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

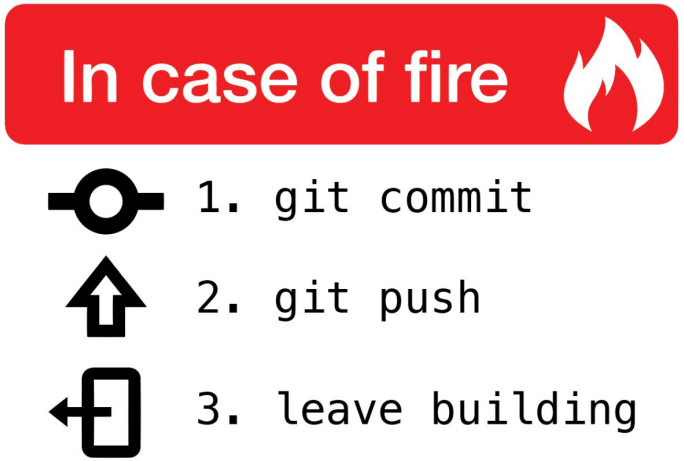
IEGULDĪJUMS TAVĀ NĀKOTNĒ

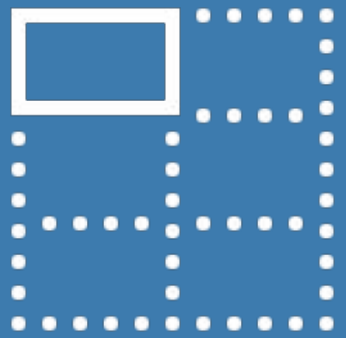
ESF projekts Nr. 8.4.1.0/16/I/001 ''Nodarbināto personu profesionālās kompetences pilnveide"

GIT plūsma (*flow*)

GIT plūsma (*flow*)

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





CSS selektori

CSS sintakse

Ar selektoriem mēs sasaistam HTML elementus ar tiem paredzētām vizuālajām īpašībām un to vērtībām.

```
selektors {
    īpašība: vērtība;
}
```

```
button {
    height: 50px;
}
```

CSS selektori

Selektors ar HTML elementiem sasaista pēc:

Birkām (tags)	`h1 { īpašība: vērtība; }`
Klasēm	`.heading { īpašība: vērtība; }`
Id	`#heading { īpašība: vērtība; }`
Atribūtiem	`a[href="#"] { īpašība: vērtība; }`
Universālais selektors	`* { īpašība: vērtība; }`

CSS selektoru kombinēšana

Vienam selektoram var atbilst vairāki HTML elementi. Ja vēlamies atlasīt specifiskāk - kombinējam selektorus, lai atlasītu tikai nepieciešamos elementus. Ir izmantojami arī selektoru kombinatori (+, >, ~), pseido klases(:before, :after) un citi papildinājumi, tos apskatīsim vēlāk.
Selektoru kombinēšanē pierakstot šādi:

.class1.class2 {}	<div class=**"class1 class2"**>
.class1 .class2 {}	 <div class=**"class2"**>
p.class {}	<p class=**"class"**>
p .class {}	 <em class=**"class"**>
.class1, .class2 {}	<div class=**"class1"**> <div class=**"class2"**>

CSS selektoru kombinēšana

h1	visi **<h1>** elementi
#main	visi elementi ar id **“main”**
.class1	visi elementi ar klasi **“class1”**
.class1.class2	visi elementi ar klasi **“class1”** UN **“class2”**
.class1 .class2	visi elementi ar klasi **“class1”** VAI **“class2”**

Selektoru specificitāte

CSS specificitāte (specificity)

Jo precīzāks tiek norādīts selektors, jo lielāks specifitātes svars tam pakārtotajiem stiliem.
Ja uz vienu elementu attieksies vairāki CSS selektori ar vienādām īpašībām, bet atšķirīgām vērtīgām - attēlots tiks noformējums ar lielāko specifitātes svaru.
Specifitātes grupas:

Zemākais specifitātes svars ir **birkas nosaukumu** selektoriem.
Lielāks svars piemīt **klašu un atribūtu** selektoriem.
Vēl lielāks ir **id** selektoram.

