

# Veebiarendus

## Front-End arendus

Martti Raavel

[martti.raavel@tlu.ee](mailto:martti.raavel@tlu.ee)

# Teemad

- Mida me tegema hakkame?
- HTML
- CSS

# Mida me tegema hakkame?

- Front-End arendus (kasutajaliides)
- Ei loo disaini, vaid teeme disaini elluviimiseks vajalikud tehnilised lahendused
- HTML, CSS, JavaScript
- CSS raamistikud (Bootstrap, Material Design)
- Hiljem ka React

# HTML - 1

HTML on lühend, mis märgenditab Hyper Text Markup Language ja tähendab tõlgituna Hüperteksti Märgendikeel.

Märgendikeel tähendab, et erinevalt programmeerimiskeele kasutamisest arvuti juhendamiseks, kasutatakse märgendeid erinevat tüüpi sisu määratlemiseks.

HTML on standardne märgendikeel veebilehtede loomiseks ja see kirjeldab veebilehe struktuuri.

## HTML - 2

HTML koosneb elementide seeriast, mida kasutatakse sisu teatud viisil kuvamiseks või käitumiseks. Sisu ümbritsevad märgendid võivad muuta sõna või pildi hüperlingiks kuskile mujale lehele, võivad sõnad kursiivis kuvada, muuta fondi suuremaks või väiksemaks jne.

## HTML - 3

HTML-is ei öelda arvutile, mida teha, vaid öeldakse, mis midagi on. Näiteks võib öelda "see on lõik", "see on pealkiri", "see on link" jne. Veebilehitseja teab siis, kuidas kirjutatud sisu kuvada.

# HTML märgendid - 1

HTML märgendid on elemendinimed, mis on ümbritsetud nurksulgudega `<>` – HTML-is kasutatakse märgendit HTML elementide loomiseks, mis on HTML-lehtede ehituskivid. HTML märgendid on märksõnad nurksulgude vahel:

```
<tagname></tagname>
```

## HTML märgendid - 2

HTML märgendid tulevad tavaliselt paarides nagu `<p>` ja `</p>`. Esimene märgend paarist on algustmärgend, teine märgend on lõpumärgend. Lõpumärgendit kirjutatakse nagu algusmärgendit, kuid enne märgendi nime lisatakse kaldkriips. Mõnedel HTML märgendil ei ole lõpumärgendit, nagu `<br>` märgend (mis näitab reavahetust).



## HTML märgendid - 3

- `<h1>` - esimese taseme pealkiri;
- `<p>` - lõik, paragrahv;
- `<div>` - kasutatakse jaotuse või sektsiooni loomiseks HTML dokumendis;
- `<a>` - link, hüperlink;
- `<img>` - pildi kuvamise märgend;
- `<table>` - tabeli loomise märgend;

## HTML märgendid - 4

- `<form>` - sisestusvormi loomise märgend;
- `<input>` - sisestusvormi välja loomise märgend;
- `<button>` - nupp;
- `<ul>` - järjestamata loend (täpploend);
- `<ol>` - järjestatud loend (nummerdatud loend);
- `<li>` - loendi element;
- jne

# HTML elemendid - 1

HTML element on defineeritud algustähise, mõne sisu ja lõputähisega:

```
<tagname>Sisu läheb siia...</tagname>
```

Tähiseid kasutatakse HTML elemendi alguse ja lõpu märgenditamiseks. Elemendi sisu on kõik algus- ja lõpumärgenid vahel olev, kaasa arvatud algus- ja lõpumärgendid.

## HTML elemendid - 2

Näiteks järgmine HTML kood defineerib esimese taseme pealkirja ja lõigu elemendi:

```
<h1>See on pealkiri</h1>  
<p>See on lõik.</p>
```

`<h1>` märgend defineerib esimese taseme pealkirja. `<p>` märgendi defineerib lõigu.

## Pesastatud HTML elemendid (nested elements)

HTML elemendid võivad olla pesastatud (elemendid võivad sisaldada elemente). Enamik HTML elemente võib olla pesastatud (välja arvatud mõned HTML elemendid, nagu `<br>` märgendid).

