

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 iestatīs jaunizveidoto branch kā aktīvo:

git checkout -b *branch-name*

4. Veikt imziņas failos jaunajā branch.

5. Pievienot visas izmaiņas augšupielādes sarakstam (stage changes). Simbols * šajā komandā nozīmē, ka tiks iekļauti visi faili ar izmaiņām - tā vietā iespējams norādīt arī konkrētu failu, piem., **index.html**:

git add *

6. Saglabāt aktīvajā branch veiktās izmaiņas no augšupielādes saraksta un apkopot tās vienumā(*commit*):

git commit -m "*commit message*"

7. Augšupielādēt saglabātās izmaiņas (*commit*), kas pievienotas augšupielādes sarakstam (*staged changes*):

git push

In case of fire 



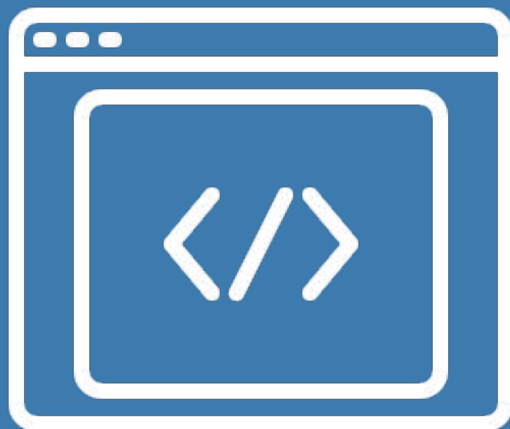
1. git commit



2. git push



3. leave building



HTML vēsture (*Hyper Text Markup Language*)

HTML vēsture (Hyper Text Markup Language)



FRONTEND mājaslapas izstrāde

1991. - HTML prototips

Tim Berners Lee izveido HTML valodu, [pirmo pārlūkprogrammu](#) un [pirmo serveri](#).

Šajā laikā Tim strādā par fiziķi CERN institūtā Šveicē. Tim sāk prototipēt sistēmu ar ko zinātnieki varētu vienuviet apkopot un dalīties ar informāciju, datiem un pētījumiem.

Tim nevēlas vienkārši milzīgu apjomu ar failiem vienuviet, bet gan rīku ar ko informācijas tekstā varētu izmantot saiti (*HyperLink*) uz citu failu - pētījuma atsauci vai tā datiem. No šī koncepta tiek radīta HTML valoda ar *HyperLink* funkcionalitāti. HTML tiek balstīts uz jau plaši izplatītas strukturēta teksta pieraksta metodes datoros SGML (*Standard Generalized Mark-up Language*) - sintakse ar birkām (*tags*) aizgūta no šī pieraksta.

Lai demonstrētu sistēmas prototipa darbību, Tim izveido arī pirmo serveri un pārlūkprogrammu. Kā arī HTTP protokolu to saziņai un URL formātu saitēm. Šo sistēmu kopumus un nākotnes pielietojums tiek nodēvēts par World Wide Web.



Tim Berners Lee

WWW - world wide web

URL - uniform resource locator

HTTP - hypertext transfer protocol

HTML - hypertext markup language

HTML vēsture

(Hyper Text Markup Language)

1993. - HTML 1.0

HTML 1.0 kļuvis publiski pieejams.

Tiek izstrādātas vairākas pārlūkprogrammas(browser), bet pagaidām tos izmanto neliels skaits cilvēku.

Līdz šim brīdim Tim Berners Lee jau bija publicējis [pasaulē pirmo mājaslapu](#) un izveidojis aktīvu epasta sarakstes grupu ar citiem tehnoloģiju entuziastiem, lai apspriestu un pilnveidotu World Wide Web rīkus. Sarakstē Tim iepazīstās ar Dan Connolly, kurš kļuva par vienu no svarīgākajiem cilvēkiem web tehnoloģiju turpmākajā attīstībā.

1994. - HTML 2.0

Apkopojot visus aktuālos HTML papildinājumus, ko ieviesa dažādās pārlūkprogrammas, Dan Connolly publicē HTML 2.0 standarta dokumentāciju. Lai nodrošinātu atvērtu HTML standartu turpmāku izstrādi, Tim Berners Lee izveido [World Wide Web Consortium \(W3C\)](#).



HTML vēsture

(Hyper Text Markup Language)



FRONTEND mājaslapas izstrāde

1997. - HTML 3.2

HTML standartu dokumentācija tiek papildināta ar ievades formām, tabulām un nelielu CSS funkcionalitātes atbalstu.

Šī ir pirmā dokumentācijas versija ko centralizēti izstrādāja W3C.

1999. - HTML 4.01

Tiek definēts plašāks atbalsts multimediju izmantošanas iespējām, CSS, skriptiem un veikti uzlabojumi HTML veiktspējā (*performance*).

2014. - HTML 5

Šobrīd aktuālā HTML standarta versija.

Dokumentācija tiek papildināta ar jaunām birkām(tags), HTML tiek vienkāršots un padarīts uzturamāks. HTML dokumentā var iegult(*embed*) audio un video bez papildus rīkiem.

HTML





Dokumentācija un standarti

HTML dokumentācija un standarti

WHATWG

Lai pārstāvētu savas un lietotāju intereses pārlūkprogrammu izstrādē un pielietojumā, vairāki *Apple*, *Mozilla* un *Opera* darbinieki dibināja WHATWG (Web Hypertext Application Technology Working Group). Tieši WHATWG izstrādāja HTML5 standartu. W3C sadarbībā ar WHATWG izstrādā un publicē HTML oficiālo tehnisko dokumentāciju - [HTML living standards](https://html.spec.whatwg.org/).

W3Schools

Organizācija nav saistīta ar W3C vai WHATWG, tomēr nodrošina vērtīgu un bezmaksas informāciju par HTML un citu tehnoloģiju pielietojumu. To nevajadzētu pielīdzināt oficiālai dokumentācijai, bet var noderēt. Ļoti pieejama frontend iesācējiem.
<https://www.w3schools.com/>

MDN web docs

Mozilla pārlūkprogrammas uzturēts informācijas resurs par HTML, CSS un JS. Nav oficiāla dokumentācija, bet satur ļoti izsmelšu un samērā viegli uztveramu informāciju.
<https://developer.mozilla.org/>



1. W3C un WHATWG izstrādātā HTML standartu papildinājumu tehnisko dokumentāciju un to publicē.
2. Pārļūkprogrammas ievieš atbalstu jaunajiem HTML standartiem pēc to publicēšanas.

Atbalstu(*support*) jauniem HTML, CSS un JS standartiem pārlūkprogrammas ievieš pēc saviem ieskatiem un iespējām. Par to atbalstu dažādās pārlūkprogrammās var pārliecināties ar rīku [Can I use...](#)

[illegible]



No HTML līdz mājaslapai

HTML apstrādes posmi

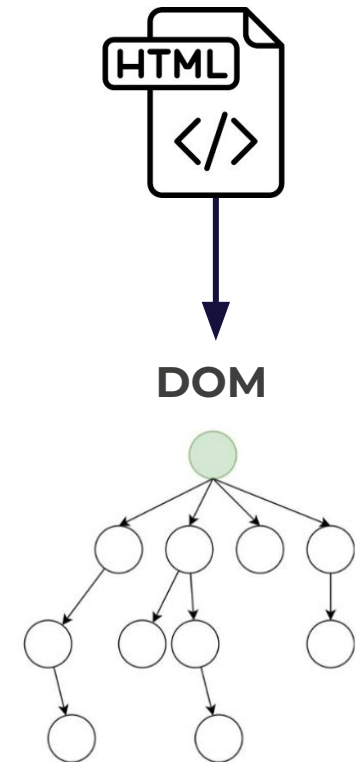
1. Fetching - pārlūkprogramma vēršas pie servera, lai iegūtu HTML dokumentu.