Svarīga ir visu specifitātes grupu kopsuma nosakot kopējo specifitātes svaru attiecībā pret citiem selektoriem.

Stila atribūts (inline-style) > specifitāte

Kad īpašības vērtība tiek noteikta iekš html elementa stila atribūta, tiks izmantots šis stils - neņemot vērā citu stilu selektoru specifitāti.

```
a {
        color: red;
}
```

0.0.1

```
a[href=#].class1 {
        color: black;
}
```

0.2.1

Saite melnā krāsā

```
<a class="class1" href="#">Saite melnā krāsā</a>
```

```
a {
        color: red;
}
```

0.0.1

```
a[href=#].class1 {
        color: black;
}
```

0.2.1`

Saite melnā krāsā

```
<a class="class1" href="#" style="color: green">Saite melnā krāsā</a>
```

!important > stila atribūts (inline-style)

Kad īpašības vērtība tiek noteikta kā !important - tā tiks izmantota, neņemot vērā citu stilu selektoru specifitāti vai inline-style.

```
a {
        color: red;
}
```

0.0.1

```
a[href=#].class1 {
        color: black;
}
```

0.2.1

Saite melnā krāsā

```
<a class="class1" href="#">Saite melnā krāsā</a>
```

```
a {
        color: red !important;
}
```

0.0.1

```
a[href=#].class1 {
        color: black;
}
```

0.2.1

Saite melnā krāsā

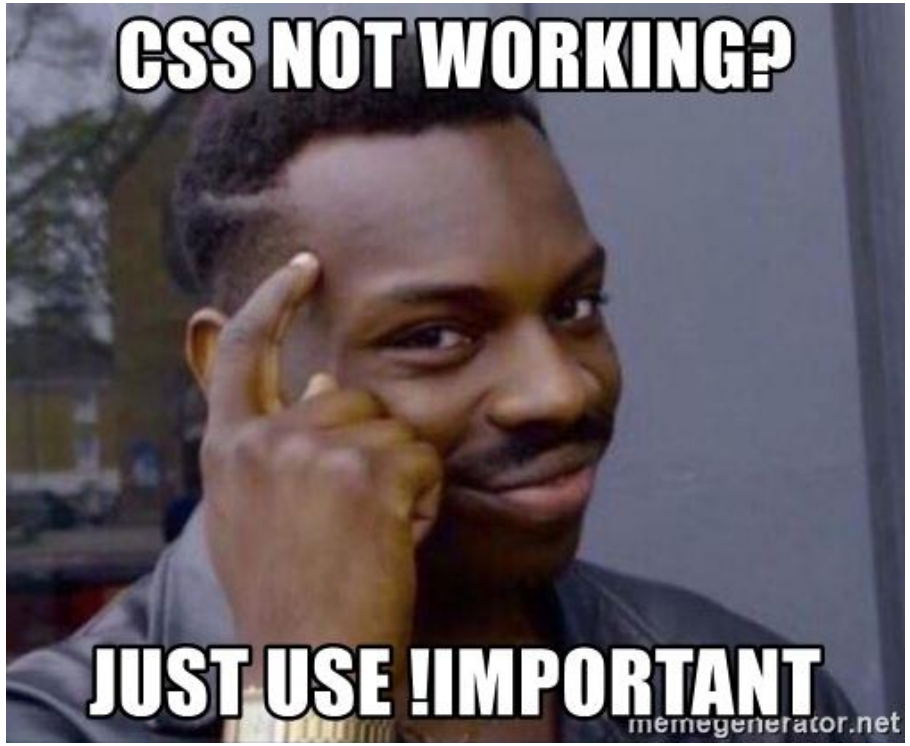
```
<a class="class1" href="#" style="color: green">Saite melnā krāsā</a>
```

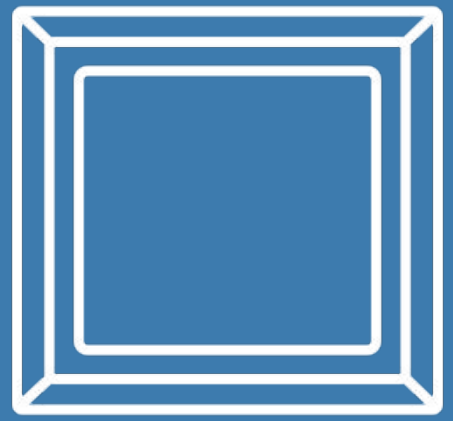
!important pielietojums

!important izmantošana var padarīt kodu grūtāk uzturamu un ierobežo elementa turmpāku noformēšanu.

