

# FRONTEND MĀJASLAPAS IZSTRĀDE

## 11. lekcija - mājaslapas izstrāde un labākā prakse



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



VADĪBAS UN  
UZNĒMĒJDARBĪBAS  
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"



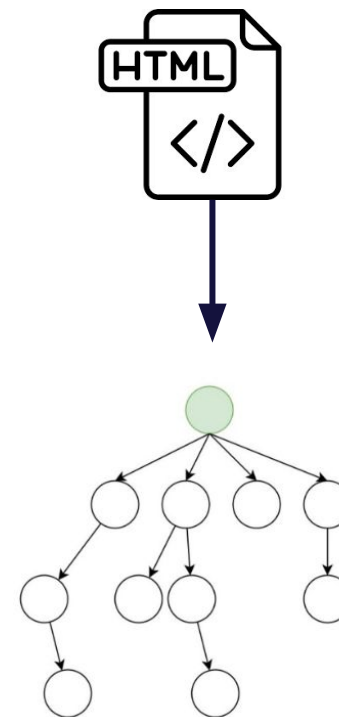
## Pārlūkprogrammas darbības

# HTML apstrādes posmi

Izstrādājot kodu vienmēr būtiski apzināties visus HTML apstrādes soļus un procesus, kas to var ietekmēt:

1. **Fetching**
2. **Parsing**
3. **CSSOM izveide**
4. **Render Tree izveide**
5. **Layout**
6. **Paint**

Sevišķi bieži rodās problēmas ar mājaslapas veiktspēju tieši **Layout** solī, kad neuzmanīgi mainam kāda elementa vizuālo noformējumu, kas liek atkārtoti aprēķināt layout arī visiem pārējiem elementiem.