2. Parsing - tiklīdz pārlūkprogramma saņem HTML dokumentu, tā uzsāks dokumenta parsēšanu (sintaktiski analizēt).

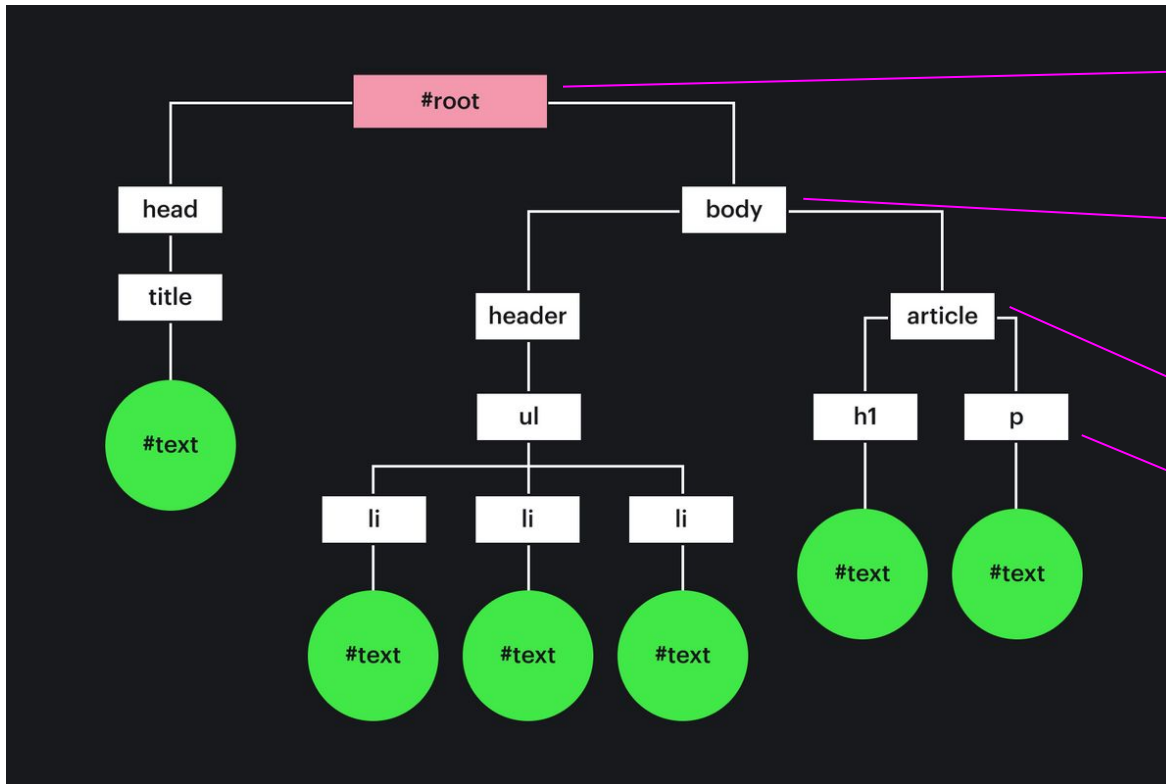
2.1 **DOM** (Document Object Model) - pārlūkprogramma izveido mājaslapas reprezentāciju objektu struktūrā.

Lietotājs saprot un strādā ar mājaslapu, izmantojot pogas, izvēlnes, ievades laukus, attēlus un tekstu. Pārlūkprogramma ar mājaslapu mijiedarbojās izmantojot DOM.

DOM izveide ir inkrementāla jeb tiek parsēta viena koda rindiņa un tad nākamā. Ja tiek atrasta saite uz kādu ārēju resursu - tas tiek ielādēts. Daži ārējie resursi var bloķēt DOM izveidi. piem., JS, CSS un fonti - vēlāk kursā apskatīsim kā no tā izvairīties.



DOM (Document Object Model)



Sākot ar dokumenta sakni (root), kas vienmēr būs `<html></html>` elements, pārlūkprogramma izveido DOM.

```

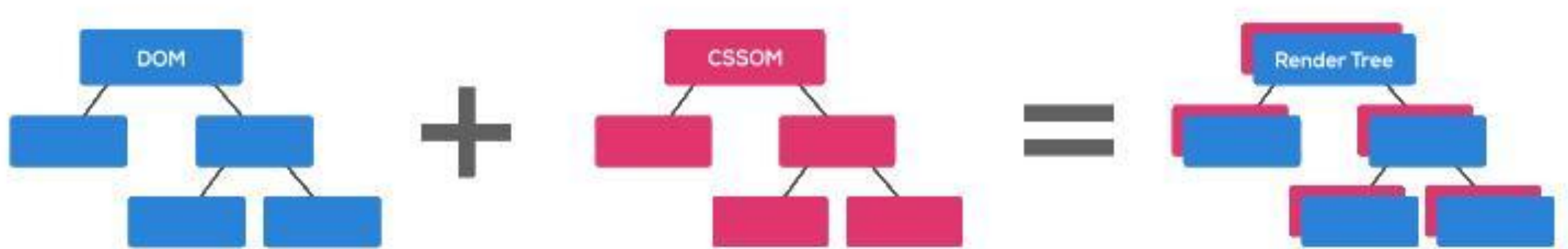
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <header>
      <ul>
        <li>Menu item 1</li>
        <li>Menu item 2</li>
        <li>Menu item 3</li>
      </ul>
    </header>
    <article>
      <h1>Heading</h1>
      <p>Paragraph of text.</p>
    </article>
  </body>
</html>
  
```

CSSOM (CSS Object Model) un Render Tree

3. CSSOM (CSS Object Model) - līdzīgi kā tiek iegūts DOM, izmantojot CSS tiek izveidota objektu struktūra mājaslapas vizuālajam noformējumam.

Atšķirībā no DOM, tā izveide nav inkrementāla un jebkādas turpmāka parsēšanas darbību izpilde tiek bloķēta līdz CSSOM ir izveidots.

