

# FRONTEND MĀJASLAPAS IZSTRĀDE



LATVIJAS UNIVERSITĀTE  
**BIZNESA, VADĪBAS  
UN EKONOMIKAS  
FAKULTĀTE**

**VUMC** VADĪBAS UN  
UZŅĒMĒJDARBĪBAS  
MĀCĪBU CENTRS

NACIONĀLAIS  
ATTĪSTĪBAS  
PLĀNS 2020



**EIROPAS SAVIENĪBA**  
Eiropas Sociālais  
fonds

IEGULDĪJUMS TAVĀ NĀKOTNĒ

ESF projekts Nr. 8.4.1.0/16/l/001 "Nodarbināto personu profesionālās kompetences pilnveide"



## GIT plūsma (*flow*)

# GIT plūsma (*flow*)

1. Pārslēgties uz main branch:  
**git checkout main**
2. Atjaunināt local main branch ar izmaiņām no GitHub repository:  
**git pull**
3. Izveidot jaunu local branch. Izpildot šo komandu, git automātiski pārslēgs aktīvo branch uz jaunizveidoto:  
**git checkout -b *branch-name***
4. Veikt imziņas failos jaunajā branch.
5. Pievienot visas saglabātās izmaiņas augšupielādes sarakstam (stage changes):  
**git add \***
6. Saglabāt aktīvajā branch veiktās izmaiņas (commit):  
**git commit -m "*commit message*"**
7. Augšupielādēt saglabātās izmaiņas (*commit*), kas pievienotas augšupielādes sarakstam (*staged changes*):  
**git push**

In case of fire



1. git commit



2. git push



3. leave building



## Adaptīvais dizains

# Mājaslapa dažādiem ekrāniem

Attēlojot mājaslapu mobilajiem ekrāniem, pārlūkprogrammas izveido logu ar noklusēto platumu 980px un mājaslapas saturu attālina, lai ietiplinātu šajā izmērā. Šī ir pārlūkprogrammu nianse, kas saglabājusies no laika, kad mājaslapas nebija pielāgotas mobilajiem ekrāniem.

Ja mūsu mājaslapas dizains ir ar CSS pielāgots mazākiem ekrāniem, izmantojam HTML meta elementu. Tas noteiks, ka mājaslapas platumam jāatbilst tieši ekrāna platumam, arī uz mobilajām ierīcēm.

