételes csoportszabály, melynek feltétele egy média lekérdezés.
- A feltételes csoportszabályok egy feltételt társítanak más CSS szabályok egy csoportjához.
  - Minden egyes feltételes csoportszabálynak van egy feltétele, mely bármikor igaz vagy hamis.
  - Amikor a feltétel igaz, a CSS feldolgozóknak alkalmaznia kell a csoportszabályon belüli szabályokat, mintha a csoportszabály helyén lennének. Amikor a feltétel hamis, a CSS feldolgozók számára tilos a csoportszabályon belüli szabályok alkalmazása.

# Média lekérdezések XML-ben

Vonatkozó specifikáció:

- [Associating Style Sheets with XML documents 1.0 \(Second Edition\)](#)  
(W3C Recommendation, 28 October 2010)
  - Az `xml-stylesheet` feldolgozási utasítás `media` pszeudo-attribútuma a médiát adja meg, melyre a hivatkozott stíluslap vonatkozik.
- Példa:

```
<?xml-stylesheet type="text/css" media="screen" href="screen.css"?>  
<?xml-stylesheet type="text/css" media="print" href="print.css"?>
```

# Média lekérdezések HTML-ben

- Vonatkozó specifikáció: [HTML Living Standard](#)
- Lásd a `media` attribútumot, mely a következő elemeken adható meg:
  - `link`
  - `meta`
  - `source`
  - `style`
- Példa:

```
<link rel="stylesheet" media="screen" href="screen.css">  
<link rel="stylesheet" media="print" href="print.css">
```

# Média lekérdezések JavaScript-ben (1)

- Vonatkozó specifikáció: [CSSOM View Module](#) (W3C Working Draft, 17 March 2016)
- A `MediaQueryList` interfész lehetővé teszi egy média lekérdezés kiértékelését programkódból.
  - Eseményfigyelők regisztrálhatók, melyek értesítve lesznek a média lekérdezés eredményének változásairól.
  - Lásd: [The MediaQueryList Interface](#)
- Támogatás: <https://caniuse.com/?search=mediaquerylist>

## Média lekérdezések JavaScript-ben (2)

Példa: (forrás: [MDN Web Docs – Testing media queries programmatically](#))

```
// Média lekérdezés lista létrehozása:  
var mql = window.matchMedia("(orientation: portrait)");  
  
// A lekérdezés eredményének vizsgálata:  
if (mql.matches) {  
    // A nézetablak jelenleg álló tájolású  
} else {  
    // A nézetablak jelenleg fekvő tájolású  
}
```



## Média lekérdezések JavaScript-ben (3)

Példa: (folytatás)

```
// Média lekérdezés lista létrehozása:
var mql = window.matchMedia("(orientation: portrait)");

// Callback függvény definiálása az eseményfigyelőhöz:
function handleOrientationChange(event) {
  if (event.matches) {
    // A nézetablak jelenleg álló tájolású
  } else {
    // A nézetablak jelenleg fekvő tájolású
  }
}

// A callback függvény regisztrálása eseményfigyelőként:
mql.addListener(handleOrientationChange);
```

# Mértékegységek média lekérdezésekben

- A média lekérdezésekben használt mértékegységek ugyanazok, mint a CSS más részeiben.
- A relatív hosszúság mértékegységek a média lekérdezésekben a kezdőértékükön alapulnak, mely azt jelenti, hogy ezeket soha nem befolyásolják deklarációk eredményei.
  - Például HTML-ben az `em` mértékegység a `font-size` tulajdonság kezdőértékéhez relatív, melyet a felhasználói ágens vagy a felhasználó beállításai határoznak meg, nem pedig az oldal stílusa.

# Média lekérdezés szintaxis (1)

Egy média lekérdezés egy opcionális módosítóból, egy opcionális média típusból és egy média feltételből áll:

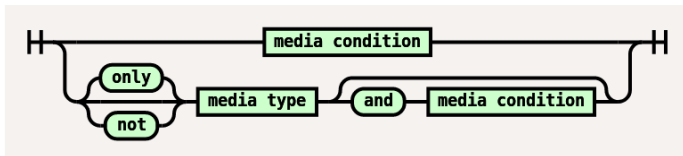


Figure 3: Média lekérdezés szintaxis

## Média lekérdezés szintaxis (2)

Több média lekérdezés egy vesszővel elválasztott média lekérdezés listává kombinálható:

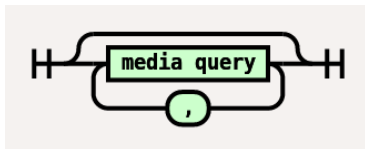


Figure 4: Média lekérdezés lista szintaxis

# Média lekérdezés kiértékelés (1)

- Egy média lekérdezés egy logikai kifejezés, mely igaz vagy hamis. Egy média lekérdezés igaz, ha teljesülnek az alábbiak:
  - Ha meg van adva a média típus, akkor az illeszkedik annak az eszköznek a média típusára, ahol a felhasználói ágens fut.
  - A média feltétel igaz.
- Egy média lekérdezés lista igaz, ha valamely komponens média lekérdezése igaz, és csak akkor hamis, ha minden komponens média lekérdezése hamis.

## Média lekérdezés kiértékelés (2)

- A felhasználói ágenseknek újra ki kell értékelniük a média lekérdezéseket válaszul a felhasználói környezetben történt olyan változásokra, melyeknek tudatában vannak, és megfelelően kell változtatniuk valamennyi, a szóban forgó média lekérdezésektől függő konstrukció viselkedését.
- Ha például egy eszköz fekvőről álló tájolására dől.

# Média lekérdezés módosítók (1)

Egy média lekérdezés elején opcionálisan elhelyezhető egy média lekérdezés módosító, mely egy olyan kulcsszó, mely a következő média lekérdezés jelentését módosítja.

## Média lekérdezés módosítók (2)

not módosító:

- Egy önálló média lekérdezés eredménye negálható az elejére a not kulcsszót írva.
- Ha a média lekérdezés értéke egyébként igaz lenne, az elejére írt not hamis értéket eredményez, és fordítva.
- Példa:

```
<link rel="stylesheet" media="not screen and (color)"  
      href="style.css">
```



# Média lekérdezés módosítók (3)

only módosító:

- Az `only` kulcsszó a média lekérdezés elejére írásával rejthető el a média lekérdezés az ősi felhasználói ágensek előtt.
- Az `only` kulcsszónak nincs hatása a média lekérdezés eredményére, de az ősi felhasználói ágensek úgy fogják a média lekérdezést értelmezni, mintha az az `only` ismeretlen média típust írná elő és így figyelmen kívül hagyják.
- Példa:

```
<link rel="stylesheet" media="only screen and (color)"  
      href="style.css">
```

# Média típusok (1)

- Egy média típus felhasználói ágens eszközök egy széles csoportja, melyeken megjeleníthető egy dokumentum.
- Nem összetévesztendő az IANA média típusokkal, mint például `text/html`!

## Média típusok (2)

A következő média típusok definiáltak média lekérdezésekben való felhasználásra:

- `all`: az összes eszközre illeszkedik.
- `print`: a nyomtató eszközökre illeszkedik és az olyan eszközökre, melyek célja egy nyomtatott megjelenítés visszaadása, mint például egy dokumentum nyomtatási képét mutató böngésző (*print preview*).
- `screen`: a `print`-re vagy `speech`-re nem illeszkedő összes eszközre illeszkedik.
- `speech`: a képernyőolvasókra és hasonló eszközökre illeszkedik, melyek hangosan olvasnak fel egy oldalt.

## Média típusok (3)

- Továbbá a következő elavult média típusok is definiáltak: tty, tv, projection, handheld, braille, embossed, aural.
- A szerzők számára tilos ezen média típusok használata, helyettük megfelelő média jellemzők választása javasolt, melyek jobban fejezik ki annak az eszköznek a jellegét, melyre stílust próbálnak megadni.
- A felhasználói ágensek érvényesként kell, hogy felismerjék ezeket a média típusokat, de semmit sem illeszthetnek rájuk.

# Média jellemzők (1)

Egy média jellemző (*media feature*) egy, a média típusoknál finomabb teszt, mely a felhasználó ágens vagy megjelenítő eszköz egy bizonyos jellemzőjét vizsgálja.

- Példák: width, height, orientation, resolution

## Média jellemzők (2)

- A média jellemzők szintaktikailag a CSS tulajdonságokra hasonlítanak: egy jellemző névből, egy kettőspontból és egy értékből állnak.
- Logikai alakban is írhatók, mint egyetlen jellemző név, vagy egy összehasonlító művelettel tartomány alakban.

