

Reszponzív webdizájn

Jeszenszky Péter

Debreceni Egyetem, Informatikai Kar

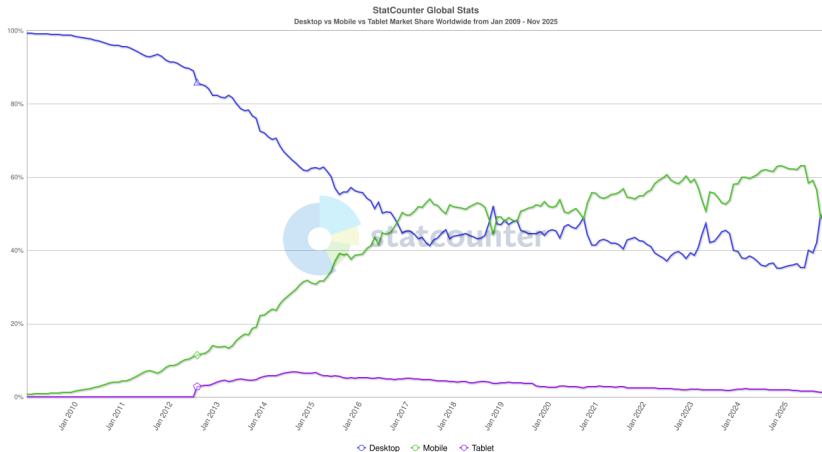
2025. november 19.

Felhasználási feltételek

This work is licensed under a [Creative Commons](#) “[Attribution 4.0 International](#)” license.



Webelérés asztali és mobil eszközökről



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

Mobilegeddon

- A szó a webmesterek és webfejlesztők rémületét fejezi ki, amelyet 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:

- Chrome Lighthouse:
 - Licenc: *Apache License 2.0*
 - Webhely: <https://developer.chrome.com/docs/lighthouse/>
 - Tároló: <https://github.com/GoogleChrome/lighthouse>
 - A Lighthouse jelentések a Chrome DevTools Lighthouse füléről érhetőek el.

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)

- Korábban megszokott gyakorlat a mobil webtartalom egy külön webhelyről történő szolgáltatása, amely egy (gyakran m nevű, mint például `m.example.com`) altartományon üzemel.
- Példák ma is így működő webhelyekre:
 - Facebook: <https://m.facebook.com/>
 - YouTube: <https://m.youtube.com/>

Mi az adaptív webdizájn?

- 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.
- 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.

Mi a reszponzív webdizájn? (1)

- A **reszponzív webdizájn** (RWD) egy webfejlesztési megközelítés olyan weblapok készítéséhez, amelyek 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, amelyen 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.

Weboldalak renderelését befolyásoló tényezők (1)

- A kimeneti eszköz jellemzői, amelyek hatással lehetnek arra, hogy hogyan jelenik meg egy weboldal:
 - a kimeneti eszköz típusa (például képernyő vagy nyomtató),
 - a nézetablak mérete (elsősorban a szélessége),
 - a pixel-/pontosűrűség,
 - a tájolás (álló vagy fekvő),
 - a rendelkezésre álló színek,
 - a támogatott médiatípusok (például képformátumok),
 - a felhasználó preferenciái (például hogy a sötét és világos témák közül melyiket preferálja).
- A legfontosabbak a nézetablak szélessége és tájolása.

Weboldalak renderelését befolyásoló tényezők (2)

Ezen tényezők figyelembevételével egy weboldal az alábbiakban igazodhat az eszközhöz, amelyen megjelenítésre kerül:

- elrendezés,
- a használt betűtípusok és tipográfiai jellemzőik (például betűméret és sortávolság),
- médiatartalmak,
- színek.

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.
 - Médialekérdezések révén különböző statikus elrendezéseket határoz meg különböző nézetablak szélességekhez.
- **Reszponzív:** folyékony elrendezések egy sorozatként tekinthető.
 - Médialekérdezések révén különböző folyékony elrendezéseket határoz meg különböző nézetablak szélességekhez.

Reszponzív webhelyek

Példák:

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

Pixelfogalom (1)

- Vonatkozó CSS specifikáció:
 - [CSS Values and Units Module Level 3](#) (W3C Candidate Recommendation Draft, 22 March 2024)
- Különbözőek a CSS-pixelek és a fizikai pixelek!

Pixelfogalom (2)

Eszközpixel (*device pixel*):

- A kimeneti eszköz területének az a legkisebb része, amely a rendelkezésre álló színek mindegyikét képes megjeleníteni.
 - A hagyományos színes kijelzőknél egy olyan négyzet vagy téglalap alakú terület, amely egy piros, zöld és kék alpixelt tartalmaz.
- A CSS nem eszközpixelet használ a hosszúságok kifejezéséhez, hanem CSS-pixeletet.