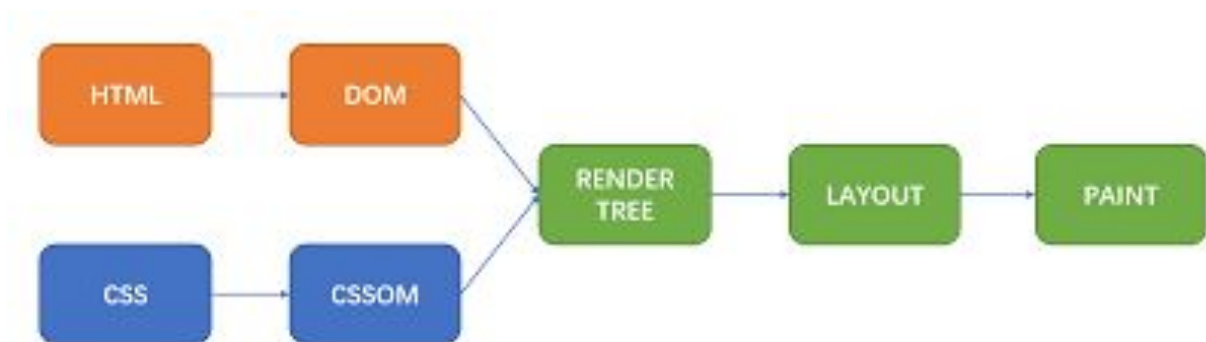
4. Render tree - apvienojot DOM un CSSOM objektu struktūras, pārlūkprogramma izveido jaunu struktūru, kur CSS noformējums ir sasaistīts ar attiecīgajiem HTML elementiem. Render tree tiek iekļauti tikai redzamie elementi - kādam HTML elementam var būt piešķirts noformējams, ka tas nav redzams - šis elements netiek iekļauts render tree.



Layout un Paint

5. Layout - izmantojot render tree struktūru layout posmā pārlūkprogramma aprēķina kur ir novietoti HTML element, kāds ir to izmērs un to attiecība vienam pret otru. Jo vairāk elementi ir DOM struktūrā, jo ilgāks laiks būs nepieciešams šiem aprēķiniem. Kad lietotājs samazina pārlūkprogrammas loga izmēru vai, piem., pagriež telefonu horizontāli kā arī ar JS vai CSS mainot render tree struktūru - atkal ir jāveic visi šie aprēķini, layout posms jāatkārto. Lai optimizētu mājaslapas darbību, ļoti jāizvairās no darbībām, kas izsauc atkārtotu layout posmu.

6. Paint - paint posmā pikseļi tiek aizpildīti ar krāsu un attēloti uz ekrāna. Pārlūkprogrammas ir pārkrāsos tikai tos pikseļus, kas nepieciešami dotajā brīdī, piem., tikai pogu virs kuras atrodas kursora un tai mainās krāsa.





Pārlūkprogrammas rīki

Chrome developer tools

Visās pārlūkprogramma ir pieejami rīki ar ko var iegūt informāciju par jebkuras mājaslapas veiktspēju(*performance*), pielāgotību meklēšanas rīkiem (piem., Google), izmantotajiem interneta resursiem, saziņu ar serveri un citām darbībām ko veic pārlūkprogramma.

Kursa norisē izmantosim Chrome pārlūkprogrammu un tās izstrādātāju rīkus ērtības un plašā pielietojuma dēļ.

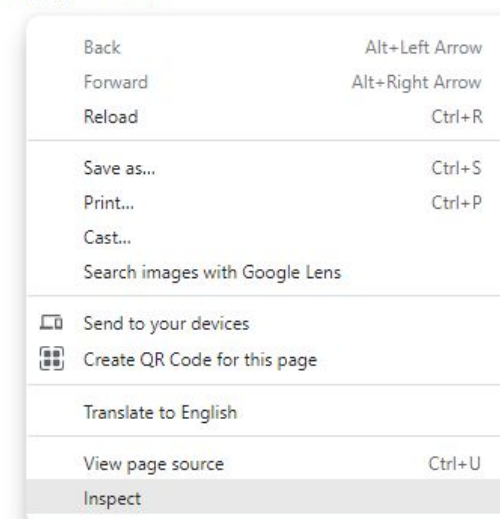
Lai piekļūtu Chrome Developer Tools, jebkur pārlūkprogrammā atvērtās mājaslapas logā:

- Spiežam peles labo taustiņu
- Izvēlamies “Inspect”

- Menu item 1
- Menu item 2
- Menu item 3

Heading

Paragraph of text.



Elements cilne

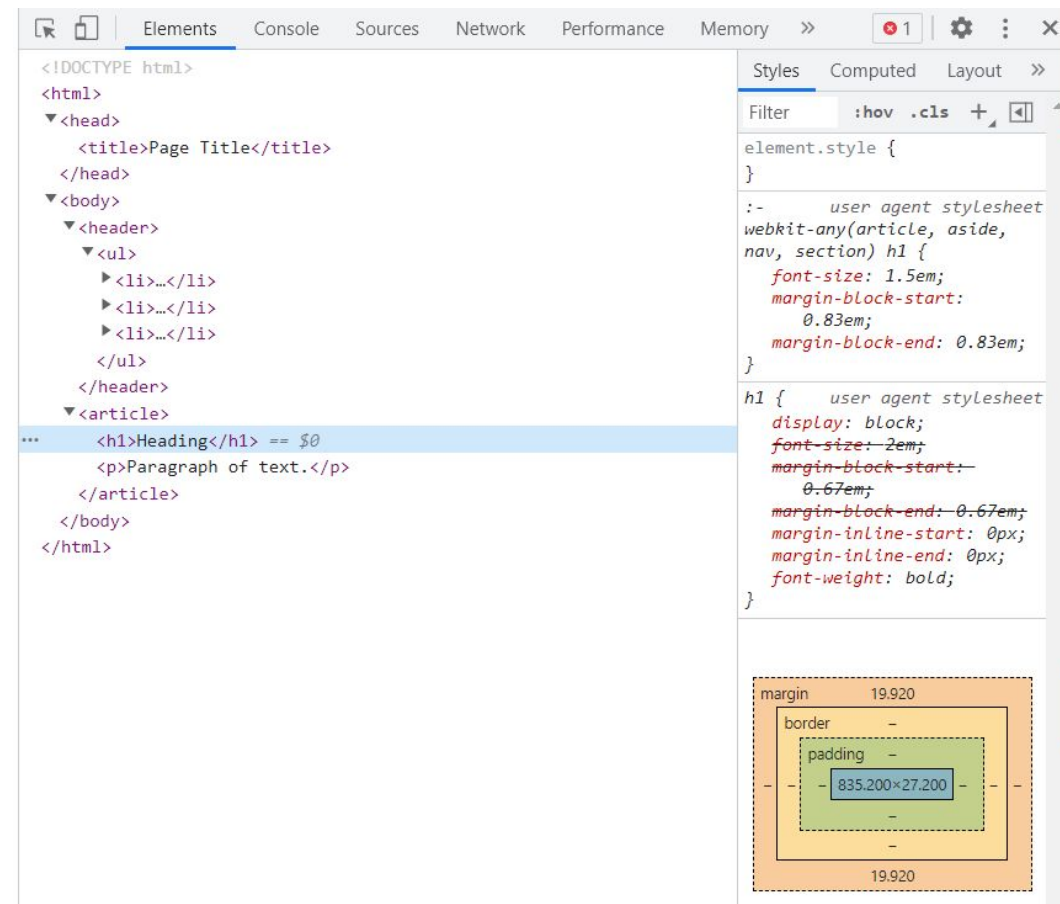


FRONTEND mājaslapas izstrāde

Kreisais panelis satur vizuālu reprezentāciju DOM struktūrai.