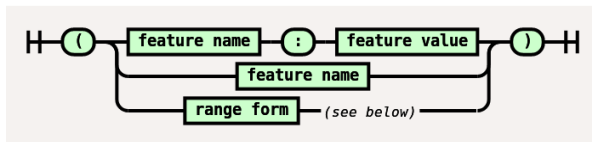


Figure 5: Média jellemző szintaxis

## Média jellemzők (3)

Eltérések a tulajdonságok és a média jellemzők között:

- A tulajdonságok célja az, hogy információkat adjanak arról, hogy a dokumentum hogyan jelenjen meg. A média jellemzők az output eszköz követelményeinek leírására szolgálnak.
- A média lekérdezéseket mindig zárójelek közé kell zárni és az `and` vagy `or` kulcsszavakkal kombinálhatók, mint például `(color) and (min-width: 600px)`, nem pedig pontosvesszőkkel kerülnek elválasztásra.
- Egy média jellemző megadható csupán a nevével (a kettőspont és az érték elhagyásával) a jellemző logikai kontextusban történő kiértékeléséhez. Ez egy kényelmes rövid kifejezőmód olyan jellemzőkhöz, melyek értékeként elfogadható 0 vagy `none`.
  - Például `(color)` akkor igaz, ha a `color` média jellemző értéke nem nulla.

# Média jellemzők (4)

Eltérések a tulajdonságok és a média jellemzők között: (folytatás)

- A tartomány típusú média jellemzők tartomány kontextusban írhatók, mely a kettőspont helyett a szabványos matematikai összehasonlító operátorokat használja, vagy a nevük elé  $\min$ - vagy  $\max$ - előtag helyezhető el.
- A tulajdonságok néha elfogadnak összetett értékeket, például több más értéket tartalmazó számításokat. A média jellemzők egyetlen értéket fogadnak el, például egy kulcsszót vagy számot.



## Média jellemzők (5)

- Ha egy média jellemző egy olyan fogalomra hivatkozik, mely nem létezik azon az eszközön, melyen a felhasználói ágens fut, akkor a média lekérdezés értéke mindig hamis kell, hogy legyen.
- Például hangos felhasználói ágenseknél nem értelmezett a szélesség (`width`).

# Média jellemző típusok

- Minden média jellemző meghatározza a típusát, mely tartomány vagy diszkrét.
  - A diszkrét média jellemzők egy halmazból veszik fel értéküket.
    - Az értékek lehetnek kulcsszavak vagy logikai számok (0 és 1), de az bennük a közös, hogy nem értelmezett rajtuk rendezés.
  - A tartomány média jellemzők egy tartományból veszik fel értéküket.
    - Bármely két érték összehasonlítható annak megállapításához, hogy melyik a kisebb és melyik a nagyobb.
- A két típus közötti egyetlen lényeges különbség az, hogy a tartomány média jellemzők kiértékelhetők logikai kontextusban és a nevük előtt megadható  $\min$ - vagy  $\max$ - előtag.

# Média jellemzők kiértékelése logikai kontextusban

- A média jellemzők írhatók egyszerűen csupán a nevükkel, mint például `color`.
- Az ebben az alakban való megadásnál a média jellemző logikai kontextusban kerül kiértékelésre.
  - Ha a jellemző igaz lenne a 0 számtól, egy 0 értékű dimenziótól vagy a `none` kulcsszótól eltérő bármely más értékre, akkor a média jellemző értéke igaz.
  - Egyébként hamis az értéke.

# Média jellemzők kiértékelése tartomány kontextusban (1)

A tartomány típusú média jellemezők tartomány kontextusban is írhatók, mely azt használja ki, hogy az értékeik a szokásos matematikai összehasonlító operátorok segítségével rendezettek:

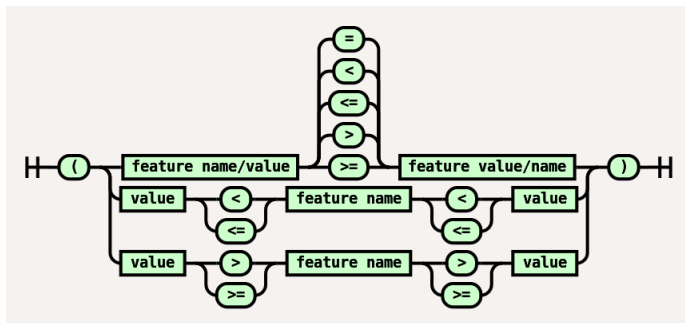


Figure 6: Média jellemző szintaxis tartomány kontextusban

## Média jellemzők kiértékelése tartomány kontextusban (2)

- Az egy jellemző névből, egy összehasonlító operátorból és egy értékből álló egyszerű alak igazat ad vissza, ha a reláció igaz.
  - Példa: `(height > 600px)`, `(600px < height)`
- A többi alak, ahol a jellemző neve két érték összehasonlítás közé van ágyazva, igazat ad vissza, ha mindkét összehasonlítás igaz.
  - Példa: `(400px < width < 1000px)`

# Média jellemzők kiértékelése tartomány kontextusban (3)

- Támogatás:
  - A Firefox 2018 óta támogatja.
  - A Chromium-alapú böngészőkben csak mostanában került támogatásra.
  - A Safari nem támogatja jelenleg.
- Lásd: <https://caniuse.com/css-media-range-syntax>

# A min- és max- előtagok használata tartomány típusú jellemzőkön

- Egy tartomány típusú média jellemző a tartomány kontextusban történő kiértékelés helyett írható közönséges média jellemzőként, de a neve előtt a min- vagy max- előtaggal.
  - A min- előtag használata egy jellemző néven ekvivalens a  $\geq$  operátor használatával.
    - Például (`min-height: 600px`) ugyanaz, mint (`height  $\geq$  600px`).
  - A max- előtag használata egy jellemző néven ekvivalens a  $\leq$  operátor használatával.
    - Például (`max-width: 40em`) ugyanaz, mint (`width  $\leq$  40em`).
- Diszkrét típusú média jellemzőkhöz nem adható meg min- vagy max-előtag.

# Média jellemzők kombinálása

- Több média jellemző egy média feltétellé kombinálható össze a `not/and/or` logikai műveletekkel.
  - Bármely média jellemző negálható elé `not` helyezésével, mint például `not(color)`.
  - Két vagy több média jellemző összefűzhető közéjük `and/or` helyezésével.
    - Példa: `(width < 600px) and (height < 600px), update: slow) or (hover: none)`
  - Média feltételek csoportosíthatók zárójelek közé zárással, mint például `not((color) or (hover))`.
- Nem megengedett egy média lekérdezés ugyanazon “szintjén” keverni az `and/or/not` logikai műveleteket.
  - Például `(color) and (pointer) or (hover)` érvénytelen.



# Hibakezelés média lekérdezésekben

Egy szintaktikailag érvénytelen média lekérdezést az elemzés közben `not all-ra` kell kicserélni.

# Rendelkezésre álló média jellemzők (1)

Nézetablak/oldal dimenzió média jellemzők:

- `aspect-ratio: <ratio>`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/aspect-ratio>
- `height: <length>`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/height>
- `orientation: portrait | landscape`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/orientation>
- `width: <length>`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/width>

# Rendelkezésre álló média jellemzők (2)

## Megjelenítési minőségi média jellemzők:

- `grid: 0 | 1`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/grid>
- `overflow-block: none | scroll | optional-paged | paged`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/overflow-block>
- `overflow-inline: none | scroll`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/overflow-inline>
- `resolution: <resolution> | infinite`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/resolution>
- `scan: interlace | progressive`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/scan>
- `update: none | slow | fast`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/update-frequency>

# Rendelkezésre álló média jellemzők (3)

## Szín média jellemzők:

- `color: <integer>`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/color>
- `color-index: <integer>`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/color-index>
- `color-gamut: srgb | p3 | rec2020`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/color-gamut>
- `monochrome: <integer>`  
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/monochrome>

# Rendelkezésre álló média jellemzők (4)

## Interakciós média jellemzők:

- `any-hover: none | hover`

<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/any-hover>

- `any-pointer: none | coarse | fine`

<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/any-pointer>

- `hover: none | hover`

<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/hover>

- `pointer: none | coarse | fine`

<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/pointer>

# Tipikus média lekérdezés töréspontok

## Példa: Bootstrap – Responsive breakpoints

```
// Extra small devices (portrait phones, less than 576px):  
// No media query for `xs` since this is the default in Bootstrap  
/* ... */  
  
// Small devices (landscape phones, 576px and up):  
@media (min-width: 576px) { /* ... */ }  
  
// Medium devices (tablets, 768px and up):  
@media (min-width: 768px) { /* ... */ }  
  
// Large devices (desktops, 992px and up):  
@media (min-width: 992px) { /* ... */ }  
  
// Extra large devices (large desktops, 1200px and up):  
@media (min-width: 1200px) { /* ... */ }  
  
// XX-Large devices (larger desktops, 1400px and up):  
@media (min-width: 1400px) { /* ... */ }
```