`<meta name="viewport" content="width=device-width"/>`

Bez `<meta/>`



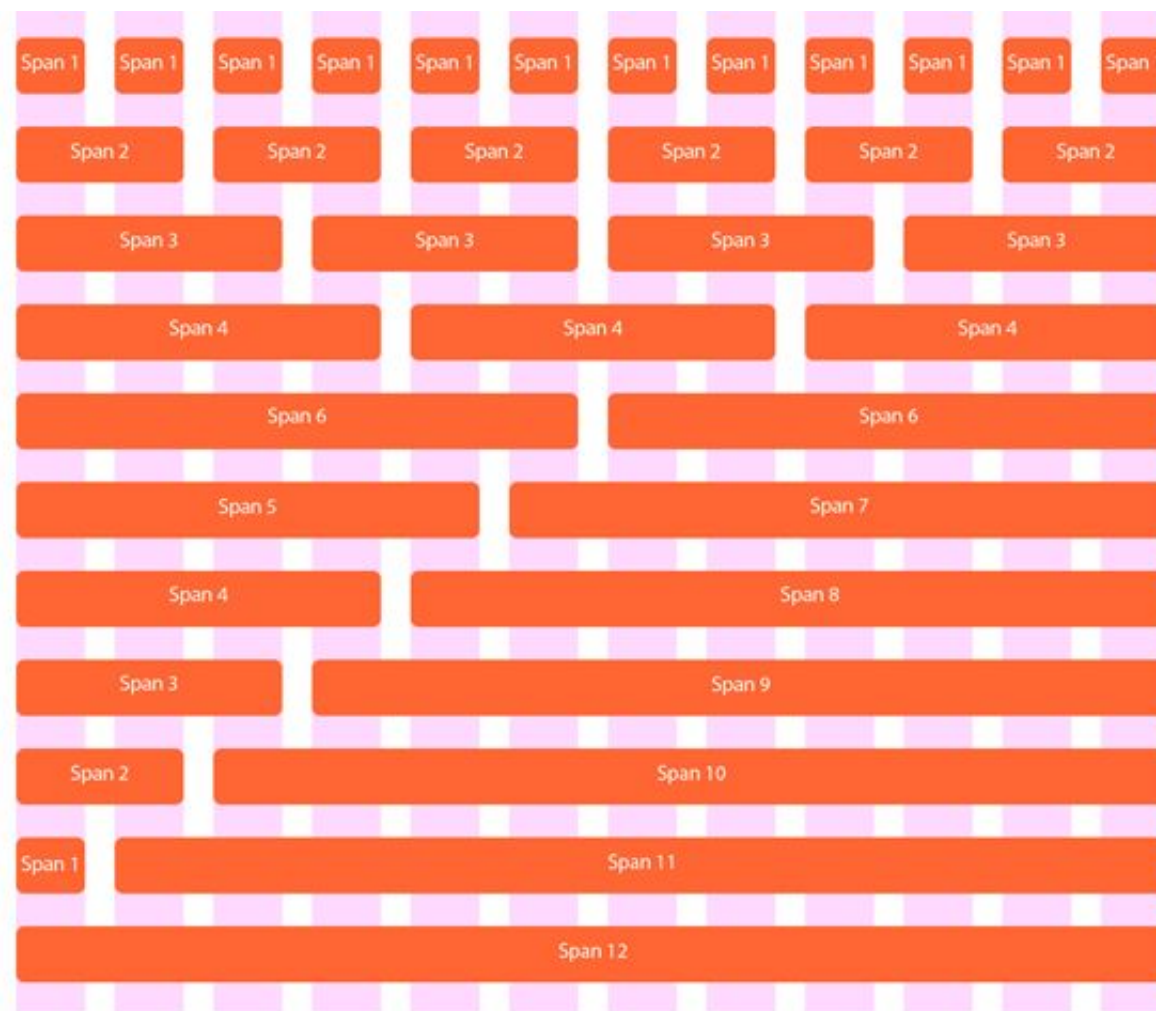
Ar `<meta/>`



# Kolonu un rindu izkārtojums

Veidojot mājaslapas izkārtojumu ir praktiski par to domāt kolonnās un rindās. Tādējādi to ir vieglāk pielāgot dažādiem ekrāna izmēriem, mainot kolonnu platumu attiecīgi tam cik to ietilpst ekrāna izmērā.

```
.col-1 {width: 8.33%}
.col-2 {width: 16.66%}
.col-3 {width: 25%;}
.col-4 {width: 33.33%}
.col-5 {width: 41.66%}
.col-6 {width: 50%}
.col-7 {width: 58.33%}
.col-8 {width: 66.66%}
.col-9 {width: 75%}
.col-10 {width: 83.33%}
.col-11 {width: 91.66%}
.col-12 {width: 100%}
```



# Izmēri dažādiem ekrāniem

Atšķirīgiem ekrānu izmēriem elementu izmēru nepieciešams pielāgot. Ja uz datora mājaslapas izkārtojums 3 kolonās ir pārskatāms, uz mobilās ierīces ekrāna piemērotāks būs izkārtojums, piem., 1 kolonā.

```
.nav,  
.content,  
.widget {  
  width: 100%;  
}
```



```
.nav,  
.widget {  
  width: 25%;  
}  
  
.content {  
  width: 50%;  
}
```



# Ekrānam adaptīvs CSS

Lai piešķirtu atšķirīgas vērtības CSS īpašībām, varam izmantot **@media** tehniku. Dotajā piemērā elementi ekrānā aizņems 100% platumu līdz izpildīsies nosacījums - ekrāna izmērs ir vismaz 1200px.

```
.nav,  
.content,  
.widget {  
    width: 100%;  
}
```

```
@media only screen and (min-width: 1200px) {  
    .nav,  
    .widget {  
        width: 25%;  
    }  
  
    .content {  
        width: 50%;  
    }  
}
```





# Lūzumpunkts (breakpoint)

Ar lūzuma punktu saprotam ekrāna izmēra un izšķirtspējas parametrus kuriem vēlamies izmantot atšķirīgu CSS noformējumu.