Pixelfogalom (3)

- Napjaink mobiltelefonjai nagyon nagy felbontású kijelzőkkel rendelkeznek.
 - Ha a CSS az eszközpixelesen alapulna, akkor az eszköapixel kis mérete miatt a legtöbb létező weboldal használhatatlan lenne mobil képernyőkön.
 - Például a 12 eszköapixel betűmérettel megjelenített szöveg hihetetlenül kicsi lenne.
- A probléma megoldásaként a CSS egy saját pixel absztrakciót használ: a **CSS-pixelt**, amely több eszköapixelből állhat.
 - Változhat az egy CSS-pixelt alkotó eszköxpixelek száma.
 - A nagyítás úgy kerül megvalósításra, hogy több és több eszköapixel kerül allokálásra egy CSS-pixelhez.

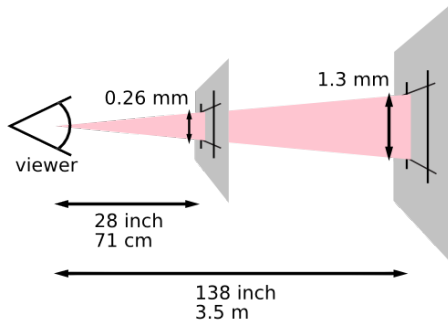
Pixelfogalom (4)

Referenciapixel:

- A referenciapixel az a látószög, amely 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>

Pixelfogalom (5)

Referenciapixel (folytatás):



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

Pixelfogalom (6)

- 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](#)
 - [YesViz.com](#)

Pixelfogalom (7)

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	$1\text{px} = 1\text{in}/96$

Pixelfogalom (8)

- Az abszolút hosszúság 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 referenciapixelhez való kötésével.
- A px mértékegységet **látószög mértékegységnek** (*visual angle unit*) is nevezik.

Pixelfogalom (9)

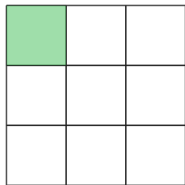
- Képernyő médiáknál, kis pixelsűrűségű eszközöknél és szokatlan távolságról nézett eszközöknél a px mértékegységhez ajánlott kötni a többi mértékegységet.
 - A px mértékegység azt az egész eszköz pixelszámot jelenti, amely a legjobban közelíti a referenciapixelt.
 - Az ilyen eszközökön a fizikai mértékegységek (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.

Pixelfogalom (10)

- 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](#)
- Asztali böngészőkben a DPR általában 1 a 100% nagyítási szinten.
 - Kivételt képeznek a nagyon nagy pixelsűrűségű kijelzők, mint például az Apple Retina kijelzői.
- Azonban a mobil böngészőkben a DPR már a 100% nagyítási szinten is egynél nagyobb.

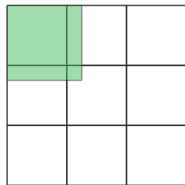
Pixelfogalom (11)

Zoom level: 100%



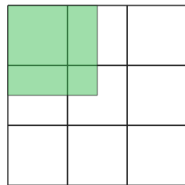
DPR: 1

Zoom level: 125%



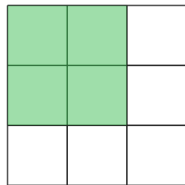
DPR: 1.25

Zoom level: 150%



DPR: 1.5

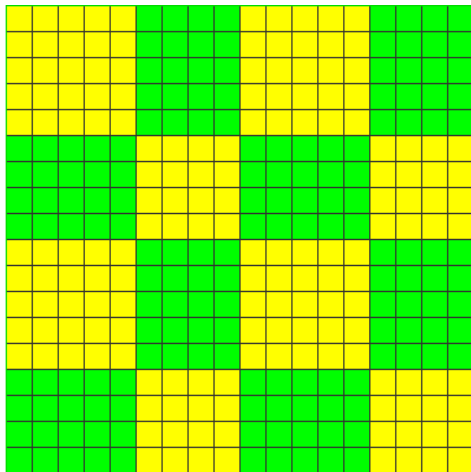
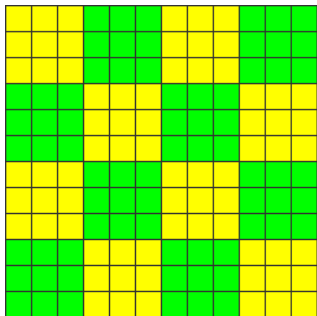
Zoom level: 200%



DPR: 2

3. ábra. Egy CSS-pixel mérete egy asztali böngészőben különféle nagyítási szinteken

Pixelfogalom (12)



4. ábra. Egy 12×12 eszközpixel méretű sakktábla minta 150%-os nagyítás alatt az asztali Google Chrome böngészőben

Pixelfogalom (13)

- Demonstráció: <https://pixel-example.surge.sh/>
- További ajánlott olvasnivaló:
 - Peter-Paul Koch. [A pixel is not a pixel is not a pixel](#). 20 April 2010.