# Média lekérdezések – további olvasnivaló

- MDN Web Docs – Media queries
- MDN Web Docs – Using media queries
- MDN Web Docs – Using Media Queries for Accessibility

# Nézetablak (1)

- A mobil/kézi eszközök böngészőinek nézetablaka általában sokkal keskenyebb egy, a CSS által ajánlott CSS pixelméretet adó nagyítási szinten, mint egy asztali böngészőablaké.
- A keskeny nézetablak gondot jelent olyan dokumentumoknál, melyeket úgy terveztek, hogy az asztali böngészőkben nézzenek ki jól.



## Nézetablak (2)

- Ezért a mobil böngészők két nézetablakot tartanak fenn:
  - **Látható nézetablak** (***visual viewport***): az oldal a kijelzőn látható része.
  - **Elrendezési nézetablak** (***layout viewport***): a nézetablak, melybe a böngésző kirajzol egy oldalt.
- Lásd:
  - [MDN Web Docs Glossary – Viewport](#)
  - [MDN Web Docs Glossary – Visual Viewport](#)
  - [MDN Web Docs Glossary – Layout Viewport](#)

## Nézetablak (3)

Látható nézetablak

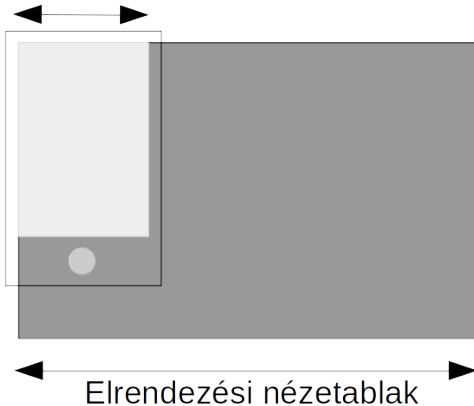


Figure 7: A két nézetablak

## Nézetablak (4)

Böngésző nézetablak méretek lekérdezése JavaScript-ben:

- Elrendezési nézetablak szélesség és magasság: `window.innerWidth`, `window.innerHeight`
  - Lásd: <https://developer.mozilla.org/en-US/docs/Web/API/Window>
- Látható nézetablak szélesség és magasság: `window.visualViewport.width`, `window.visualViewport.height`
  - Lásd: <https://developer.mozilla.org/en-US/docs/Web/API/Window/visualViewport>

## Nézetablak (5)

Megszokott nézetablak méretek mobil és tablet eszközökhöz:

- [mydevice.io](https://mydevice.io) – web devices capabilities
- Screen Sizes – Viewport Sizes and Pixel Densities for Popular Devices

## Nézetablak (6)

- A viewport meta címkét abból a célból vezették be, hogy lehetővé tegye a szerző számára az (elrendezési) nézetablak méretének és a kezdeti nagyítási arány közvetlen megadását.
- Először az Apple implementálta a Safari/iPhone böngészőhöz, de azóta sok más mobil böngészőhöz is implementálták.
- Példa:

```
<meta name="viewport"  
      content="width=device-width, initial-scale=1">
```

## Nézetablak (7)

A viewport meta elem jelenlétét/hiányát szemléltető példák:

<https://jeszy75.github.io/viewport-example/>

<https://github.com/jeszy75/viewport-example/>

## Nézetablak (8)

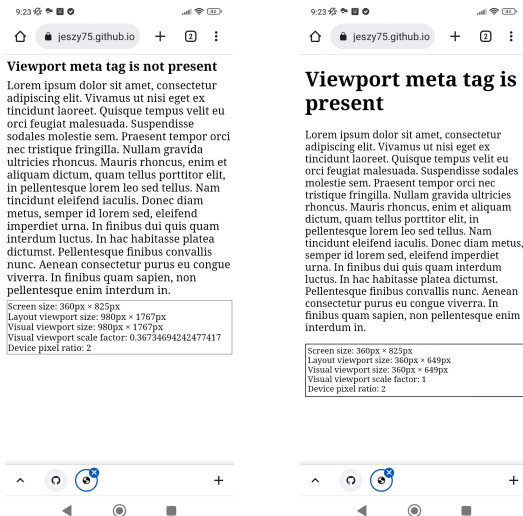


Figure 8: A viewport meta címke hiányának/jelenlétének hatása

# Nézetablak (9)

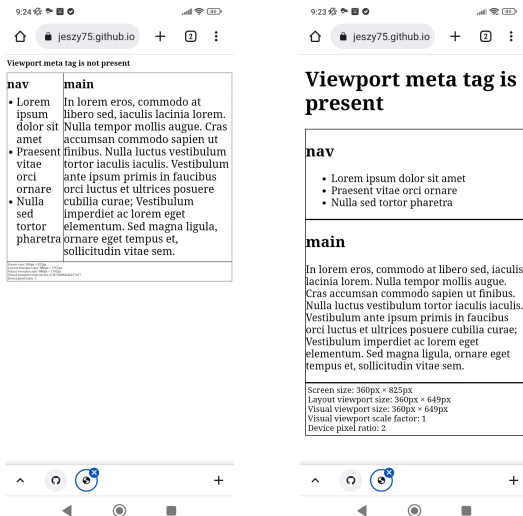


Figure 9: A viewport meta címke hiányának/jelenlétének hatása



# Nézetablak (10)

További olvasnivaló:

- [MDN Web Docs – Viewport meta tag](#)
- Peter-Paul Koch. [A tale of two viewports – part one](#).
- Peter-Paul Koch. [A tale of two viewports – part two](#).
- Peter-Paul Koch. [The Mobile Web Handbook](#). Smashing Magazine, 2014.
- [Safari Web Content Guide – Configuring the Viewport](#)

# Folyékony rács

- A RWD egy rugalmas rács-alapú elrendezést igényel, ahol a “rács” kifejezés nagyon általánosan értendő.
  - Sok létező CSS keretrendszer segítségével megvalósítható rács.
  - Napjainkban már szabványos módszerek is léteznek (lásd a *flexbox* és a rács elrendezéseket).
- Lásd:
  - Ethan Marcotte. [Fluid Grids](#). March 03, 2009.

# Flexbox elrendezés (1)

Vonatkozó specifikáció:

- [CSS Flexible Box Layout Module Level 1](#) (W3C Candidate Recommendation, 19 November 2018)

## Flexbox elrendezés (2)

- Felhasználói felületek tervezéséhez optimalizált CSS dobozmodell.
- A flex elrendezési modellben egy flex konténer gyermekei bármely irányban elrendezhetők és rugalmasan változtathatják a méretüket: nyúlhatnak a nem használt hely kitöltéséhez vagy összezsugorodhatnak elkerülendő azt, hogy túlcsorduljanak a szülőn.
- Könnyedén kezelhető a gyermekek vízszintes és függőleges igazítása is.
- Ilyen dobozok egymásba ágyazása felhasználható elrendezések két dimenzióban történő létrehozásához.

## Flexbox elrendezés (3)

- Támogatás: <https://caniuse.com/flexbox>
- További olvasnivaló:
  - MDN Web Docs – CSS Flexible Box Layout
  - MDN Web Docs – Learn web development – Flexbox

# Flexbox elrendezés (4)

- Játsszóterek az elrendezéssel való játszódáshoz:

- <https://flexbox.tech/>
- <https://flexiting.com/playground>
- <https://flexbox-css.com/>

- Mintapéldák:

- <https://jeszy75.github.io/responsive-layout-examples/>  
<https://github.com/jeszy75/responsive-layout-examples>

# Rács elrendezés (1)

Vonatkozó specifikáció:

- [CSS Grid Layout Module Level 1](#) (W3C Candidate Recommendation Draft, 18 December 2020)

## Rács elrendezés (2)

- Egy felhasználói felületek tervezéséhez optimalizált kétdimenziós rács-alapú elrendezési rendszert határoz meg.
- A rács elrendezési modellben egy rács konténer gyermekei tetszőleges rekeszekbe helyezhetők egy előre definiált rugalmas vagy rögzített méretű elrendezési rácson.
- A rács elrendezés lehetővé teszi a dobozok és tartalmuk méretezésének és elhelyezésének vezérlését.
- Folyékony rács:
  - Az `fr` mértékegység egy rugalmas hosszt ad meg, mely a rács konténerben fennmaradó hely egy hányadát ábrázolja.
  - Lásd: <https://www.w3.org/TR/css-grid-1/#fr-unit>



## Rács elrendezés (3)

A *flexbox*-szal ellentétben, mely egytengelyű, a rács elrendezés olyan 2 dimenziós elrendezésekhez optimalizált, melyeknél a tartalom igazítása mindkét irányban szükséges.