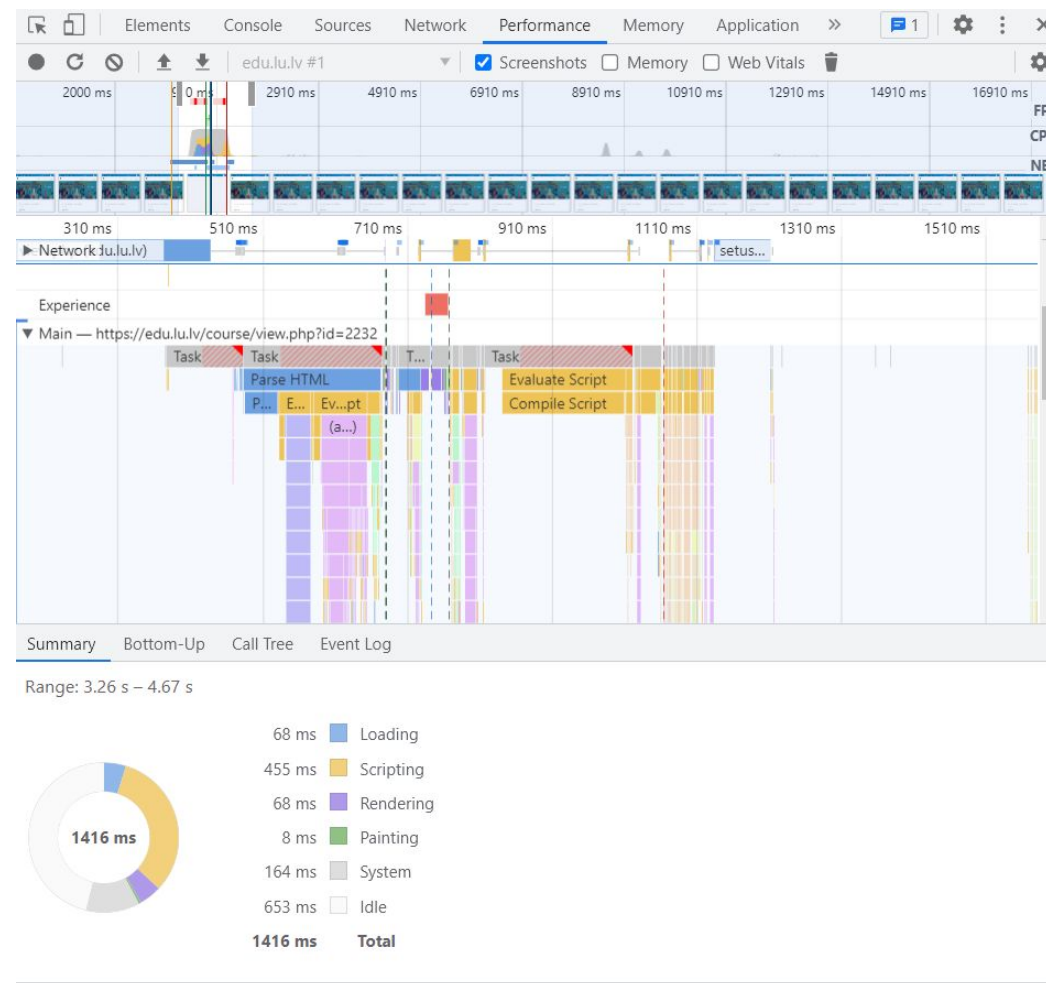
# Performance cilne



FRONTEND mājaslapas izstrāde

Iespējams ierakstīt un apskatīt mājaslapas veikspēju un visus tās ietekmējošos faktoros.

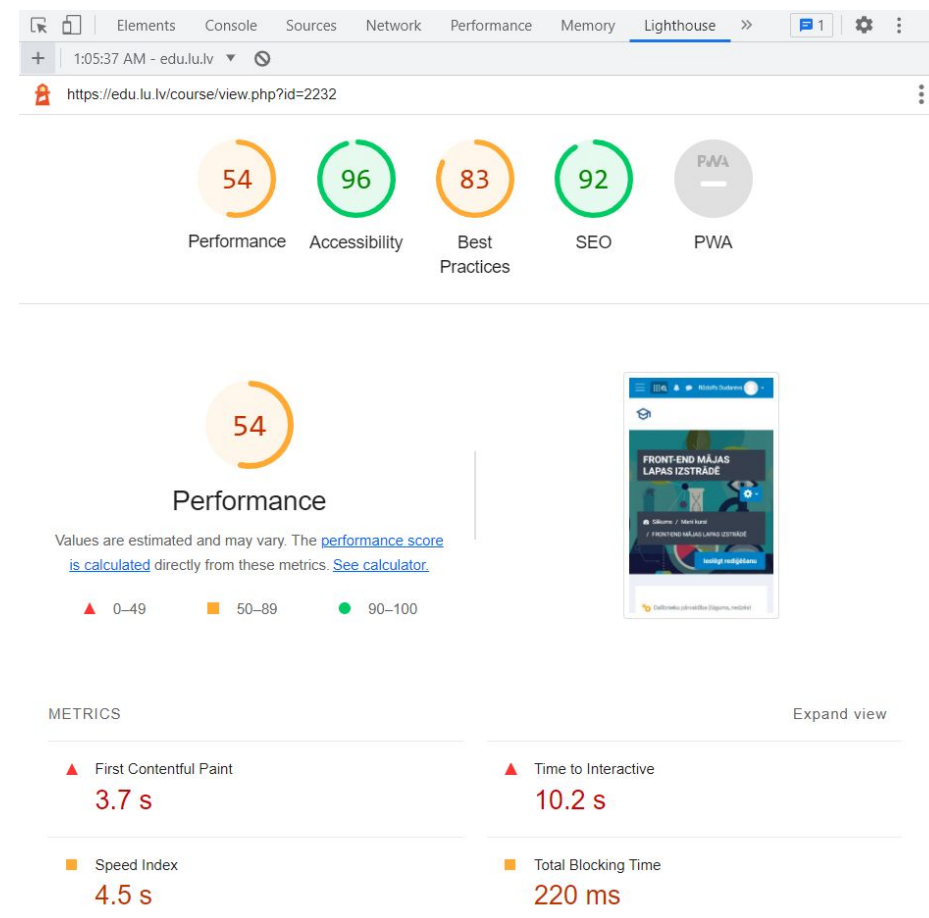
Jo sevišķi būtiski šī cilne palīdz atrast problēmas ar kādu no jau iepriekš apskatītajiem mājaslapas ielādes posmiem, piem., rendering (DOM, CSSOM, render tree, layout) un painting. Kā arī novērot JS darbības un to ietekmi uz veikspēju.