A reszponzív webdizájn alkotóelemei

- A reszponzív webdizájn fő komponensei:
 - Relatív hosszúság mértékegységek
 - Médialekérdezések
 - Nézetablak
 - Reszponzív elrendezések (folyékony/rugalmas rácsok)
 - Reszponzív képek
 - Reszponzív tipográfia
 - Reszponzív CSS könyvtárak és keretrendszerek
- Lásd: [MDN Web Docs – The building blocks of responsive design](#)

A CSS relatív hosszúság mértékegységei

- `ch`: az elemhez tartozó betűkészlet 0 (nulla) karakterjének szélességével egyezik meg.
- `em`: a `font-size` tulajdonság az elemhez tartozó számított értékét jelenti.
- `ex`: az elemhez tartozó betűkészlet *x-height* értékét jelenti (a `font-size` tulajdonság esetén a szülő elemhez tartozó *x-height* értéket).
 - Azért nevezik így, mert gyakran a betűkészlet *x* karakterének magasságával egyezik meg értéke.
- `rem`: az `em` egység a gyökéremhez tartozó számított értékét jelenti.
- `vw`, `vh`, `vmin`, `vmax`: lásd később.

Nézetablak (*viewport*)

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

Króm

- A “króm” (*chrome*) kifejezést használjuk a böngésző felhasználói felületének a nézetablakon kívüli részére.
- Beleértjük a krómba például a címsort, a menüsort és a görgetősávokat.

CSS nézetablak mértékegységek

- Vonatkozó specifikáció:
 - [CSS Values and Units Module Level 3](#) (W3C Candidate Recommendation Draft, 22 March 2024)
- 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 médialekérdezések? (1)

- Egy **médialekérdezés** (*media query*) egy módszer a felhasználói ágens vagy eszköz bizonyos jellemzőinek vizsgálatára, amelyen a dokumentum megjelenítésre kerül.
- Stíluslapok feltételes alkalmazására szolgálnak.

Mik a médialekérdezések? (2)

- A médialekérdezések (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.

Médialekérdezés specifikáció

- Az aktuális szabvány:
 - [Media Queries Level 3](#) (W3C Recommendation, 21 May 2024)
- A szabvány következő verziója:
 - [Media Queries Level 4](#) (W3C Candidate Recommendation Draft, 25 December 2021)

Médialekérdezés támogatás

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

Médialekérdezések rendelkezésre állása

Médialekérdezések az alábbi nyelvekben használhatók:

- CSS
- XML
- HTML
- JavaScript

Médialekérdezések 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;
```

Médialekérdezések CSS-ben (2)

@import: (folytatás)

- Az opcionális médialekérdezés lista a feltételeket fejezi ki, amelyek teljesülése esetén alkalmazható az @import.
- Ha a médialekérdezés nem illeszkedik, akkor az importált stíuslap szabályai nem lesznek alkalmazva, pontosan úgy, mintha a stíuslap az adott médialekérdezést tartalmazó @media blokkba lenne csomagolva.

Médialekérdések CSS-ben (3)

@media:

- Vonatkozó specifikáció:
 - [CSS Conditional Rules Module Level 3](#) (W3C Candidate Recommendation Draft, 15 August 2024)
 - Lásd: [Media-specific style sheets: the @media rule](#)
- Példa:

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

Médialekérdések CSS-ben (4)

@media: (folytatás)

- A @media szabály egy olyan feltételes csoportszabály, amelynek feltétele egy médialekérdés.
- A **feltételes csoportszabályok** egy feltételt társítanak más CSS-szabályok egy csoportjához.
 - Minden egyes feltételes csoportszabálynak van egy feltétele, amely 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.

Médialekérdések 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` pseudo-attribútuma a médiát adja meg, amelyre 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"?>
```

Médialekérdezések HTML-ben

- Vonatkozó specifikáció: [HTML Living Standard](#)
- Lásd a `media` attribútumot, amely 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">
```

Médialekérdezések JavaScript-ben (1)

- Vonatkozó specifikáció: [CSSOM View Module](#) (W3C Working Draft, 16 September 2025)
- A `MediaQueryList` interfész lehetővé teszi egy médialekérdezés kiértékelését programkódból.
 - Eseményfigyelők regisztrálhatók, amelyek értesítve lesznek a médialekérdezés eredményének változásairól.
 - Lásd: [The MediaQueryList Interface](#)
- Támogatás: https://caniuse.com/mdn-api_mediaquerylist

Médialekérdezések JavaScript-ben (2)

Példa:

```
// Médialekérdezés 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  
}
```

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

Médialekérdezések JavaScript-ben (3)

Példa: (folytatás)

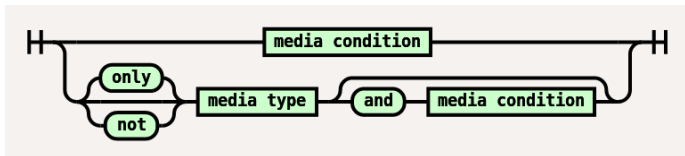
```
// Médialekérdezés lista létrehozása:  
var mql = window.matchMedia("(orientation: portrait)");  
  