Labā sāna panelī iespējams apskatīt CSS noformējumu attiecīgajam HTML elementam.

Šo abu paneļu kopumu var uzskatīt par render tree vizuālu reprezentāciju.



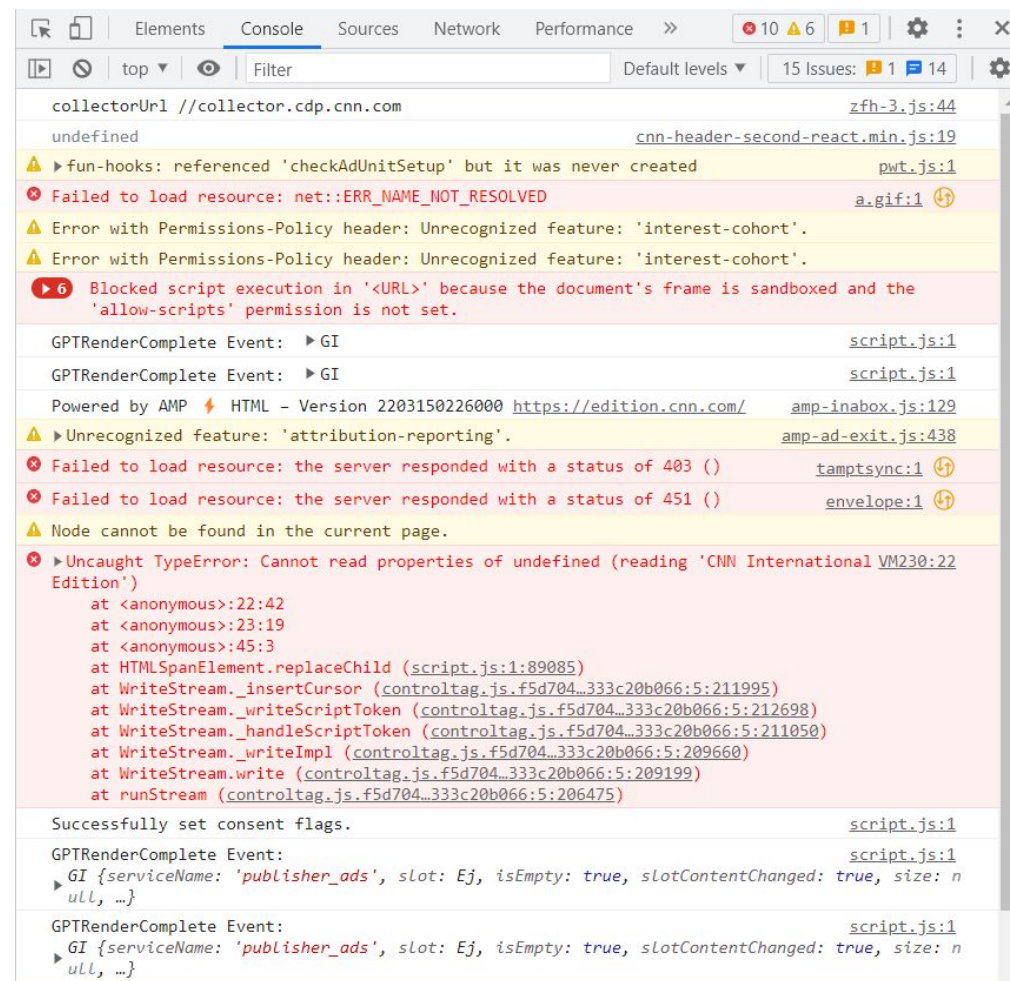
Console cilne

Console cilnē iespējams apskatīt visas ziņas un kļūdu paziņojumus, ko pārlūkprogramma vai izstrādātāja rakstīts JS izvada konsolē.

Šeit iespējams arī tiešā veidā mijiedarboties ar mājaslpu, rakstot un izpildot JS kodu.



FRONTEND mājaslapas izstrāde



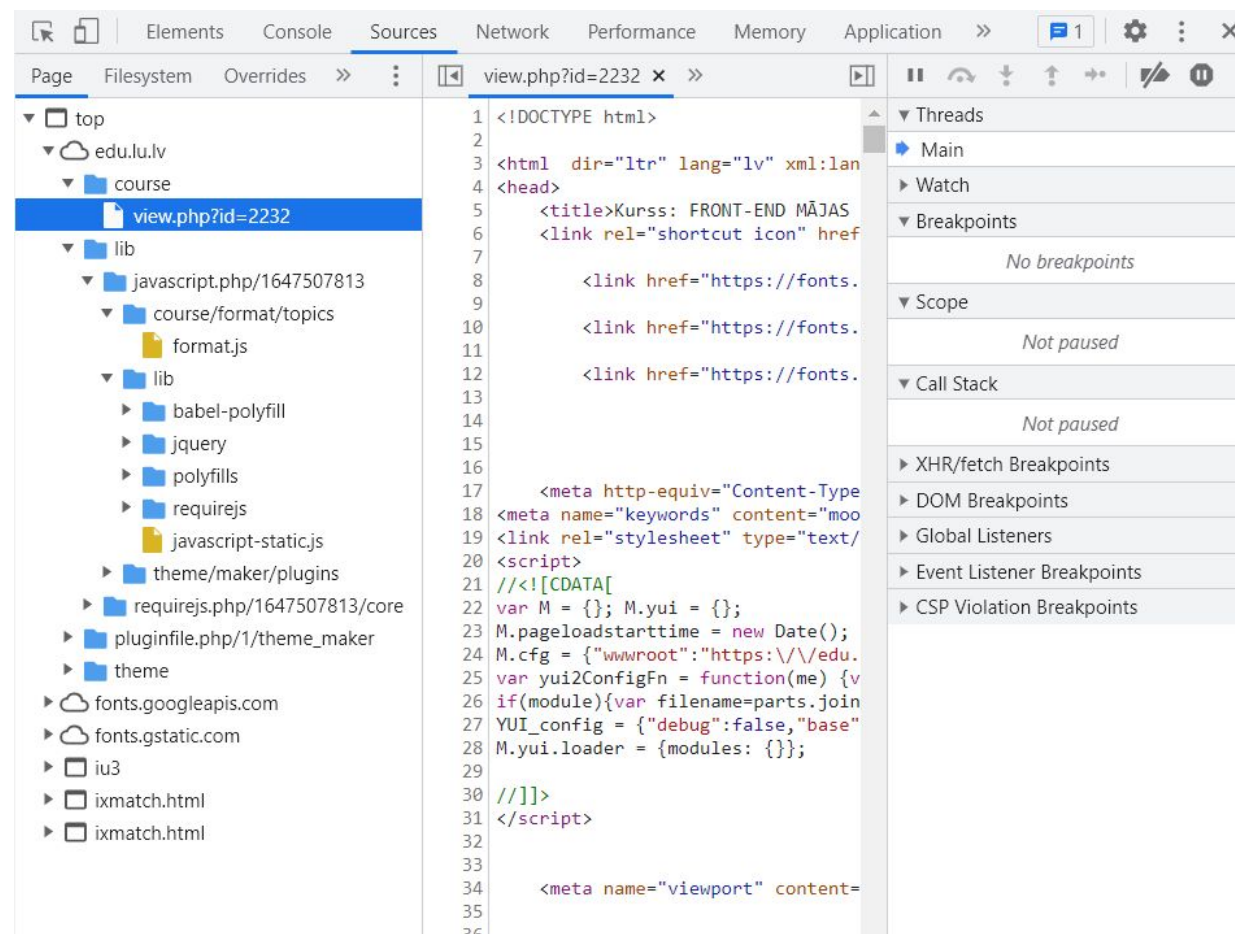
Sources cilne



FRONTEND mājaslapas izstrāde

Var aplūkot visus mājaslapā izmantotos interneta resursus un ar tiem mijiedarboties.