Tāpēc to izmanto tikai izņēmuma gadījumos.
Ieteicamais pielietojums:

- Meklēt iespēju Izmantot selektoru specificitāti, lai panāktu vēlamo rezultātu, pirms apsvērt !important.
- Izmantot !important tikai, lai pārsvērtu stilus no ārējām bibliotēkām kā, piem., Bootstrap
- Neizmantot !important CSS stilos, kas tiek izmantoti viscaur mājaslapai.





CSS ietvara modelis (box model)

CSS rāmja modelis (box-model)

DOM -> CSSOM -> Render Tree -> **Layout** -> Paint

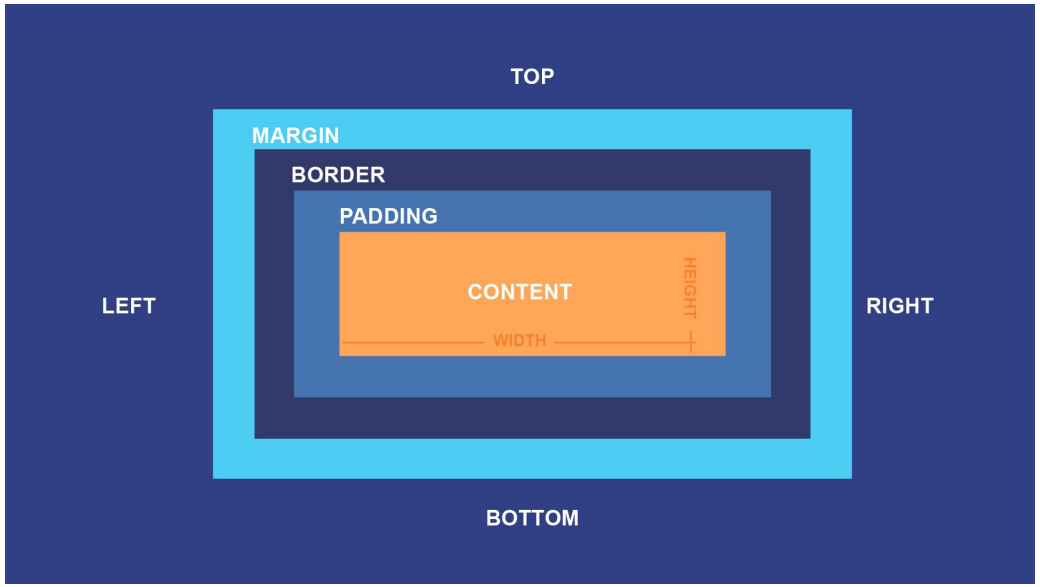
Layout posmā pārlūkprogramma izveido katra Render Tree elementa rāmi. Rāmja laukumu nosaka:

Satura izmērs - **Content**

Paplašinājum izmērs - **Padding**

Atkāpes (margin) izmērs - **Margin**

Robežas izmērs - **Border**



Mērvienības (values and units)

Statiskās mērvienības

px	1 device pixel (also expressed as 0.265mm)
Q	1Q = 1/40 no 1cm
pt	1pt = 1/72 no 1in
pc	1pc = 1/6 no 1in
in	1in = 2.54cm = 96px
cm	1cm = 37.8px = 25.2/64in
mm	1mm = 1/10 no 1cm