// Callback függvény definiálása az eseményfigyelőhöz:  
function handleOrientationChange(event) {  
  if (event.matches) {  
    // A nézetablak jelenleg álló tájolású  
  } else {  
    // A nézetablak jelenleg fekvő tájolású  
  }  
}  
  
// A callback függvény regisztrálása eseményfigyelőként:  
mql.addListener(handleOrientationChange);
```

Mértékegységek médialekérdezésekben

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

Médialekérdezés szintaxis (1)

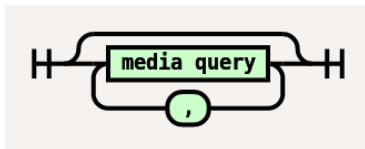
- Egy médialekérdezés egy opcionális módosítóból, egy opcionális médiatípusból és egy médiafeltételből áll:



5. ábra. Médialekérdezés szintaxis

Médialekérdezés szintaxis (2)

- Több médialekérdezés egy vesszővel elválasztott médialekérdezés listává kombinálható:



6. ábra. Médialekérdezés lista szintaxis

Médialekérdezés kiértékelés (1)

- Egy medialekérdezés egy logikai kifejezés, amely igaz vagy hamis. Egy medialekérdezés igaz, ha teljesülnek az alábbiak:
 - Ha meg van adva a médiatípus, akkor az illeszkedik annak az eszköznek a médiatípusára, ahol a felhasználói ágens fut.
 - A médiafeltétel igaz.
- Egy medialekérdezés lista igaz, ha valamely komponens medialekérdezése igaz, és csak akkor hamis, ha minden komponens medialekérdezése hamis (logikai vagy kapcsolat).
- Példa:

```
@media screen, print { /* ... */ }
```

Médialekérdezés kiértékelés (2)

- A felhasználói ágenseknek újra ki kell értékelniük a médialekérdezéseket válaszul a felhasználói környezetben történt olyan változásokra, amelyeknek tudatában vannak, és megfelelően kell változtatniuk valamennyi, a szóban forgó médialekérdezésektől függő konstrukció viselkedését.
- Ha például egy eszköz fekvőről álló tájolásúra dől.

Médialekérdezés módosítók (1)

- Egy médialekérdezés elején opcionálisan elhelyezhető egy médialekérdezés módosító, amely egy olyan kulcsszó, amely a következő médialekérdezés jelentését módosítja.

Médialekérdezés módosítók (2)

not módosító:

- Egy önálló médialekérdezés eredménye negálható az elejére a not kulcsszót írva.
- Ha a médialekérdezés 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">
```

Médialekérdezés módosítók (3)

only módosító:

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

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

Médiatípusok (1)

- Egy **médiatípus** (*media type*) felhasználói ágens eszközök egy széles csoportja, amelyeken megjeleníthető egy dokumentum.
- Nem összetévesztendő az IANA média típusokkal, mint például `text/html`!

Médiatípusok (2)

A következő médiatípusok definiáltak médialekérdezésekben való felhasználásra:

- `all`: az összes eszközre illeszkedik.
- `print`: a nyomtató eszközökre illeszkedik és az olyan eszközökre, amelyek célja egy nyomtatott megjelenítés visszaadása, mint például egy dokumentum nyomtatási képét mutató böngésző (*print preview*).
- `screen`: a `print`-re nem illeszkedő összes eszközre illeszkedik.

Médiatípusok (3)

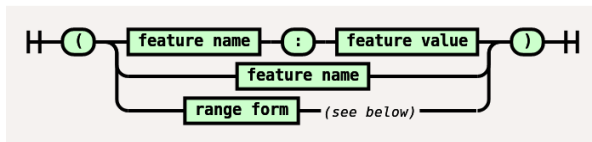
- Továbbá a következő elavult médiatípusok is definiáltak: tty, tv, projection, handheld, braille, embossed, aural, speech.
- A szerzők számára tilos ezen médiatípusok használata, helyettük megfelelő médiajellemzők választása javasolt, amelyek jobban fejezik ki annak az eszköznek a jellegét, amelyre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

Médiajellemzők (1)

- Egy **médiajellemző** (*media feature*) egy, a médiatípusoknál finomabb teszt, amely a felhasználó ágens vagy megjelenítő eszköz egy bizonyos jellemzőjét vizsgálja.
- Példák: width, height, orientation, resolution

Médiajellemzők (2)

- A médiajellemzők 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.



7. ábra. Médiajellemző szintaxis

Médiajellemzők (3)

Eltérések a tulajdonságok és a médiajellemzők között:

- A tulajdonságok célja az, hogy információkat adjanak arról, hogy a dokumentum hogyan jelenjen meg. A médiajellemzők az output eszköz követelményeinek leírására szolgálnak.
- A médialekérdezéseket 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 médiajellemző megadható csupán a nevével (a kettőspont és az érték elhagyásával) a jellemző logikai kontextusban történő kiértékeléséhez. Ez egy kényelmes rövid kifejezés mód olyan jellemzőkhöz, amelyek értékeként elfogadható `0` vagy `none`.
 - Például `(color)` akkor igaz, ha a `color` médiajellemző értéke nem nulla.