# Rács elrendezés (4)

- Támogatás: <https://caniuse.com/css-grid>
- További olvasnivaló:
  - MDN Web Docs – CSS Grid Layout
  - MDN Web Docs – Learn web development – Grids

# Rács elrendezés (5)

- Játsszóterek az elrendezéssel való játszódáshoz:
  - CSS Grid Playground
  - Grid Garden
  - Interactive CSS Grid Generator
- Mintapéldák:
  - <https://github.com/jeszy75/responsive-layout-examples>  
<https://jeszy75.github.io/responsive-layout-examples/>

# Rugalmas helyettesített elemek (1)

**Helyettesített elem:** egy olyan elem, melynek tartalma kívül esik a CSS formázási modellen, azaz tartalmára nincs hatással az aktuális dokumentum stílusa.

- Példák: audio, iframe, img, video
- Lásd: [CSS 2.1 Specification – Conformance: Requirements and Recommendations](#)

## Rugalmas helyettesített elemek (2)

- A `width` tulajdonság 100%-ra történő beállításával nyújtunk egy helyettesített elemet (például egy képet), hogy töltsse ki a tartalmazó elemet.
- A `max-width` tulajdonság 100%-ra történő beállításával előzzük meg, hogy egy helyettesített elem (például egy kép) szétnyomja a tartalmazó elemet.
- Példa:

```
img {  
  max-width: 100%;  
}
```

# Adaptív/reaktív képek (1)

Lásd:

- [HTML Living Standard – Images](#)
- [MDN Web Docs – Responsive images](#)
- [MDN Web Docs – <img>: The Image Embed element](#)

## Adaptív/respónzív képek (2)

- Számos olyan helyzet van, ahol a szerző több képerőforrást kívánhat használni, melyek közül a felhasználói ágens választhat:
  - Eltérhetnek a különböző felhasználók környezeti jellemzői, mint például
    - a fizikai kijelzőméret,
    - a kijelző pixelsűrűsége,
    - a nagyítási szint,
    - a kijelző tájolása,
    - a hálózati sebesség vagy sávszélesség.
  - A szerző ugyanazt a képtartalmat eltérő méretben kívánhatja megjeleníteni, általában a nézetablak szélességétől függően (nézetablak-alapú kiválasztás).
  - A szerző eltérő képtartalmat kívánhat mutatni a megjelenített mérettől függően ("művészi rendezés").
- A fenti helyzetek nem zárják ki egymást kölcsönösen.

## Adaptív/reszponzív képek (3)

Az srcset és sizes attribútumok:

- Az img és source elemeken adhatók meg.
- Alternatív erőforrások egy halmazának megadására szolgálnak, melyből a felhasználói ágens a legmegfelelőbbet választhatja használatra.



# Adaptív/reszponzív képek (4)

Az srcset attribútum:

- Az attribútum értéke egy vagy több olyan sztring vesszővel elválasztott listája, melyek mindegyike a következő komponensekből áll:
  - Egy kép URI-ja.
  - Opcionálisan az alábbi leírók egyike:
    - **Szélesség leíró:** egy pozitív egész, melyet  $w$  követ közvetlenül, és amely a kép tényleges szélességét adja meg pixelben.
    - **Pixelsűrűség leíró:** egy pozitív lebegőpontos szám, melyet  $x$  követ.
- Ha egy képhez nincs megadva leíró, akkor  $1x$  az alapértelmezés.

# Adaptív/reszponzív képek (5)

A `sizes` attribútum:

- Értéke egy vagy több olyan sztring vesszővel elválasztott listája, melyek mindegyike a következő, *whitespace* karakterekkel elválasztott komponensekből áll:
  - Egy média feltételből, melyet el kell hagyni a lista utolsó eleménél.
  - Egy forrás méret (egy nemnegatív hosszúság), mely a kép kívánt megjelenítési szélességét adja meg.