Pieejami rīki JS koda izpētei - mēdz noderēt meklējot kļūdas(*bugs*).

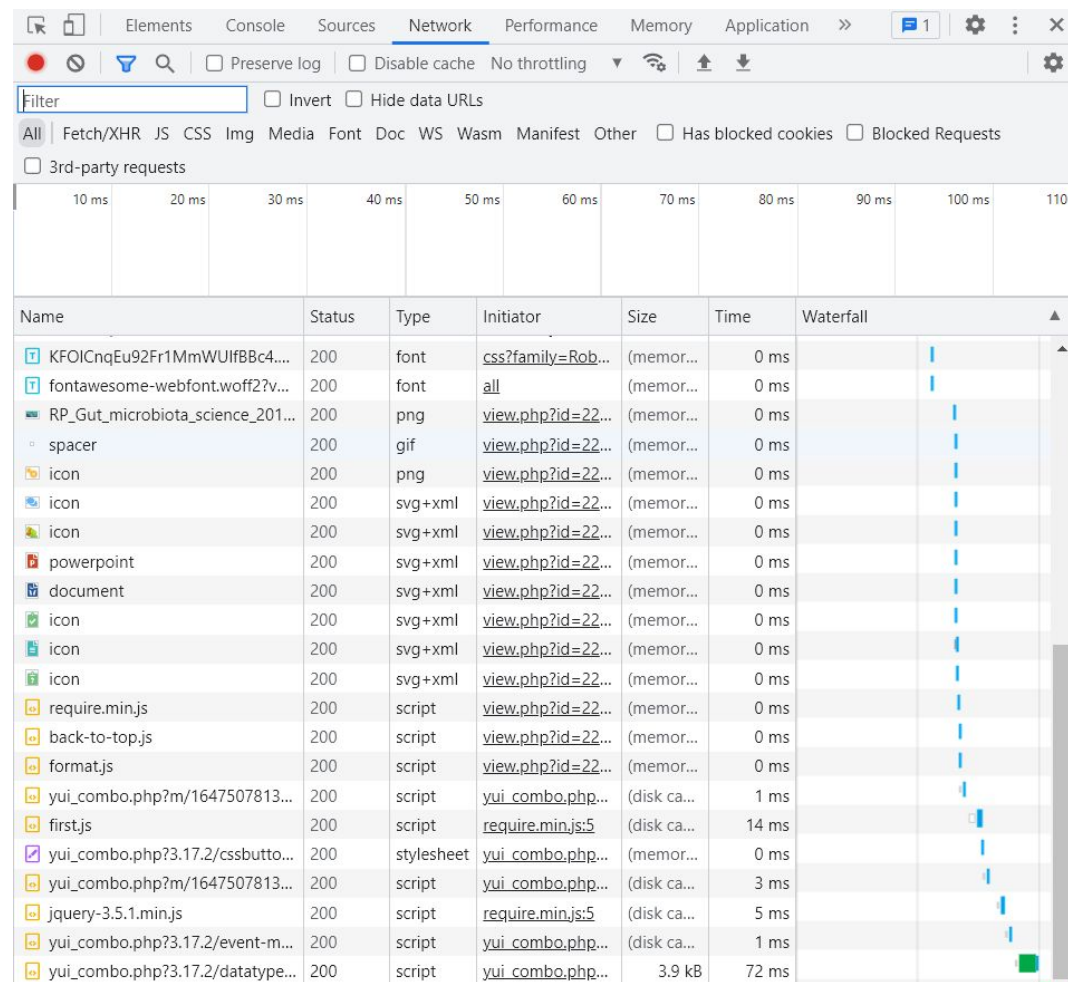


Network cilne



FRONTEND mājaslapas izstrāde

Var aplūkot mājaslapas saziņu ar serveriem un attiecīgos interneta resursus, to ielādes ātrumu un izmēru.



Performance cilne

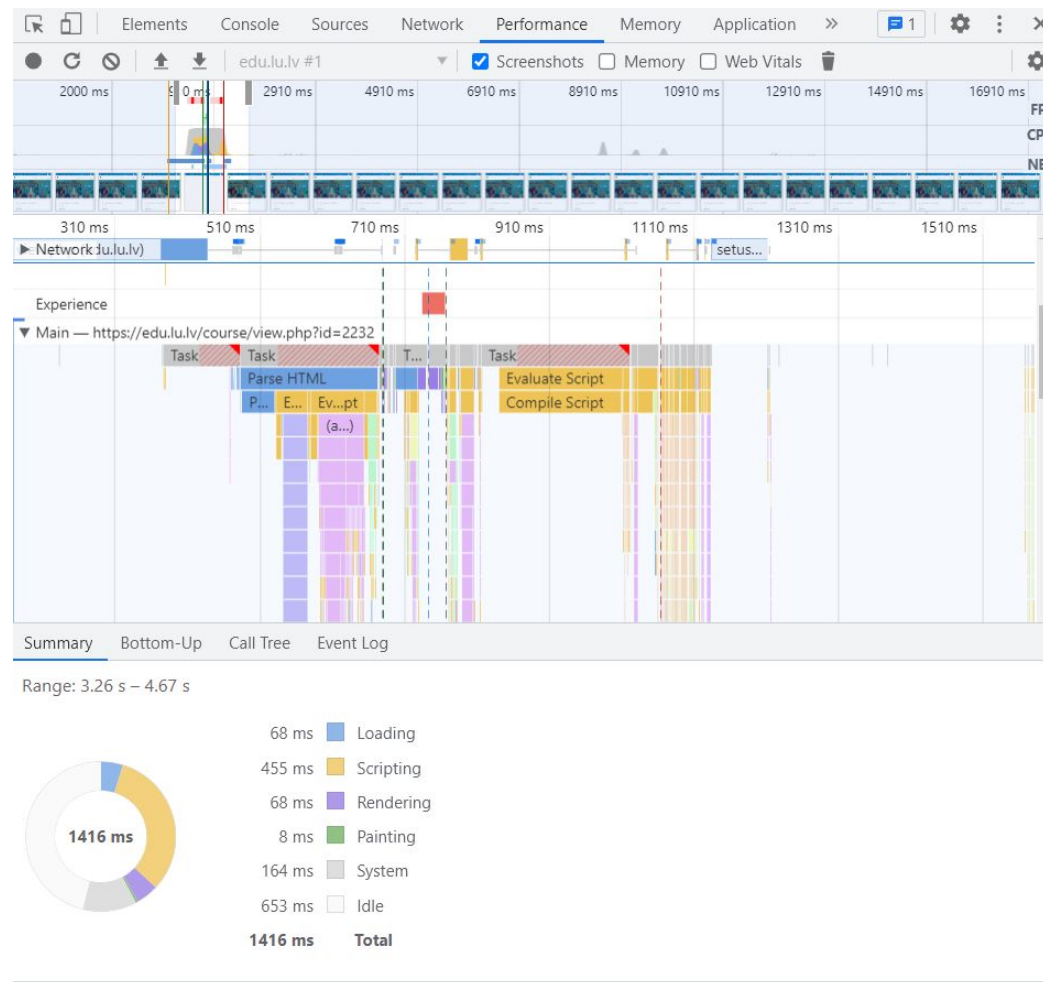


FRONTEND mājaslapas izstrāde

Iespējams ierakstīt un apskatīt mājaslapas veikspēju un visus mājaslapas ielādes posmus..