Médiajellemzők (4)

Eltérések a tulajdonságok és a médiajellemzők között: (folytatás)

- A tartomány típusú médiajellemzők tartomány kontextusban írhatók, amely 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 médiajellemzők egyetlen értéket fogadnak el, például egy kulcsszót vagy számot.

Médiajellemzők (5)

- Ha egy médiajellemző egy olyan fogalomra hivatkozik, amely nem létezik azon az eszközön, amelyen a felhasználói ágens fut, akkor a médialekérdezés értéke mindig hamis kell, hogy legyen.
- Például hangos felhasználói ágenseknél nem értelmezett a szélesség (`width`).

Médiajellemző típusok

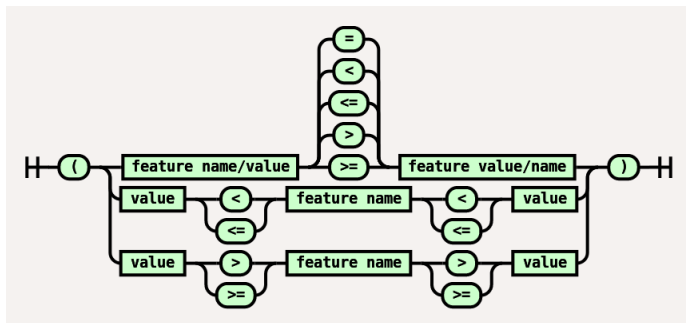
- Minden médiajellemző meghatározza a típusát, amely tartomány vagy diszkrét.
 - A diszkrét médiajellemzők egy halmazból veszik fel értéküket.
 - Az értékek lehetnek kulcsszavak vagy logikai számok (0 és 1), de az bennük a közös, hogy nem értelmezett rajtuk rendezés.
 - A tartomány médiajellemzők egy 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 médiajellemzők kiértékelhetők logikai kontextusban és a nevük előtt megadható \min - vagy \max - előtag.

Médiajellemzők kiértékelése logikai kontextusban

- A médiajellemzők írhatók egyszerűen csupán a nevükkel, mint például (color).
- Az ebben az alakban való megadásnál a médiajellemző 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 médiajellemző értéke igaz.
 - Egyébként hamis az értéke.

Médiajellemzők kiértékelése tartomány kontextusban (1)

- A tartomány típusú médiajellemzők tartomány kontextusban is írhatók, amely azt használja ki, hogy az értékeik a szokásos matematikai összehasonlító operátorok segítségével rendezettek:



8. ábra. Médiajellemző szintaxis tartomány kontextusban

Médiajellemzők 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: $(\text{height} > 600\text{px})$, $(600\text{px} < \text{height})$
- A többi alak, ahol a jellemző neve két érték összehasonlítás közé van ágyazva, igazat ad vissza, ha mindkét összehasonlítás igaz.
 - Példa: $(400\text{px} < \text{width} < 1000\text{px})$

Médiajellemzők kiértékelése tartomány kontextusban (3)

- Támogatás:
 - A Firefox 2018 óta támogatja.
 - A Chromium-alapú böngészők 2022 óta támogatják.
 - A Safari 2023 márciusa óta támogatja.
- Lásd: <https://caniuse.com/css-media-range-syntax>

A min- és max- előtagok használata tartomány típusú jellemzőkön

- Egy tartomány típusú médiajellemező a tartomány kontextusban történő kiértékelés helyett írható közönséges médiajellemezőként, 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 médiajellemezőkhöz nem adható meg min- vagy max- előtag.

Médiajellemzők kombinálása

- Több médiajellemző egy **médiafeltétellé** (*media condition*) kombinálható össze a not/and/or logikai műveletekkel.
 - Bármely médiajellemző negálható elé not helyezésével, mint például not (color).
 - Két vagy több médiajellemző összefűzhető közéjük and/or helyezésével.
 - Példa: (width < 600px) and (height < 600px), update: slow) or (hover: none)
 - Médiafeltételek csoportosíthatók zárójelek közé zárással, mint például not ((color) or (hover)).
- Nem megengedett egy médialekérdezés ugyanazon “szintjén” keverni az and/or/not logikai műveleteket.
 - Például (color) and (pointer) or (hover) érvénytelen.