# Veiktspējas izpēte ar LightHouse

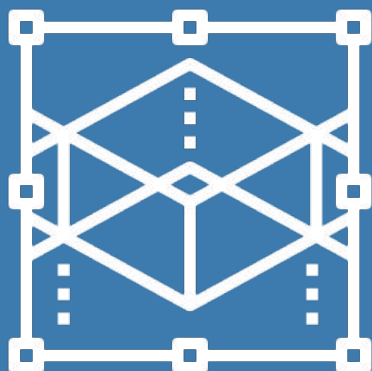
LightHouse cilne piedāvā aplūkot dažādus mājaslapas veiktspējas aspektus un tiek piedāvāti arī konkrēti iespējami risinājumi, lai tos uzlabotu

Kopējais audits mājaslapai norādīs arī pielāgotību meklēšanas rīkiem(*crawlers, bots*) un cilvēkiem ar redzes vai dzirdes traucējumiem.



Izmantojot jaunākus HTML, CSS vai JavaScript papildinājumus, ļoti svarīgi pārliecināties par to pieejamību un atbalstu pārlūkprogrammās.

[illegible]



## CSS darbības

# Klases definēšana

Izstrādājot CSS svarīgi pareizi izmantot selektorus un to kombinācijas, lai noteiktu konkrētā CSS specifitāti un meklētos elementus.

Pēc iespējas izmantojam klašu un id selektorus. Tikai izņēmuma gadījumos, kad jāpārraksta kāds CSS - izmantojam !important. Inline-style galvenokārt izmanto, lai mijiedarbotos ar elementa vizuālos noformējumu ar JavaScript.

<code>h1</code>	visi <b>&lt;h1&gt;</b> elementi
<code>#main</code>	visi elementi ar id <b>"main"</b>
<code>.class1</code>	visi elementi ar klasi <b>"class1"</b>
<code>.class1.class2</code>	visi elementi ar klasi <b>"class1"</b> UN <b>"class2"</b>
<code>.class1 .class2</code>	visi elementi ar klasi <b>"class1"</b> VAI <b>"class2"</b>

!important	inline-style	id	class, attribute	tag
0	0	0	0	0
specifitāte (specificity)				



# Ekrānam adaptīvs CSS

Lai noformētu CSS pielāgoti ekrāna izmēram, varam izmantot `@media`. Dotajā piemērā elementi ekrānā aizņems 100% platumu līdz izpildīsies nosacījums - ekrāna izmērs ir vismaz 1200px.

Ieteicams vienmēr izstrādāt kodu ar mobile first pieeju un tad to pielākot desktop ekrāniem

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

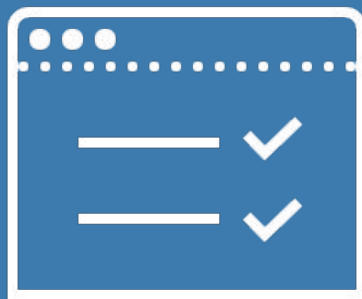
```
@media only screen and (min-width: 1200px) {  
    .nav,  
    .widget {  
        width: 25%;  
    }  
  
    .content {  
        width: 50%;  
    }  
}
```



## Rīki CSS izstrādei

Lai atvieglotu CSS izstrādi, ir pieejami dažādi rīki ar jau gataviem šabloniem un rīkiem. Kursa ietvarā izmantojām Bootstrap, bet katra projekta īpatnību dēļ cits rīks varētu būt pat labāk piemērots.





## JavaScript koncepti

# JavaScript darbības

JavaScript varam izveidot mainīgos, kas saturēs kāda tipa vērtību. Ieteicams izmantot **let** un **const** novecojušās **var** deklarācijas vietā.

Kad nepieciešams veikt kādu darbību atkārtoti, jo sevišķi strādājot ar masīviem - izmantojam ciklus **forEach**, **for**, **while** u.c.

Lai apkopotu kodu, kas tiks izmantots atkārtoti, izveidojam funkcijas.

Funkcijas deklarējam ar sintaksi **function funcName()** **{ }** vai **() => { }** atkarībā no tā vai mums nepieciešama **this** atslēgvārda sasaiste ar doto scope.

# Modulārs kods ar klasēm

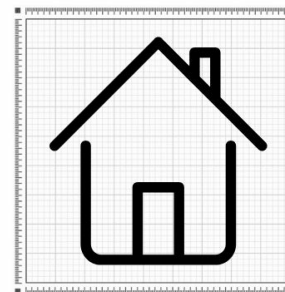
Klase ir šablons objektu izveidei. Klasi var uztvert kā objekta rasējumu. Definējam klasi vienreiz - izveidojam cik vien objektus nepieciešams, izmantojot definēto klasi jeb rasējumu.