Näiteks järgmine HTML kood defineerib esimese taseme pealkirja ja lõigu elemendi ning lõigu element sisaldab lingi elementi:

```
<h1>See on pealkiri</h1>
<p>See on lõik.
  <a href="https://www.google.com/">Google</a>
</p>
```

## Pesatud HTML elemendid (nested elements) - 2

```
<ul>  
  <li>Üksus 1</li>  
  <li>Üksus 2</li>  
</ul>
```

```
<ol>  
  <li>Üksus 1</li>  
  <li>Üksus 2</li>  
</ol>
```

# HTML atribuudid - 1

HTML atribuudid on erilised sõnad, mida kasutatakse algusmärgendites elemendi käitumise juhtimiseks ja HTML elementidele lisainformatsiooni andmiseks.

HTML atribuudid on alati määratud algusmärgendites.

HTML atribuudid on nimetatud/väärtustatud paaridena nagu: `nimi="väärtus"`.

## HTML atribuudid - 2

Järgmine HTML kood defineerib esimese taseme pealkirja koos id atribuudiga:

```
<h1 id="heading">See on pealkiri</h1>
```

`id` atribuut määrab HTML elemendile unikaalse id. `id` atribuudi väärtus peab olema HTML dokumendis unikaalne. `id` atribuuti kasutatakse elemendi tuvastamiseks lingi abil (kasutades fragmendi identifikaatorit), skriptimisel või stiilimisel (CSS-i abil).



## HTML atribuudid - 3

On mõned märgendid, millel on enamasti ka mõned atribuudid. Näiteks `<a>` märgendit kasutatakse hüperlingi loomiseks ja sellel on tavaliselt `href` atribuut, mis määrab lingi sihtkoha URL-i.

`<img>` märgendit kasutatakse pildi kuvamiseks ja sellel on tavaliselt `src` atribuut, mis määrab kuvatava pildi URL-i. `img` märgendit jaoks kasutatakse ka tavaliselt `alt` atribuuti, mis määrab pildi alternatiivse teksti, kui pilti ei saa kuvada.

```
<a href="https://www.google.com/">Google</a>  

```

# HTML dokumendi struktuur

HTML dokument on fail, mis sisaldab HTML koodi. HTML dokument koosneb järgmistest osadest:

- HTML dokumendi deklaratsioon;
- HTML dokumendi juurelement;
- HTML dokumendi päis;
- HTML dokumendi keha.

# HTML dokumendi deklaratsioon

HTML dokumendi deklaratsioon defineerib HTML dokumendi tüübi ja versiooni. Deklaratsioon kirjutatakse alati HTML dokumendi algusesse.

```
<!DOCTYPE html>
```

# HTML dokumendi juurelement

HTML dokumendi juurelement on `<html>` element. See element on HTML dokumendi juur ja sisaldab kogu HTML dokumendi sisu, välja arvatud HTML dokumendi deklaratsioonitüüp.

```
<html>  
...  
</html>
```

# HTML dokumendi päis

HTML dokumendi päis on `<head>` element. Päises on teave HTML dokumendi kohta, nagu pealkiri, skriptid, stiilid, metaandmed jne.

HTML dokumendi `head` kirjutatakse nii:

```
<head>  
  <title>Lehe pealkiri</title>  
</head>
```

# HTML dokumendi keha

HTML dokumendi keha on `<body>` element. See element sisaldab kogu HTML dokumendi tegelikku sisu, mida kuvatakse veebilehitsejas.

```
<body>  
  <h1>See on pealkiri</h1>  
  <p>See on lõik.</p>  
</body>
```

# HTML dokument kokku

```
<!DOCTYPE html>
<html>
  <head>
    <title>Lehe pealkiri</title>
  </head>
  <body>
    <h1>See on pealkiri</h1>
    <p>See on lõik.</p>
  </body>
</html>
```

Oluline on märkida, et kõik HTML elemendid on HTML dokumendi juurelemendi järeltulijad.

# HTML dokumendi metaandmed

HTML dokumendi metaandmed on andmed HTML dokumendi kohta. HTML dokumendi metaandmeid kasutavad veebilehitsejad (kuidas sisu kuvada või lehte uuesti laadida), otsingumootorid (märksõnad) jne. HTML dokumendi metaandmeid määratakse HTML dokumendi `head`-is, kasutades `meta` märgendit ja atribuute. Näiteks:

```
<head>
  <title>Lehe pealkiri</title>
  <meta charset="UTF-8">
  <meta name="description" content="Tasuta veebiõpetused">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="John Doe">
</head>
```



## HTML dokumendi käivitamine (vaatamine)

HTML dokumenti saab vaadata veebilehitsejas. Selleks tuleb HTML kood salvestada faili, mille laiend on `.html` või `.htm` ja avada see veebilehitsejas.

