# Reszponzív weboldal képekkel

Egy weboldalnak minden eszközön jól kell kinéznie! Ez nem csak azt jelenti, hogy esztétikus legyen,. hanem azt is, hogy kényelmesen használható, jól olvasható legyen, jól megtalálhatóak legyenek az információk mind a legkisebb telefonon, mind a széles asztali képernyőkön.

reszponzív = a képernyő méretét figyelembe véve változik a weboldal megjelenése

## Reszponzív képek

A képeket úgy állítjuk be, hogy minden böngésző mérethez igazodjanak. Azaz a szélességét nem pixelben, hanem százalákban adjuk meg.

<img src="img\_girl.jpg" **style="width:100%; height:auto;"**>

Amennyiben a kép mérete sokkal kisebb, mint a böngésző mérete, elképzelhető, hogy a kép az eredeti méretéhez képest nagyítva jelenik meg, és ezért pixeles lesz. (túl kicsi a felbontása a megjelenítő eszközhöz képest)

Ekkor használhatjuk a max-width tulajdonságot. Ebben az esetben a böngészőablak méretezésével növekszik ugyan a kép mérete is, de sosem lesz nagyobb, mint az eredeti képméret.

<img src="img\_girl.jpg" style="**max-width:100%;**height:auto;">

Különböző eszközökön különböző méretű képeket használjunk! Ezzel befolyásolhatjuk az oldal betöltésének idejét!

## A **pictures** tag

<picture>  
  <source srcset="img\_smallflower.jpg" media="(max-width: 600px)">  
  <source srcset="img\_flowers.jpg" media="(max-width: 1500px)">  
  <source srcset="flowers.jpg">  
  <img src="img\_smallflower.jpg" alt="Flowers">  
</picture>

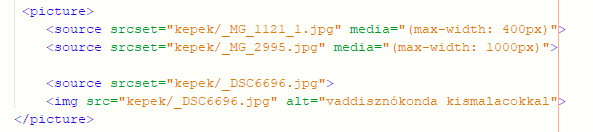
A picture csomagoló elem között megadott source tag-ek segítségével a megjelenítő média méretétől függően tölthetünk be képeket.

Az img tag kötelezően a **picture tag utolsó eleme**.

Az img tag src atribútumában megadott elérési úton lévő kép akkor fog betöltődni, ha a böngésző nem ismeri a picture elemet. A kép címkénél kell megadnunk az alt és egyéb a kép megjelenítése során használni kívánt tulajdonságokat (nem a picture elemnél), mint az id, class, title!

A source tag srcset kötelező attribútumában megadott kép akkor töltődik be, ha a megjelenítő eszköz mérete a media attributumban megadott határnak megfelelő. Ezzel elérhetjük, hogy különböző eszközökön eltérő méretű képeket töltsünk le, amivel – ha jól csináljuk – kevésbé terheljük meg mobil eszközünk a sávszélességét és tárhelyét.

PRÓBÁLD KI! Helyezz el picture elemet az index.html-ben. 3 képet adj meg forrásként:



Majd méretezd a böngésződet! Látni fogod, hogy a böngésző különböző méreténél különböző kép töltődik be. Fontod, hogy a max-widh attribútumot használod, akkor a méretek értékeit növekvő sorrendbe kell megadnod.

A jelentése: Amennyiben a képernyő mérete kisebb, mint 400px, az első kép fog betöltődni.

400px- és 1000px között a második,

1000px felett pedig a harmadik kép.

A valóságban persze nem három különböző képet adunk meg, hanem a 3 képnek 3 különböző méretét/felbontását!

## Reszponzív képgaléria készítése

Készítsd el az alábbi mintán látható weboldalt!

## Állományok és mappák létrehozása

Ehhez hozz létre egy mappát: SAJAT:NEV\_KEPGALERIA néven.

Hozz létre egy index.html állományt és a KEPEK mappát. A képek mappába másodl bele a reszponzív\_feladat\_ZIP-ben található képeket!