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.



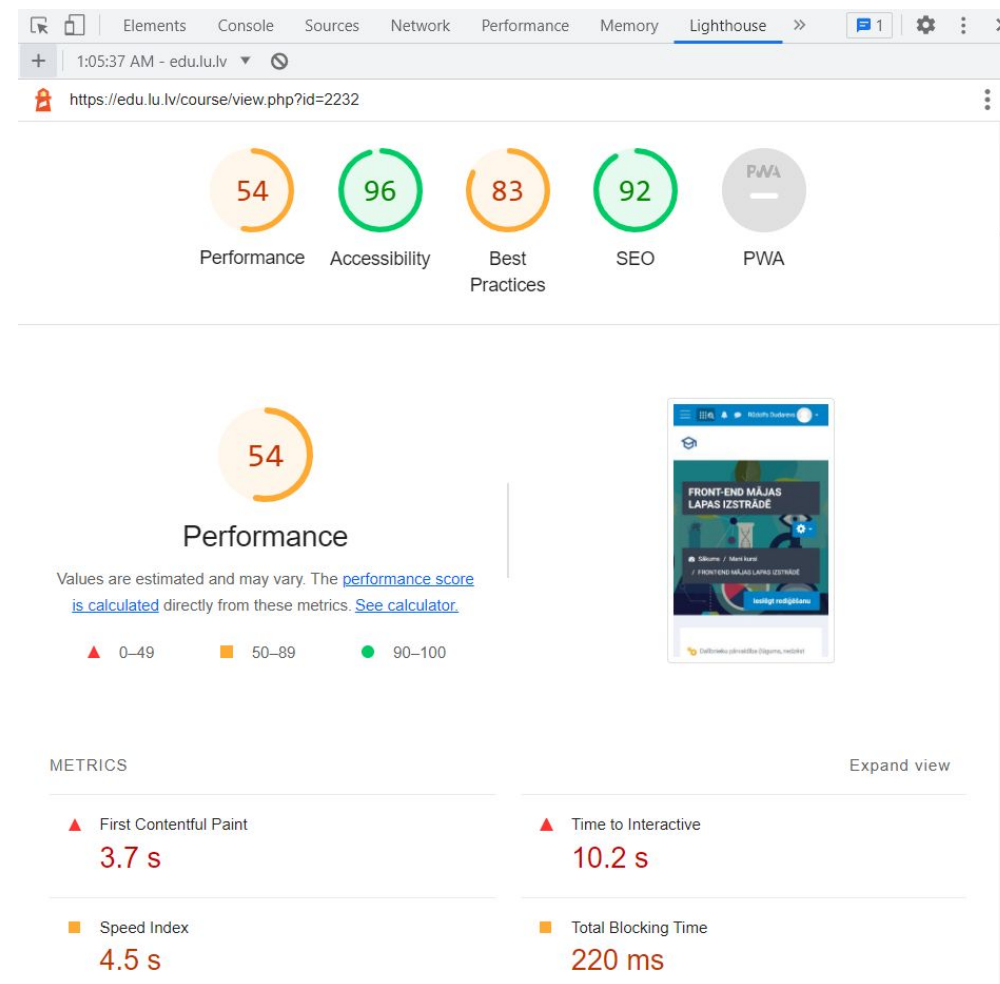
Lighthouse cilne

Šajā cilnē iespējams veikt auditu kopējās mājaslapas veikspējai, pielāgotību meklēšanas rīkiem(*crawlers, bots*), viedtālruniem un cilvēkiem ar invaliditāti.

Tiek piedāvāti arī konkrēti risinājumi kā uzlabot kādu no mājaslapas veikspējas aspektiem.



FRONTEND mājaslapas izstrāde





HTML birkas un atribūti

HTML elementi

HTML elementus pieraksta, izmantojot birkas (tags):

<nosaukums> saturs **</nosaukums>**

Birku nosaukumi tiek rakstīti ar mazo burtu.

Lai pārlūkprogramma varētu pareizi interpretēt HTML dokumenta saturu, nepieciešams norādīt kur sākas un kur beidzas katrs HTML elements. To mēs panākam ar sākuma un beigu birkām.

Sākuma birku no beigu birkas atšķir simboli ap tās nosaukumu:

Sākuma birka **<nosaukums>**

Beigu birka **</nosaukums>**

Pašnoslēdzoša birka **<nosaukums/>**

[Apskatīt piemēru W3Schools koda redaktorā](#)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <header>
      <ul>
        <li>Menu item 1</li>
        <li>Menu item 2</li>
        <li>Menu item 3</li>
      </ul>
    </header>
    <article>
      <h1>Heading</h1>
      <p>Paragraph of text.</p>
    </article>
  </body>
</html>
```

Sapārotas un nesapārotas birkas (paired and unpaired tags)

Sapārotas birkas

Lielākā daļa HTML birku ir pārotas (paired tags). Šīm birkām paredzēts ietvert sevī saturu - tekstu vai citus HTML elementus.

Lai pārlūkprogramma varētu izveidot DOM, ir nepieciešams norādīt kur birkas saturs sākas **<nosaukums>** un kur tās saturs beidzas **</nosaukums>**.

Ja tiks izmantota tikai sākuma birka bez beigu birkas - mājaslapa netiks pareizi attēlota.

Nesapārotas jeb pašnoslēdzošas birkas

Vairākas HTML birkas nav paredzētas, lai tās ietvertu sevī tekstu vai citus HTML elementus. Tās pieraksta tikai ar sākuma birku **<nosaukums>**.

Piemēram:

**
**

<hr>

<meta>

<input>

Līdz ar HTML5 ir pieņemts tās dēvēt arī par pašnoslēdzošām birkām (self-closing tags) un uzskatāmības dēļ arī pierakstīt kā **<nosaukums/>** jeb, piem., **<input/>**, bet tomēr tā nav obligāta prasība.

Redzamais saturs

Pārlūkprogrammā būs redzams saturs, kas ir ievietots iekš **<body>redzamais saturs</body>**.

```
<html>
```

```
<head>
```

```
<title>Page title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

Birku kategorijas

Ar katras birkas pielietojumu un vizuālu piemēru variet apskatīt [W3Schools](https://www.w3schools.com/html/html_elements.asp).