Hibakezelés médialekérdezésekben

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

Rendelkezésre álló médiajellemzők (1)

Nézetablak/oldal dimenzió médiajellemzők:

- `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 médiajellemzők (2)

Megjelenítési minőségi médiajellemzők:

- `grid: 0 | 1`
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/grid>
- `overflow-block: none | scroll | 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 médiajellemzők (3)

Szín médiajellemzők:

- `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 médiajellemzők (4)

Interakciós médiajellemzők:

- `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-hover>
- `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>

Példák médialekérdezésekre

```
@media (orientation: landscape) { /* ... */ }
```

```
@media (orientation: portrait) { /* ... */ }
```

```
@media print and not (color) { /* ... */ }
```

```
@media screen and (prefers-color-scheme: dark) { /* ... */ }
```

```
@media screen and (max-width: 576px) { /* ... */ }
```

```
@media screen and (min-width: 576px) and (max-width: 768px) {  
    /* ... */  
}
```

```
@media screen and (min-width: 768px) { /* ... */ }
```

Médialekérdezések sorrendje (1)

Példa: *mobile first design*

```
/* Small devices (phones, less than 768px): */  
main {  
  width: 100%;  
}  
  
/* Medium devices (tablets, 768px and up): */  
@media (min-width: 768px) {  
  main {  
    width: 720px;  
  }  
}  
  
/* Large devices (desktops, 992px and up): */  
@media (min-width: 992px) {  
  main {  
    width: 960px;  
  }  
}
```

Médialekérdezések sorrendje (2)

Példa: *desktop first design*

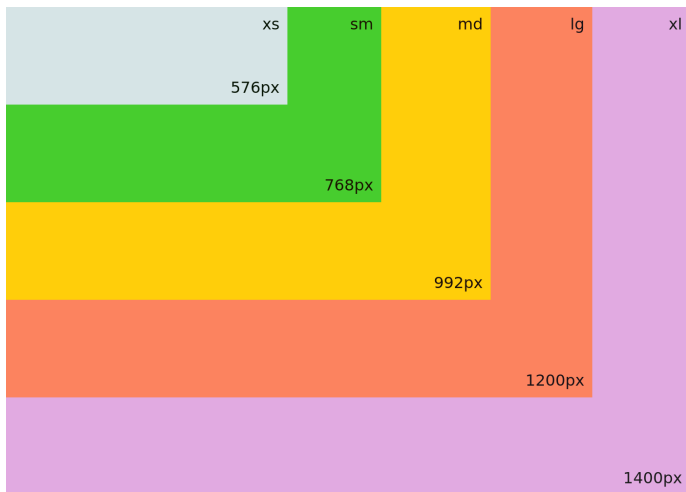
```
/* Large devices (desktops, 992px and up): */  
main {  
    width: 960px;  
}  
  
/* Medium devices (tablets, less than 992px): */  
@media (max-width: 992px) {  
    main {  
        width: 720px  
    }  
}  
  
/* Small devices (phones, less than 768px): */  
@media (max-width: 768px) {  
    main {  
        width: 100%;  
    }  
}
```

Tipikus médialekérdezés töréspontok (1)

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) { /* ... */ }
```

Tipikus médialekérdezés töréspontok (2)



9. ábra. A Bootstrap által használt reszponzív töréspontok

Media Queries Level 5

- Specifikáció: [Media Queries Level 5](#) (W3C Working Draft, 18 December 2021)
- Példa új médiajellemzőre:
 - `prefers-color-scheme: light | dark`
<https://developer.mozilla.org/en-US/docs/Web/CSS/outline>
 - Támogatás: <https://caniuse.com/prefers-color-scheme>
 - Firefox: lásd a `ui.systemUsesDarkTheme` opciót (`about:config`).
 - Lásd még: Thomas Steiner. [prefers-color-scheme: Hello darkness, my old friend](#).

Számítások médiajellemzők értékeiben (1)

- A médialekérdezések specifikáció az alábbi mondja:
*Properties sometimes accept complex values, e.g., calculations that involve several other values. **Media features only accept single values**: one keyword, one number, etc.*
- Azonban úgy tűnik, hogy minden főbb böngésző megengedi a `calc()` függvény használatát médiajellemzők értékeiben.
 - Lásd: https://caniuse.com/mdn-css_at_rules_media_calc

Számítások médiajellemzők értékeiben (2)

Például az alábbi működik minden főbb böngészőben:

```
@media screen and (max-width: calc(10in + 5rem)) {  
  .cards {  
    flex-direction: column;  
  }  
}
```

Médialekérdezések – további olvasnivaló

- [MDN Web Docs – Media queries](#)
- [MDN Web Docs – Using media queries](#)
- [MDN Web Docs – Using Media Queries for Accessibility](#)

Nézetablak (1)

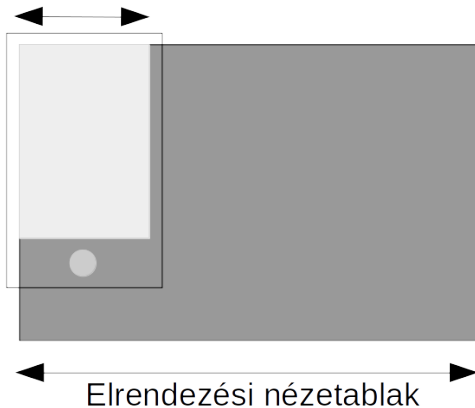
- A mobil/kézi eszközök böngészőinek nézetablaka általában sokkal keskenyebb egy, a CSS által ajánlott CSS-pixelméretet adó nagyítási szinten, mint egy asztali böngészőablaké.
- A keskeny nézetablak gondot jelent olyan dokumentumoknál, amelyeket úgy terveztek, hogy az asztali böngészőkben nézzenek ki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



10. ábra. A két nézetablak

Nézetablak (4)

- Környezetfüggő lehet önmagában a “nézetablak” kifejezés jelentése:
 - A nézetablak mértékegységeknél a látható nézetablakot jelenti.
 - A médialekérdezéseknél azonban az elrendezési nézetablakot jelenti.

Nézetablak (5)

Nézetablak méretek tesztelése médialekérdezésekben:

- A mobil/kézi eszközök böngészőiben a `height` és `width` médiajellemzők az elrendezési nézetablak magasságát és szélességét tesztelik.
 - Lásd: <https://www.quirksmode.org/css/tests/mediaqueries/width.html>

Nézetablak (6)

Böngésző nézetablak méretek lekérdezése JavaScript-ben:

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

Nézetablak (7)

Megszokott nézetablak méretek mobil és tablet eszközökhöz:

- [Screen Sizes – Viewport Sizes and Pixel Densities for Popular Devices](#)
- [YesViz.com – Viewport Size for Devices](#)

Nézetablak (8)

- 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">
```

- Szabványosítási erőfeszítések: [CSS Viewport Module Level 1](#) (W3C First Public Working Draft, 25 January 2024)

Nézetablak (9)

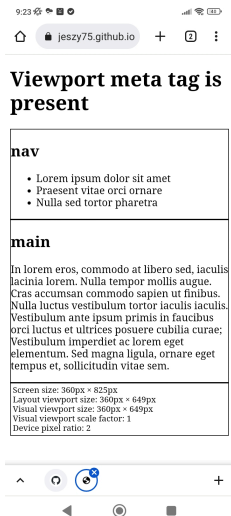
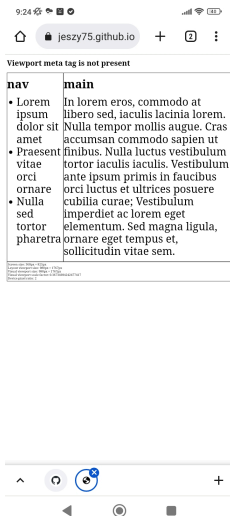
- A viewport meta elem jelenlétét/hiányát szemléltető példák:
<https://jeszy75.github.io/viewport-example/>
<https://github.com/jeszy75/viewport-example/>

Nézetablak (10)



11. ábra. A viewport meta címke hiányának/jelenlétének hatása

Nézetablak (11)



12. ábra. A viewport meta címke hiányának/jelenlétének hatása

Nézetablak (12)

További olvasnivaló:

- [MDN Web Docs – Viewport concepts](#)
- [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. [A new Device Adaptation spec.](#)
- 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 rácsos 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 túlcsoorduljanak 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>

Flexbox elrendezés (5)

Példa reszponzív flexbox elrendezés médialekérdezésekkel történő megvalósítására (francia kártyalapok):

```
.cards {  
  display: flex;  
  flex-direction: column;  
  gap: 1rem;  
  margin: 1rem;  
}  
  
@media screen and (min-width: calc(10in + 5rem)) {  
  .cards {  
    Flexbox elrendezés (5)  
    flex-direction: row;  
  }  
}
```

Flexbox elrendezés (6)

- Érdekesség: a flexbox elrendezés lehetővé teszi a reszponzivitás médialekérdezések használata nélküli megvalósítását.
- Ugyanaz az elrendezés médialekérdezések használata nélkül (francia kártyalapok):

```
.cards {  
  display: flex;  
  flex-wrap: wrap;  
  gap: 1rem;  
  margin: 1rem;  
}
```

- A fenti stílus szabályokkal létrehozott elrendezés teljesen reszponzív a `flex-wrap: wrap` deklarációnak köszönhetően.

Rácsos elrendezés (1)

Vonatkozó specifikáció:

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

Rácsos elrendezés (2)

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

Rácsos elrendezés (3)

- A *flexbox*-szal ellentétben, amely egytengelyű, a rácsos elrendezés olyan kétdimenziós elrendezésekhez optimalizált, amelyeknél a tartalom igazítása mindkét irányban szükséges.