Dinamiskās mērvienības

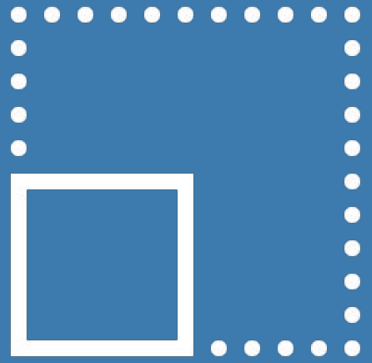
em	vecāka (parent element) fonta izmērs
rem	html elementa fonta izmērs
vw	1vw = 1% no mājaslapas loga platuma
vh	1vh = 1% no mājaslapas loga augstuma
vmin	1vmin = 1% no mājaslapas loga īsākās malas
vmax	1max = 1% no mājaslapas loga augstākās malas
%	1% = 1% no parent element sakrītošā izmēra

Īpašības (properties)

Kura ietvarā mums svarīgas CSS īpašību grupas būs:

animācijas - animation, transition
transformācijas - transform
fons - background
teksta noformējums - font-size, color
izmēri - width, height, padding, margin
pozīcija - position
saraksti - list
režģī - grid
flex-box - flex

Šie ir tikai pāris būtisko īpašību nosaukumi. Visbiežāk sastopamo CSS īpašību sarakstu var apskatīt MDN mājasllapā.



Adaptīvais rāmis (flex-box)

Parastie rāmji (block un inline)

HTML elementi iedalās bloka (block-level) un rindas (inline) līmeņa rāmjos.

Block-level - vienmēr atradīsies jaunā rindā un pārlūkprogramma automatīski piešķirs tam atkāpes (margin). Bloka līmeņa element vienmēr tiecās aizpildīt visu tiem pieejamo platumu jeb 100%.
Divi visbiežāk sastopamie bloka līmeņa elementi:

<p>
** **

```
selector {
  display: block;
}
```

Inline - nesākas jaunā rindā. Aizņem tikai tik daudz platumu, cik nepieciešams. Nedrīkst saturēt bloka līmeņa elementu.
Piem.,:

****

```
selector {
  display: inline;
}
```

Adaptīvais rāmis (flex-box)

Flex-box ir CSS rīks ar ko elementiem piešķirt adaptīvas īpašības un ērtāk tos izkārtot.
Elements ar īpašību **dispaly**: **flex** padara to par flex-box un pakārto tā ietvertos elementus flex-box noformējumam.
Pilns flex-box pielietojuma apskats pieejams [css-tricks.com.](css-tricks.com)

```
<div id="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

#flex-container {
  display: flex;
}
```

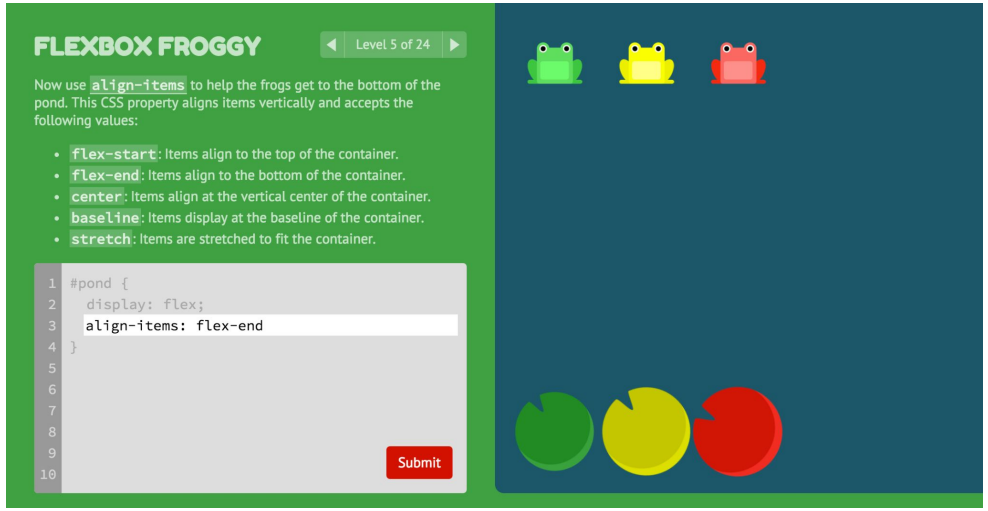
Biežāk izmantotās flex-box īpašības

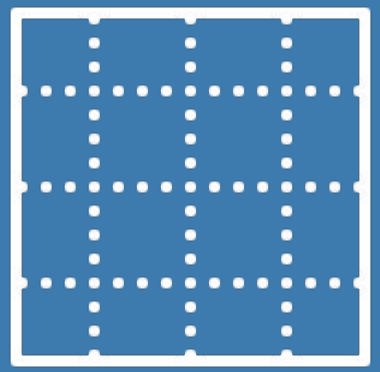
Lai izprasu dažādās flexbox īpašības, izmēģiniet CSS spēli [Flexbox froggy](#).
Dažas no visbiežāk izmantotajām flex-box īpašībām.