Visas birkas ko aplūkosim nav jāatcerās. Vairākas no tām neizbēgami iegaumēsiet, jo tās tiks bieži izmantotas. Tomēr ir svarīgi sākumā iepazīties ar birku kategorijām pēc to pielietojuma, lai gūtu priekšstatu par HTML iespējām.

- Basic HTML birkas
- Formatting HTML birkas
- Forms and Input birkas
- Frame birkas
- Images birkas
- Audio/Video birkas
- Link birkas
- List birkas
- Table birkas
- Style birkas
- Meta birkas
- Programming birkas
- Eksperimentālās / vēl oficiāli neapstiprinātās birkas

Basic birkas

<!DOCTYPE>

nosaka dokumenta tipu

<html>

nosaka, kur sākās HTML dokuments

<head>

satur metadatus un informāciju pārlūkprogrammai

<title>

nosaka dokumenta virsrakstu - nosaukums, ko redzam mājaslapas cilnē

<body>

nosaka, kur sākās lietotājam redzamais saturs

<!-- komentāra teksts -->

satur komentāru. Tekstu, ko pārlūkrpogramma neattēlos lietotājam, bet satur informāciju mājaslapu izstrādātājam.

<h1> līdz **<h6>**

izmanto mājaslapas satura virsrakstiem. Nosaukuma numerācija nosaka tā nozīmīgumu un arī noklusēto noformējumu - izmēru. Arī *crawlers*, *boti* veido savu interpretāciju un vērtējumu par mājaslapu pēc šiem virsrakstiem, kur h1 saturs ir svarīgāks par h6.

<p>

satur tekstu

**
**

ievieto vienas līnijas lielu atstarpi

<hr>

ievieto redzamu līniju starp html elementiem jeb atšķirīgām satura daļām. Bet vizuālu noformējumu veidošanai ieteicams lietot CSS, šī birka paredzēta semantiski atdalīt saturu nevis vizuāli to noformēt.

Formatting birkas (1/3)

Birkas tiek izmantotas teksta noformēšanai



FRONTEND mājaslapas izstrāde

<abbr>

nosaka akronīvu vai abreviāciju.

<address>

satur mājaslapas izstrādātāja adresi

saturošo tekstu padara uzsvērtu (bold)

<bdi>

nodala teksta daļu, kas atšķirīgās rakstības dēļ varētu traucēt pareizi attēlot tekstu pārlūkprogrammai.
Piem., retos gadījumos, kad izmanto arābu alfabētu, kas rakstās no labās uz kreiso pusi), iekš angļu valodas teksta.

<bdo>

pavērs pretēji saturošā teksta rakstības virzienu

<blockquote>

satur citātu

<cite>

atsauces uz sveša satura nosaukums

<code>

satur koda fragmentu kā tekstu

izsvītro tekstu

<dfn>

satur teksta definīciju, attēlots slīprakstā

izcelts teksts, attēlots slīprakstā

<i>

atšķirīga motīva teksts, attēlots slīprakstā

Formatting birkas (2/3)

Birkas tiek izmantotas teksta noformēšanai

<ins>

teksts ar apakšlīniju.

<kbd>

noformē tekstu kā taustiņu kombināciju.

<mark>

kā ar marķieri pārzīmēts teksts.

<meter>

izmantojot atribūtus, attēlo tekstu kā mērījuma stabiņu.

<pre>

neformatēts teksts pēc pārlūkprogrammas noklusētajiem uzstādījumiem.

<progress>

izmantojot atribūtus, attēlo tekstu kā progresu stabiņu.

<q>

citāta teksts.

<rp>

satur alternatīvu tekstu, ja pārlūkprogramma nespēj attēlot oriģinālo valodu. Izmantots āzijas hieroglifu valodu kontekstā.

<rt>

satur paskaidrojuma nozīmi vai fonētisko īpatnību, izmantots āzijas hieroglifu kontekstā.

<ruby>

satur anotāciju āzijas hieroglifu valodu kontekstā.

Formatting birkas (3/3)

Birkas tiek izmantotas teksta noformēšanai

<s>

nosvītrots teksts, norāda, ka vairs nav pareizs.

<samp>

koda piemērs kā teksts.

<small>

pārvērs tekstu mazākā izmērā.

saturošo tekstu padara uzsvērtu (bold), norāda, ka teksts ir svarīgs.

<sub>

pārvērš tekstu *subscript*.

<sup>

pārvērš tekstu *superscript*.

<template>

saturs ko pārlūkprogramma paslēpj tiklīdz mājaslapa ir ielādēta. Saturu var parādīt ar JS.

<time>

nosaka tekstu kā laika vienību.

<u>

pavītrots teksts, nosaka, ka teksts ir stilistiski atšķirīgs no pārējā teksta.

<var>

teksta matemātiskam pierakstam, norāda uz mainīgo.

<wbr>

norāda pārlūkrpogrammai, kur ieteicams lauzt tekstu jaunā līnijā, ja tas nepieciešams ekrāna izmēra dēļ.

Forms and inputs birkas

<form></form>

nosaka ievades formu lietotāja datiem, apkopo visus saturošos ievades lauku elementus vieglākais datu apstrādei un nosūtīšanai serverim.

<input/>

teksta ievades lauks (viena teksta rindiņa)

<textarea>

teksta ievades lauks (vairākas teksta rindiņas)

<button/>

izveido podu ar saturošo tekstu, funkcionalitāti var piešķirt ar atribūtiem un/vai JS.

<select>

dropdown izvēlne

<optgroup>

sagrupē vairākas izvēles no *dropdown* izvēlnes

<option>

dropdown izvēlnes saraksta elements

<label>

virsraksts ievades laukiem

<fieldset>

grupē vairākus ievades laukus formā

<legend>

nosaukums, atšifrējums *<fieldset>* birkas saturam

<datalist>

satur teksta ievades lauka ieteiktās vērtības. Vienkārša auto-fill versija.

<output>

izmantojot atribūtus, var veikt vienkāršās matemātiskas darbības ar citiem ievades laukiem, attēlo rezultātu

Frame birkas

<iframe>

satur citu HTML dokumentu - mājaslapa iekš mājaslapas.
Ja tiek izvietots ārējs HTML dokuments (no cita *host*),
darbības ar iFrame saturu ir ļoti ierobežotas.

Images birkas

satur attēlu. Attēla resursu norāda atribūtos.

<map>

sasaista attēlu ar interaktīviem elementiem uz attēla.

<area>

interaktīvie elementi iekš <map>

<canvas>

grafisko elementu un sarežģītu animāciju attēlošana, izmantojot skriptus, visbiežāk JavaScript.

<figcaption>

<figure> grupas paskaidrojošais teksts. Piemēram attēls un oriģinālā autors vai paskaidrojošais teksts zem attēla.

<figure>

grupē attēlu elementu ar paskaidrojošu tekstu.

<picture>

grupē vairākus attēlu elementus.

<svg>

satur SVG formāta vektora attēlu. Parasti izmanto ikonām.

Audio/Video Birkas



FRONTEND mājaslapas izstrāde

<audio>

audio saturs

<source>

saraksts ar atšķirīgiem audio, video un attēla interneta resursiem (<video>, <audio> and <picture>). Piem., dažādas izšķirtspējas attēli un video atšķirīgām ierīcēm.