Lūzumspunktus nosakām ar jau pieminēto **@media** nosacījumu. Katram projektam var būt atšķirīgi lūzuma punkti. Bieži vien pietiek ar iedalījumu maziem(mobile), vidējiem(tablet) un lieliem(large ekrāniem).

Šo tabulu var izmantot kā ieteicamu, bet ne obligātu lūzumpunktu piemēr dažādiem ekrānu platumiem.

**@media only screen and (max-width: 576px) {...}**  
**@media only screen and (min-width: 576px) {...}**  
**@media only screen and (min-width: 768px) {...}**  
**@media only screen and (min-width: 992px) {...}**  
**@media only screen and (min-width: 1200px) {...}**  
**@media only screen and (min-width: 1400px) {...}**

X-Small	< 576px
Small	≥ 576px
Medium	≥ 768px
Large	≥ 992px
Extra large	≥ 1200px
Extra extra large	≥ 1400px



## Bootstrap ietvars

# Bootstrap pielietojums

Bootstrap ietvars sākotnēji tika izmantota Twitter izstrādei. Tomēr 2011. gadā tā kļuva pieejama kā bezmaksas rīks visiem mājaslapu izstrādātājiem.

Tā ir kolekcija ar HTML, CSS un JavaScript rīkiem, kas atvieglo mājaslapu izstrādi - jo sevišķi pateicoties ekrānam adaptīva CSS risinājumiem un plašam pārlūkprogrammu atbalstam.

Izmantojot Bootstrap, frontend izstrādātāji var vairāk laika veltīt koda izstrādei un mazāk CSS noformējuma veidošanai.

<https://getbootstrap.com>

Šī mājaslapa funkcionē arī kā oficiālā Bootstrap dokumentācija ar izsmelšiem piemēriem.



**Bootstrap**

## Citi CSS ietvari un bibliotēkas

Līdzīgus risinājumus piedāvā arī citi CSS noformējuma veidoti ietvari un bibliotēkas.



# Bootstrap uzstādīšana

Lai izmantotu Bootstrap mūsu mājaslapā, nepieciešams to norādīt kā āreju interneta resursu HTML faila `<head></head>` sadaļā.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOMLASjC" crossorigin="anonymous"/>
```

Šīs CSS satur visu Bootstrap CSS optimizētu versiju. Jūsu projektam pielāgotākai uzstādīšanai ir iespēja konfigurēt un pielāgot Bootstrap ietvaru:

<https://getbootstrap.com/docs/5.1/getting-started/introduction>

Iepazīties ar Bootstrap var W3Schools mācību materiālos:

<https://www.w3schools.com/bootstrap5/index.php>

# Bootstrap adaptīvais režģis

## Konteineris

Lai izmantotu, piem., Bootstrap adaptīvo režģa risinājumu (kolonnām un rindām), nepieciešams konteineris jeb HTML elements ar **.container** klasi.

## Rinda

Lai mājaslapas noformējumu un kolonnas grupētu rindās, nepieciešams izmantot klasi **.row**.

## Kolonna

Bootstrap izmanto 12 kolonnu sistēmu. Izmantojot kādu kolonnas klasēm, iespējams norādīt elementa platumu dažādos lūzuma punktos **.col-\*. \*\***

Dotajā piemērā tiek izveidotas trīs vienāda platumā kolonnas, pielāgojot klases nosaukumu ir iespējams mainīt tās platumu un lūzuma punktus (breakpoints).

- \* - norāda kolonnas lūzuma punktu
- \*\* - norāda cik kolonnu platums piemītīs elementam no 1 līdz 12