order - noteikt elementu kārtību

flex-direction - kārtot elementus horizontāli vai vertikāli
justicy-content - noteikt elementu horizontālo līdzinājumu
align-items - noteikt elementu vertikālo līdzinājumu

flex-gorw - elementu izmēra attiecība izplešoties
flex-shrink - elementu izmēra attiecība sašaurinoties
flex-basis - sākotnējais elementa izmērs





Režģis (grid)

Režģis (grid)

Režģis ir CSS rīks ar ko veidot lapas izklājumu rindās un kolonās, kas ir arī adaptīvas ekrānam izmēram.
Elements ar īpašību **dispaly**: **grid** padara to par režģi (grid).
Pilns grid pielietojuma apskats pieejams [css-tricks.com.](#)

```
<div id="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

#grid-container {
  display: grid;
}
```

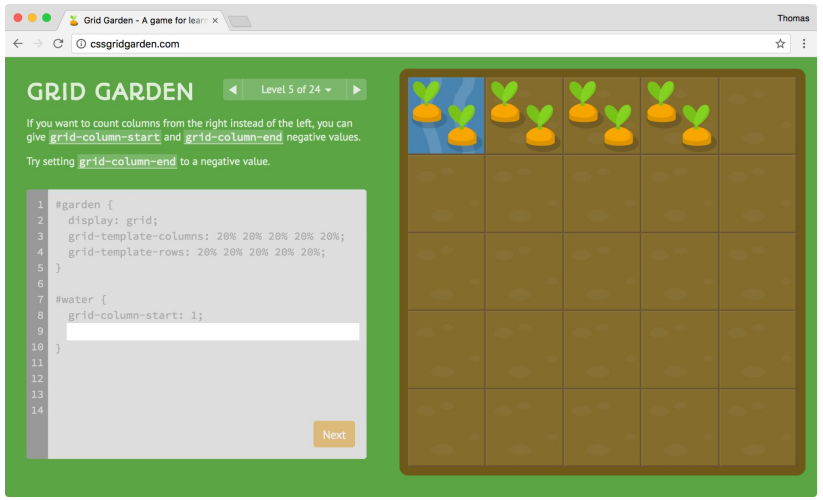
Biežāk izmantotās flex-box īpašības

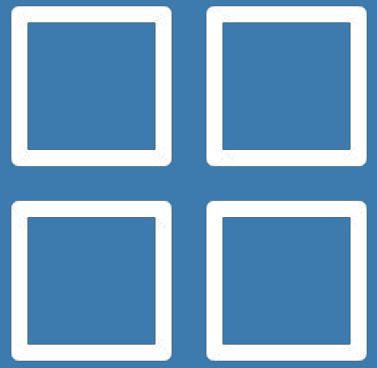
Lai izprasu dažādās flexbox īpašības, izmēģiniet CSS spēli [Grid garden](#).
Dažas no visbiežāk izmantotajām flex-box īpašībām.

grid-template-columns - uzstādīt režģa kolonas
column-gap- kolonu atstarpe
grid-template-rows - uzstādīt režģa rindas
row-gap - rindu atstarpe

justify-content - elementu izkārtojums rindā
align-content - elementu izkārtojums kolonā

grid-column - elementu novietojums kolonā
grid-row - elementu novietojums rindā





CSS dokumentācija

Interneta resursi

Oficiālā dokumentācija
https://www.w3.org/Style/CSS/specs.en.html
https://www.w3.org/TR/?tag=css

W3Schools resursi:
https://www.w3schools.com/cssref/default.asp

Can I use..
https://caniuse.com/