Katrs jaunizveidotais objekts ir klases instance un ir nesaistīts ar citām klases instancēm.

Izmantojot klases varam iekapsulet sarežģītu loģiku un ietvert to atsevišķā failā. Tādējādi veidojam modulāru kodu, ko būs ērti izmantot atkārtoti un uzturēt.

```
class ClassName {  
    constructor() {  
        // setting class properties  
    };  
  
    // define class methods  
}
```

House Class



House Instance



House Instance



# Saziņa ar API

Gluži kā mājaslapas apmeklētājam ir pieejamas pogas, izvēlnes un ievadlauki, lai iegūtu vēlamu informāciju - no JavaScript puses mums ir pieejams līdzvērtīgs interfeiss serverī ar kuru varam mijiedarboties jeb API.

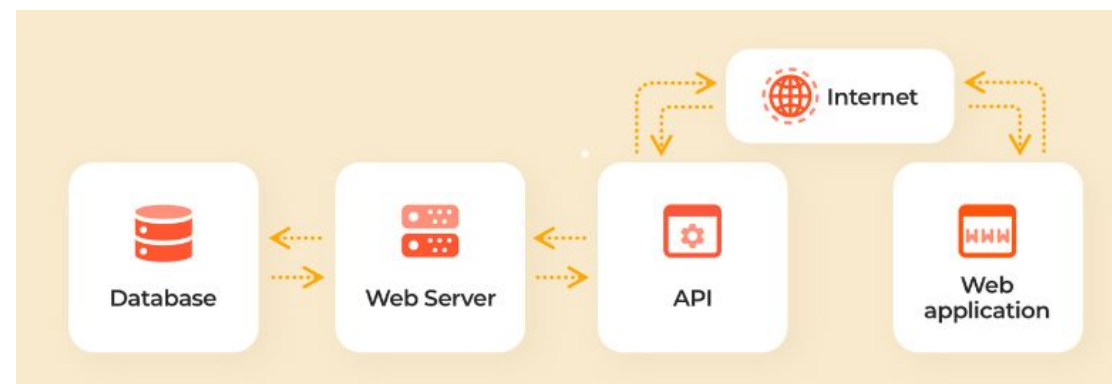
Pieprasot un sūtot datus serverim no pārlūkprogrammas, izmantojam servera API (Application Programming Interface) jeb servera lietojumprogrammas interfeisu.

Serveris veic datu iegūšanu un saglabāšanu datubāzē - katram API piemīt nosacījumi kā pareizi pieprasīt un nosūtīt datus no pārlūkprogrammas, lai serveris varētu atgriezt aplikācijai nepieciešamo informāciju.

Katras aplikācijas servera konfigurācija un formāts datu pieprasīšanai/sūtīšanai būs atšķirīgs un dokumentēts katrā attiecīgajā projektā.

Lekcijas piemērā izmantosim bezmaksas API, lai demonstrētu šo saziņu starp pārlūkprogramm un serveri:

[Randomuser.me dokumentācija](https://randomuser.me)






# Rīki datu saglabāšanai

Izstrādājot sarežģītākas mājaslapas, var būt nepieciešams saglabāt datus pārlūkprogrammā, lai uzlabotu lietotāja pieredzi un iegūtu analītikas datus.

Trīs ērtākie veidi kā to panākt atšķirās to pielietojumā un īpašībās:

- Lokālā glabātuve (local storage)
- Sesijas glabātuve (session storage)
- Sīkdatnes (cookies)

Criteria	Local Storage	Session Storage	Cookies
Storage Capacity	5-10 mb	5-10 mb	4 kb
Auto Expiry	No	Yes	Yes
Server Side Accessibility	No	No	Yes
Data Transfer HTTP Request	No	No	Yes
Data Persistence	Till manually deleted	Till browser tab is closed	As per expiry TTL set

# VUMC

VADĪBAS UN  
UZŅĒMĒJDARBĪBAS  
MĀCĪBU CENTRS

NACIONĀLAIS  
ATTĪSTĪBAS  
PLĀNS 2020



**EIROPAS SAVIENĪBA**  
Eiropas Sociālais  
fonds

---

I E G U L D Ī J U M S   T A V Ā   N Ā K O T N Ē

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

VUMC

VADĪBAS UN  
UZŅĒMĒJDARBĪBAS  
MĀCĪBU CENTRS

Programmas nosaukums