## Adaptív/reszponzív képek (6)

A picture elem:

- **Nulla vagy több** source elemet és **pontosan egy** azokat követő img elemet tartalmaz, melyek ugyanannak a képnek több változatát szolgáltatják eltérő eszközökhöz/felhasználáshoz.
  - A böngésző minden egyes source elemgyermeket figyelembe vesz és a legjobban illeszkedőt választja, ha egyik sem megfelelő, akkor pedig az img elem src attribútumát választja.
  - A kiválasztott kép ezután az img elem által elfoglalt helyen kerül megjelenítésre.
- Lásd:
  - [HTML Living Standard – The picture element](#)
  - [MDN Web Docs – <picture>: The Picture element](#)

# Adaptív/reaktív képek (7)

Mintapéldák:

<https://jeszy75.github.io/responsive-image-examples/>

<https://github.com/jeszy75/responsive-image-examples>

# Adaptív/reszponzív képek: eszköz pixelarány-alapú kiválasztás, amikor a kép megjelenített mérete rögzített (1)

A felhasználói ágens az adott erőforrások bármelyikét választhatja a felhasználó kijelzőjének pixelsűrűségétől, a nagyítási szinttől és esetleg egyéb információktól függően, mint például a felhasználó hálózati körülményei.

## Adaptív/respónzív képek: eszköz pixelarány-alapú kiválasztás, amikor a kép megjelenített mérete rögzített (2)

Példa:

```
<style>
  img#cat {
    width: 320px;
  }
</style>


```

# Adaptív/reszponzív képek: nézetablak-alapú kiválasztás (1)

- Több olyan kép adott, melyek csak a méretükben térnek el.
- A felhasználói ágens kiszámítja minden egyes kép tényleges pixelsűrűségét a megadott `w` leírókból és a `sizes` attribútumban megadott megjelenített méretből.
- Ezután az adott erőforrások bármelyikét választhatja a felhasználó kijelzőjének pixelsűrűségétől, a nagyítási szinttől és esetleg egyéb információktól függően, mint például a felhasználó hálózati körülményei.

# Adaptív/reszponzív képek: nézetablak-alapú kiválasztás (2)

Példa:

```

```



# Adaptív/reszponzív képek: művészi rendezés-alapú kiválasztás

- A picture elem és source elem a media attribútummal együtt felhasználható több eltérő tartalmú kép szolgáltatására.
  - Például egy kisebb kép lehet egy nagyobb kép levágott változata.
- Példa:

```
<picture>  
  <source media="(min-width: 1200px)" srcset="cat-desktop.jpg">  
  <source media="(min-width: 768px)" srcset="cat-tablet.jpg">  
    
</picture>
```

# Adaptív/reszponzív képek: képformátum-alapú kiválasztás

- A type attribútum a source elemen felhasználható több különböző formátumú kép szolgáltatására.
- Példa:

```
<picture>  
  <source srcset="cat.avif" type="image/avif">  
  <source srcset="cat.webp" type="image/webp">  
    
</picture>
```

# Eszközök (1)

Böngészőeszközök:

- Firefox: [Responsive Design Mode](#)
- Chromium, Google Chrome: [Device Mode \(Chrome DevTools\)](#)
- Chromium-based Edge: [Device Emulation \(Microsoft Edge DevTools\)](#)

## Eszközök (2)

Kiwi: a Chromium-on és a WebKit-en alapuló mobil böngésző, mely webfejlesztői eszközöket (!) is tartalmaz

- Webhely: <https://kiwibrowser.com/>
- Tároló: <https://github.com/kiwibrowser/src>
- Platform: Android
- Licenc: New BSD License

# Eszközök (3)

- Screen Size Map

# Reszponzív keretrendszerek

- Bootstrap (licenc: MIT License) <https://getbootstrap.com/>  
<https://github.com/twbs/bootstrap>
- Bulma (licenc: MIT License) <https://bulma.io/>  
<https://github.com/jgthms/bulma>
- Foundation (licenc: MIT License) <https://get.foundation/>  
<https://github.com/foundation/foundation-sites>
- Pure (licenc: New BSD License) <https://purecss.io/>  
<https://github.com/pure-css/pure>
- Tailwind CSS (licenc: MIT License) <https://tailwindcss.com/>  
<https://github.com/tailwindlabs/tailwindcss>