```
<div class="container">
  <div class="row">
    <div class="col">
      Viena no trim kolonām
    </div>
    <div class="col">
      Viena no trim kolonām
    </div>
    <div class="col">
      Viena no trim kolonām
    </div>
  </div>
</div>
```

# Bootstrap kolonnas

Kolonu platumu pielāgojam ar numberāciju klases nosaukuma beigās no 1 līdz 12.  
Lūzumpunktiem norādam vienu no 6 Bootstrap piedāvātajiem.

<code>.col-sm-*</code>	≥ 576px
<code>.col-md-*</code>	≥ 768px
<code>.col-lg-*</code>	≥ 992px
<code>.col-xl-*</code>	≥ 1200px
<code>.col-xxl-*</code>	≥ 1400px

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

# Bootstrap konteineri

Arī pielāgojot konteineru klasi ir iespējams iegūt atšķirīgu režģa platumu un uzvedību dažādiem ekrāna platumiem, līdzīgi kā ar kolonnu klašu nosaukumu.

Class	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<b>.container</b>	100%	540px	720px	960px	1140px	1320px
<b>.container-sm</b>	100%	540px	720px	960px	1140px	1320px
<b>.container-md</b>	100%	100%	720px	960px	1140px	1320px
<b>.container-lg</b>	100%	100%	100%	960px	1140px	1320px
<b>.container-xl</b>	100%	100%	100%	100%	1140px	1320px
<b>.container-xxl</b>	100%	100%	100%	100%	100%	1320px
<b>.container-fluid</b>	100%	100%	100%	100%	100%	100%





## Adaptīvie attēli

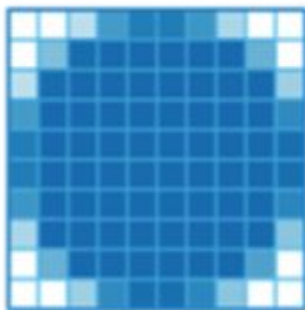
# CSS pikselis un ierīces pikselis

Līdz ar augstas izšķirtspējas ekrāniem, CSS vērtība 1px var atbilst virākiem fiziskajiem ierīces ekrāna pikseļiem. Piem.,:

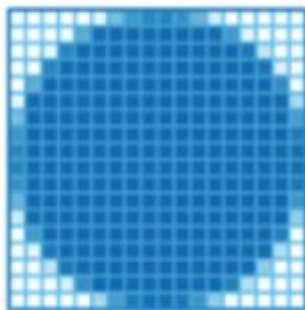
**iPhone 11**, katram CSS pikselim atbilst 4 ekrāna pikseļi

**Phone 11 Pro**, katram CSS pikselim atbilst 4 ekrāna pikseļi

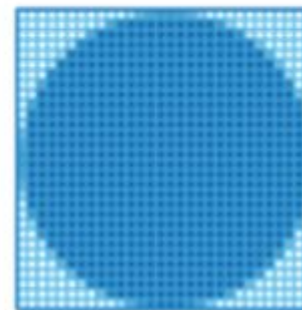
**Samsung Galaxy S10**, katram CSS pikselim atbilst 4 ekrāna pikseļi



Low Density



Medium Density



High Density

# Ekrāna izšķirtspējai pielāgoti attēli

Izmantojot CSS nosacījumus, lūzuma punktus un dažādus HTML elementus vai atribūtus, iespējams ielādēt ekrāna izšķirtspējai atbilstošu attēlu. Piemēram, lai ielādētu lielākas izšķirtspējas attēlu Retina ekrāniem, varam to panākt ar HTML vai CSS:

## CSS

```
.selector { background: url(../path_to_png/image.png); }
```

```
@media (-webkit-min-device-pixel-ratio: 2) {  
    .selector { background-image: url(../path_to_png/image@2x.png); }  
}
```

## HTML

```

```

# Ekrāna izmēram pielāgoti attēli

Izmantojot CSS nosacījumus, lūzuma punktus un dažādus HTML elementus vai atribūtus, iespējams ielādēt ekrāna izšķirtspējai atbilstošu attēlu. Piemēram, lai ielādētu lielākas izšķirtspējas attēlu Retina ekrāniem, varam to panākt ar HTML vai CSS:

## CSS

```
@media only screen and (min-width: 400px) {
  body {
    background-image: url('img_small.jpg');
  }
}
```

## HTML

```
<picture>
  <source media="(min-width:650px)" srcset="red.jpg">
  <source media="(min-width:465px)" srcset="black.jpg">
  