## Készítsük el az oldalunkat: index.html

1. Alakítsd ki az index.html állományt a szokásos módon. helyezd el a fejrészben a meta és a title tag-eket. A title tagben a saját neved legyen.
2. A body elemben legyen egy div, aminek adj egy azonosítót. <div id=”tarolo”></div>
3. Hozzunk létre ezen belül még két div tárolót. Az egyikben lesz a nagykép, a másikban pedig az indexképek. Most így néz ki a html body része:



Ezután elhelyezhetnénk a képeket a galeria id-vel ellátott div tárolóba.

1. Azonban nézzük meg, hogy a képeink mérete 1 Mbyte fölött van és a felbontásuk pedig 1600x1000 pixel. Ugyanakkor, ha az oldalon egymás mellett 4 képet helyezünk el, akkor a legszélesebb képernyő sestén sem valószínű, hogy 300px -nél nagyobb szélességet vennének fel. Ezért nem lenne célszerű, ha ekkora képeket töltenénk be , mivel sok kép esetén ez jelentősen lassítaná az oldal betöltési sebességét.

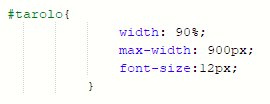
**Ezért a képeket át kell méretezni. Itt megnézheted hogyan:** [kepekkotegeltatalakitasa.docx](https://etananyag.szamalk-szalezi.hu/pluginfile.php/23084/mod_assign/introattachment/0/kepekkotegeltatalakitasa.docx?forcedownload=1)

1. Ha átméretezted a képeket, akkor hozz létre egy KEPEK\_KICSI mappát és ide másold be az átméretezett képeket.
2. Helyezd el őket az oldalon. Most így néz ki a galeria id-vel ellátott tárolónk:



## Formázzuk meg az oldalt: stilus.html

1. A szegélyeket és a betűméreteket, típusokat, háttérszíneket paddingokat és margókat állítsd be saját elképzelésed szerint. Azért hasonlítson a mintára!
2. A fő tároló szélessége legyen az oldal 90%-a, de sose legyen nagyobb, mint 900px;

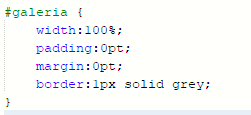


1. A nagykép formázása:



1. A galéria formázása:

A tároló szélességét beállítjuk 100%-ra. beállítjuk a külső és a belső margókat, és a szegélyt.



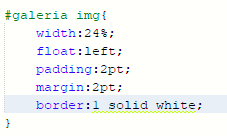
1. A galériában lévő képek formázása:

4 képet szeretnénk egymás mellett, ezért a szélességüket beállítjuk 25%-ra. Ezzel az őket tároló tároló szélességének negyede lesz az egyes képek mérete.

Ezután balra úsztatjuk a képeket – most kerülnek egymás mellé.

Belső és külső margót állítunk neki, de ebben apillanatban már csak 3 kép fér el egymás mellett, ezért a szélességüket 24%-ra állítjuk.

Ellátjuk egy szegéllyel.

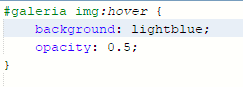


BONUS

Ha lekerekített szegélyt szeretnénk, akkor használhatjuk még pluszban a   
border-radius:5px utasítást.

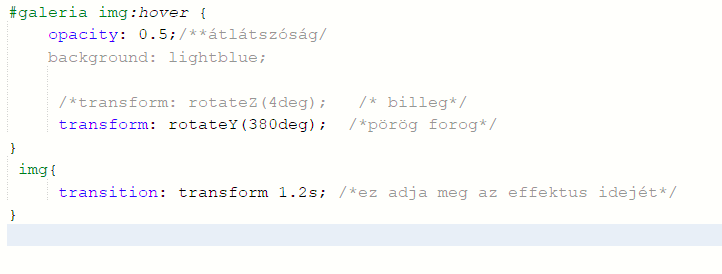
SŐT: tehetünk árnyékot is a képre a boksz-shadow utasítással : <https://www.cssmatic.com/box-shadow>

A képeket tovább módosíthatjuk, ha beállítjuk, hogy mi történjen , ha föléjük visszük az egeret. A linkek díszítéséhez hasonlóan itt is használhatjuk a :hover szelektort.