Rácsos elrendezés (4)

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

Rácsos 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/>

Rácsos elrendezés (6)

Példa reszponzív rács médialekérdezésekkel történő megvalósítására (francia kártyalapok):

```
.cards {
  display: grid;
  gap: 1rem;
  margin: 1rem;
}
@media screen and (min-width: calc(5in + 3rem)) {
  .cards {
    grid-template-columns: repeat(2, 2.5in);
  }
}
@media screen and (min-width: calc(7.5in + 4rem)) {
  .cards {
    grid-template-columns: repeat(3, 2.5in);
  }
}
@media screen and (min-width: calc(10in + 5rem)) {
  .cards {
    grid-template-columns: repeat(4, 2.5in);
  }
}
```

Rácsos elrendezés (7)

- Érdekesség: a rácsos elrendezés lehetővé teszi a reszponzivizás médialekérdezések használata nélküli megvalósítását.
 - Lásd például: Juan Diego Rodríguez. [Beyond CSS Media Queries](#). May 16, 2024.
- Ugyanaz az elrendezés médialekérdezések használata nélkül (francia kártyalapok):

```
.cards {  
  display: grid;  
  gap: 1rem;  
  grid-template-columns: repeat(auto-fit, 2.5in);  
  margin: 1rem;  
}
```

- Ez minden főbb böngészőben működik, lásd:
https://caniuse.com/mdn-css_properties_grid-template-rows_repeat

Többhasábos elrendezés (1)

Vonatkozó specifikáció:

- [CSS Multi-column Layout Module Level 1](#) (W3C Candidate Recommendation Snapshot, 16 May 2024)

Többhasábos elrendezés (2)

- Egy olyan elrendezés, amely lehetővé teszi a szerzők számára annak deklarációját, hogy egy elem tartalma több hasábra kerüljön tördelésre, közöttük hézaggal és elválasztó vonalakkal.

Többhasábos elrendezés (3)

- Támogatás: <https://caniuse.com/multicolumn>
- További olvasnivaló:
 - MDN Web Docs – CSS multi-column layout

Többhasábos elrendezés (4)

- Mintapéldák:

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

Rugalmas helyettesített elemek (1)

- **Helyettesített elem** (*replaced element*): egy olyan elem, amelynek 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](#)
 - [MDN Web Docs – Replaced elements](#)

Rugalmas helyettesített elemek (2)

- A `width` tulajdonság 100%-ra történő beállításával nyújtunk egy helyettesített elemet (például egy képet), hogy töltsse ki a tartalmazó elemet.

Rugalmas helyettesített elemek (3)

- 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%;  
}
```

- Ha az `` elemhez meg vannak adva a `height` és `width` attribútumok, akkor a `height` tulajdonság értékét `auto`-ra kell állítani, hogy megelőzzük a kép torzulását.

- Példa:

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

Eszközök

- Böngészőeszközök:
 - Firefox: [Responsive Design Mode](#)
 - Chromium, Google Chrome: [Device Mode \(Chrome DevTools\)](#)
 - Chromium-based Edge: [Device Emulation \(Microsoft Edge DevTools\)](#)
- *Online* eszközök:
 - [Screen Size Map](#)

Reszponzív keretrendszerek (1)

- 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>

Reszponzív keretrendszerek (2)

Piaci részesedés és népszerűség:

- [Usage statistics of CSS frameworks for websites \(W3Techs\)](#)
- [State of CSS 2025: Other Tools](#)
- [CSS Framework Ranking \(OSS Insight\)](#)

Bootstrap (1)

- Dokumentáció: [Get started with Bootstrap \(Official Docs\)](#)
- Helyi telepítés az npm-mel (például testreszabáshoz szükséges):

```
npm install bootstrap@5.3.8
```

Bootstrap (2)

- Az összes főbb mobil és asztali böngésző és platform legújabb stabil kiadását támogatja.
 - A támogatott böngészők listája: `.browserslistrc`
- *Mobile-first* megközelítést követ, ezért a *viewport* `<meta>` címkét hozzá kell adni a `<head>` elemhez.
- Sass-ben írták és a Dart Sass szükséges az `.scss` forrásállományok `.css` állományokká fordításához.
- A `box-sizing` CSS-tulajdonság értékét `border-box`-ra állítja minden elemhez.
- A reszponzív rácsrendszere a flexbox segítségével került megvalósításra a CSS rácsos elrendezés helyett.

Bootstrap (3)

Eszköztámogatás:

- Visual Studio Code:
 - Bootstrap 5 Quick Snippets <https://marketplace.visualstudio.com/items?itemName=AnbuselvanRocky.bootstrap5-vscode>
<https://github.com/anburocky3/bootstrap5-snippets>
 - Bootstrap IntelliSense <https://marketplace.visualstudio.com/items?itemName=hossaini.bootstrap-intellisense>
<https://github.com/aviiceena/bootstrap-intellisense>

Bootstrap (4)

“Hello, World!” példa a dokumentációból:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello, Bootstrap!</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css"
          rel="stylesheet"
          integrity="sha384-sRI14kxILFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkuIAvNWLzeN8tE5YBuJzqJLB"
          crossorigin="anonymous">
  </head>
  <body>
    <h1>Hello, Bootstrap!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js"
           integrity="sha384-G/EV+4j2dNv+tePo3++6LCgdCROaejBqfUeNjuKAiuXbjrxilcCdDz6ZAVfHWe1Y"
           crossorigin="anonymous"></script>
  </body>
</html>
```

Bootstrap (5)

Példák:

- <https://bootstrap-container-example.surge.sh/>
- <https://bootstrap-grid-examples.surge.sh/>