</picture>
```

# Vektoru un raster grafika

Ar raster grafiku saprotam satitskus un izmēru ziņā nemainīgus attēlus. Šādus attēlus samazinot vai palielinot veidosies vizuāli artefakti.

Vektorgrafika savukārt ir dinamiska un attēls ir matemātisku vektoru aprēķins - neatkarīgi no tā cik tas tiks palielināts, vienmēr iegūsim asu attēlu bez vizuāliem artefaktiem.

Mājaslapu izstrādē tiek izmantotas abi grafikas veidi. Attēli visbiežāk būs rastergrafikas formātā - JPG, PNG, WebP u.c.

Savukārt ikonas un fonti ir vektoru grafika. Ikonām bieži izmanto SVG failu formātu.

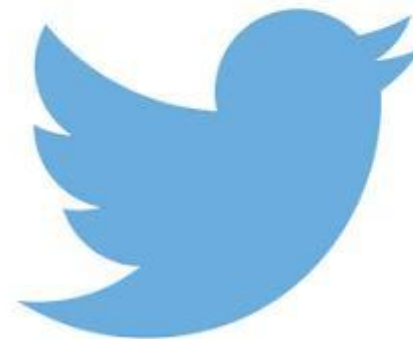
Vector 100%



Raster 300%

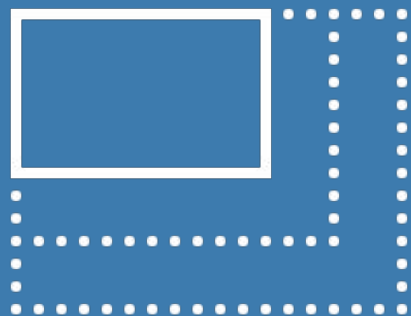


Vector 300%



Raster 300%





## Animācijas un transformācijas

# CSS transformācijas

Ar transformācijām iespējams mainīt HTML elementu attēlojumu 2D un 3D plaknēs - rotēt, mainīt izmēru, pozīciju, augstumu un platumu. Tā kā šīs darbības var ietekmēt jau izveidoto DOM un citu HTML elementu novietojumu vai izmēru, tas var pasliktināt veiktspēju un nav labi piemērotas sarežģītām animācijām. Ar transformācijām var iepazīties [W3Schools](https://www.w3schools.com/css/css3_transformations.asp).

**transform**  
**transition-property**  
**transition-duration**  
**transition-timing-function**  
**transition-delay**  
**transition**

**matrix()**  
**matrix3d()**  
**translate()**  
**translate3d()**  
**translateX()**  
**translateY()**  
**translateZ()**  
**scale()**  
**scale3d()**  
**scaleX()**  
**scaleY()**  
**scaleZ()**

**rotate()**  
**rotate3d()**  
**rotateX()**  
**rotateY()**  
**rotateZ()**  
**skew()**  
**skewX()**  
**skewY()**  
**perspective()**

# CSS animācijas

Animācijas var padarīt lapu pievilcīgāku, bet tām ir arī ļoti praktisks pielietojums - sniegt indikāciju lietotājam, ka norisinās kāds process, piem., datu sūtīšana vai saņemšana no servera. Arī subjektīvi lietotājam šis gaidīšanas laiks šķiet mazāk nepatīkams. Dažāda veida vizuālas izmaiņas un animācijas var veikt, izmantojot CSS transformācijas, tomēr veikspējai labākas ir CSS animācijas - jo tās neietekmē jau izveidoto DOM. Uzskatāmi piemēri atrodami [W3Schools](https://www.w3schools.com/css/css3_animations.asp)

**@keyframes** {...}

**animation-name**  
**animation-duration**  
**animation-iteration-count**  
**animation-direction**  
**animation-timing-function**  
**animation-delay**  
**animation-play-state**  
**animation-fill-mode**  
**animation**

**linear**  
**ease**  
**ease-in**  
**ease-out**  
**ease-in-out**  
**cubic-bezier**





## CSS animāciju un transformāciju piemēri

<https://codepen.io/FilipVitas/pen/GdMbOX>

<https://codepen.io/juliangarnier/pen/krNqZO>

<https://pattle.github.io/simpsons-in-css/>

<https://codepen.io/SampathParavasthu/pen/qBWPabx>