EXTRA BONUS

Próbáld ki az alábbi 2D-s effektusokat:



### Ugye milyen szuperül néz ki az oldal?

1. A következő lépés, hogy mobilon a képek ne egymás mellett, hanem egymás alatt helyezkedjenek el!

## Mobil nézet kialakítása Töréspontok: Media Queries

Media Queries-k használatával a stíluslapon töréspontokat helyezhetünk el. Ezeket a pontokat pixelben adjuk meg. Az egyes töréspontokhoz tartozó kapcsos zárójelek között adhatjuk meg azokat a css formázásokat, melyeket meg akarunk változtatni az eredeti css-ben meghatározott formázásokhoz képest. Amely formázási utasításokat itt nem adunk meg, azok az eredetileg megadott utasítások szerint jelennek meg.

1. A css fájlunk utolsó sorában adjuk meg az elábbi utasítást:

@media screen and (max-width:800px){

}

A kapcsos zárójelek között beírt utasítások csak akkor lesznek érvényben, ha az oldalunk mérete 800px alá csökken.

Az ide írt utasítások felülírhatják a korábban megadott utasításokat. amit nem definiálunk újra, az változatlanul érvényben marad.

PRÓBÁLD KI:

@media screen and (max-width:800px){

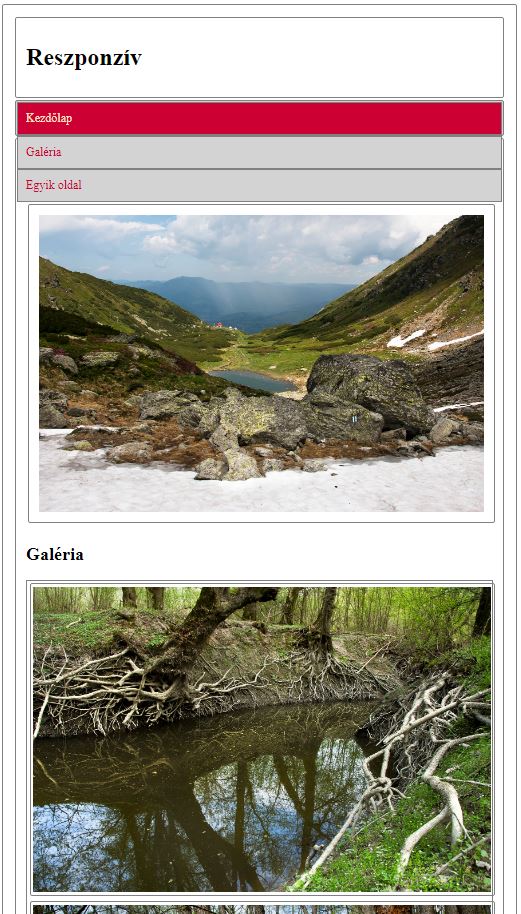
#tarolo {

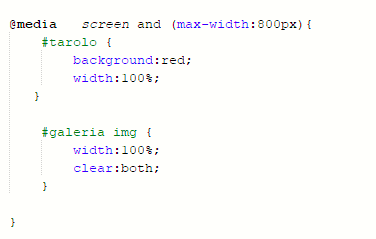
background: red;

}

}

Ennek hatására az oldalad háttérszíne jól láthatóan megváltozik, ha a böngészőablakot 800px alá csökkented.

1. Ezután még állítsuk be, hogy a mobil nézetben a képek ne egymás mellett, hanem egymás alatt helyezkedjenek el! Ehhez csak a szélességüket kell megváltoztatni, és az úsztatást megszüntetni.



Próbálkozzatok, állítsatok be minél többféle értéket és nézzétek meg az eredményt!