<track>

satur teksta elementus priekš <video> un <audio> birkām

<video>

video saturs

Link birkas

<a>

satur saiti uz interneta resursu - nu tajā pašā mājaslapā, kurā atrodās saite vai uz ārēju interneta resursu.

<link>

nosaka HTML dokumenta saikni ar ārēju interneta resursu. Visbiežāk izmanto, lai ievietotu HTML dokumentā CSS failus vai fontus.

<nav>

satur navigācijas HTML elementus, visbiežāk <a> saites, bet var izmantot arī semantisk iar citiem HTML elementiem, lai grupētu navigāciju.

List birkas

satur nenumurētus saraksta elementus

satur numurētus saraksta elementus

saraksta elements

<dl>

definīciju saraksts un paskaidrojumiem

<dt>

definīciju saraksta termins

<dd>

definīciju saraksta termina paskaidrojums

Tables birkas

<table>

satur tabulas elementus

<caption>

tabulas nosaukums

<th>

tabulas galvene

<tr>

tabulas rinda

<td>

tabulas lauks

<thead>

grupē tabulas galvenes

<tbody>

grupē tabulas saturu

<tfoot>

grupē tabulas *footer*

<col>

grupē un formatē tabulas aturu
kolonā<colgroup> element

<colgroup>

grupē tabulas kolonas

Styles un semantics birkas

<style>

satur HTML vizuāla noformējuma informāciju. Vai nu CSS vai saites uz CSS failiem.

<div>

grupē HTML elementus blokos.

grupē teksta daļu un citus HTML elementus semantiski saistītā grupā.

<header>

satur mājaslapas galveni.

<footer>

satur mājaslapas kājēni.

<main>

HTML dokumenta satura galvenā daļa.

<section>

grupē HTML elementus mājaslapas sekcijās.

<article>

satur rakstu.

<aside>

sānos izvietots mājaslapas saturs.

<details>

atverama un aizverama izvēlne, kas satur paplašinātu aprakstu.

<summary>

<details> birkas satura virsraksts

<dialog>

dialoga logs

<data>

satur informāciju, ar ko var mijiedarboties programmēšanas valodas.

Meta info birkas

<head>

satur informāciju par HTML dokumentu pārlūkprogrammai.

<meta>

satur HTML dokumenta metadatus.

<base>

nosaka URL vai internetu resursu atrašanās vietu pret kuru tiks atrisinātas atsauces uz lokāliem interneta resursiem (interneta resursi no mājaslapas host).

Programming birkas

<script>

satur skriptu, kas izpildīsies lietotāja pārlūkprogrammā

<noscript>

satur paziņojumu lietotājiem, kuru pārlūkprogrammā ir izslēgti vai kura neatbalsta JS

<embed>

satur ārēju ne-HTML saturu

<object>

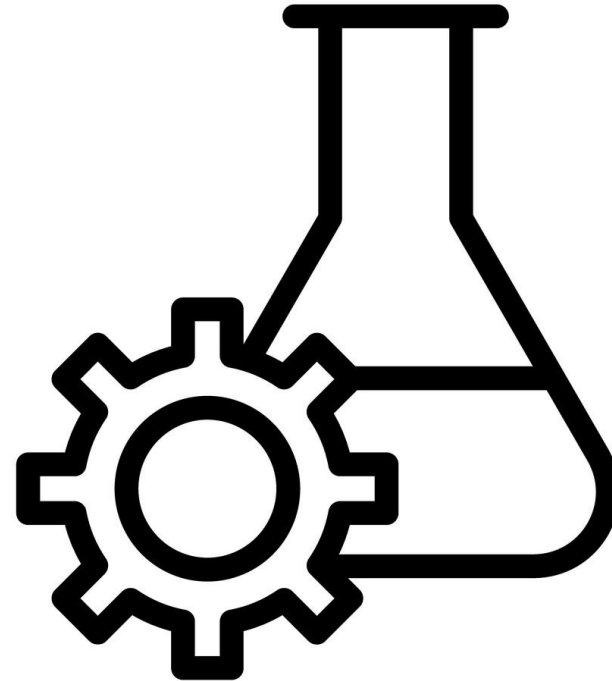
satur iegultu objektu. Objektus apskatīsim vēlāk kursā.

<param>

nosaka <object> satura parametrus.

Eksperimentālās birkas

Pārlūkprogrammas mēdz ieviest eksperimentālas HTML birkas. Tās praktiski nav ieteicams izmantot mazā atbalsta dēļ. Bija aktuālākas vecākās HTML versijās, bet HTML 5 ir kļuvis ļoti labi standartizēts un plaši pielietots - pagaidām nav radusies nepieciešamība izstrādāt eksperimentālas birkas HTML 5.



HTML elementu atribūti

Jebkurai HTML birkai var pievienot atribūtu. Tie mums ļauj pievienot informāciju HTML elementiem.

Šī informācija var būt norādes uz interneta resursiem, HTML elementa īpašības, paskaidrojošs teksts, saturs ar ko var mijiedarboties CSS un JS.

Tos vienmēr norāda sākuma birkā.

Tos pieraksta:

`nosaukums="saturs"`

Globālie atribūti

Var tikt izmantoti jebkuram HTML elementam.

Brikām specifiskie atribūti

Var tikt izmantoti tikai atbilstošiem HTML elementiem.

Globālie atribūti

accesskey

norāda īsceļa taustiņu ar ko navigēt uz attiecīgo HTML elementu

class

klases nosaukums HTML elementam. Ļauj JS un CSS identificēt elementu pēc to piederības klasei.

contenteditable

nosaka vai birkas saturs ir rediģējams.

data-*

satur datus ar ko var mijiedarboties JS.

dir

nosaka elementa satura izkārtojuma virzienu.

draggable

nosaka vai elements var tik pārvietots ar datora peli.

hidden

paslēpj elementu, apzīmē lietotājam dotajā brīdī nesvarīgus elementus.

id

unikāls identifikators elementam ko var izmantot JS, CSS vai navigējot uz to ar saiti.

lang

nosaka elementa satura valodu.

spellcheck

nosaka vai pārlūkrpogrammais veikt elementa teksta satura pareizrakstības pārbaudi.

style

nosaka augstākās prioritātes vizuālos noformējumu elementam. Satur CSS.

tabindex

nosaka pozīciju navigācijas rindā, kad tiek izmantots TAB taustiņš

title

satur elementa virsrakstu, papildus informāciju - apskatām tikai, kad peles kursora atrodas uz dotā elementa.

translate

nosaka vai elementa teksta sturu pārlūkprogrammais tulkot.

Elementiem specifiskie atribūti

Tā kā HTML birku skaits ir ļoti apjomīgs un katrai no tām var būt pieejami arī vairāki unikāli atribūti, tos prezentācijā neuzskaitīsim.

Izmantojot HTML birkas uzdevuma veikšanai, vēlams izpetīt šīs birkas dokumentāciju un izprast tās specifiku kā arī pieejamos atribūtus.

Izmantot jau lekcijā minētos resursus. Vai izmanto meklēšanas rīkus un apskatīt to paplašināti.

<https://www.w3schools.com>

<https://developer.mozilla.org>

<https://caniuse.com>

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