# Harjutused

1. Loo HTML dokument, mis sisaldab pealkirja ja lõiku.
2. Lisa pealkirja ja lõigu vahele pilt.
3. Lisa pildi alla link, mis viib Google'i kodulehele.
4. Lisa lõigu alla loend, mis sisaldab vähemalt 3 elementi.
5. Lisa loendi alla tabel, mis sisaldab vähemalt 3 rida ja 3 veergu.

# CSS - 1

CSS on lühend, mis tähistab Cascading Style Sheets. CSS on stiililehtede keel, mis kirjeldab, kuidas HTML elemendid peaksid olema kuvatud.

CSS on keel, mida kasutatakse HTML elementide stiilimiseks. Seda kasutatakse veebilehtede värvide, fontide, paigutuste ja muu lisamiseks. CSS on koos HTML-i ja JavaScriptiga üks veebi põhitehnoloogiaid.

## CSS - 2

CSS on loodud esitluse ja sisu eraldamiseks, sealhulgas paigutuse, värvide ja fontidega. See eraldamine parandab sisu ligipääsetavust, pakub rohkem paindlikkust ja kontrolli esitlusomaduste määramisel, võimaldab mitmel veebilehel jagada vormindust, määrates vastava CSS-i eraldi `.css` failis, ja vähendab struktuurse sisu keerukust ja kordust.

## CSS - 3

Kui HTML kirjeldab veebilehe struktuuri, siis CSS kirjeldab, kuidas see struktuur peaks välja nägema.

Näiteks kui soovid oma veebilehel teksti värvi muuta, saad seda teha CSS-i abil. Saad kasutada CSS-i ka fondi suuruse, fondipere, taustavärvi, piirivärvi, piiri laiuse, piiri stiili jne muutmiseks.

# Kuidas rakendada CSS-i HTML-ile

CSS-i rakendamiseks HTML-ile on kolm viisi:

- inline CSS;
- sisemine CSS;
- väline CSS;

## Inline CSS

Inline CSS on CSS, mis on määratud otse HTML elemendi atribuudis. Inline CSS-i kasutatakse siis, kui soovite rakendada CSS-i ainult ühele HTML elemendile. Näiteks:

```
<p style="color: red;">See on punane lõik.</p>
```

# Sisemine CSS

Sisemine CSS on CSS, mis on määratud HTML dokumendi `<head>` elemendis `<style>` märgendis. Sisemist CSS-i kasutatakse siis, kui soovite rakendada CSS-i kogu HTML dokumendile. Näiteks:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p {
        color: red;
      }
    </style>
  </head>
  <body>
    <p>See on punane lõik.</p>
  </body>
</html>
```



# Väline CSS

Väline CSS on CSS, mis on määratud eraldi CSS-failis. Välist CSS-i kasutatakse siis, kui soovite rakendada CSS-i mitmele HTML dokumendile. Näiteks:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <p>See on punane lõik.</p>
  </body>
</html>
```

style.css fail:

```
p {
  color: red;
}
```

# CSS Selektorid - 1

Kui soovime HTML-elemendile mingit stiili rakendada, peame selle esmalt valima, et määrata, millist elementi soovime stiilida. HTML-elementi saab valida CSS-selektori abil.

CSS-selektoreid kasutatakse HTML-elementide valimiseks, mida soovite stiilida. Selektorid muudavad CSS-i võimsamaks, võimaldades teil sihtida konkreetseid elemente oma veebilehel.

## CSS Selektorid - 2

Selektorite abil saame valida HTML-elemente näiteks:

- märgendi nime järgi;
- klassi nime järgi;
- id järgi;
- atribuudi järgi;
- pseudo-klassi järgi;
- jne.

## CSS Selektorid - 3

Kui kirjutame CSS-i, peame määrama kaks asja:

- selektori
- deklaratsiooniploki.

Selektorit kasutatakse HTML-elementide valimiseks, mida soovime stiilida.

Deklaratsiooniplokk on kasutatud CSS-reeglite määramiseks, mis rakendatakse valitud HTML-elementidele.

# CSS Deklaratsiooniplokk

Deklaratsiooniplokk on ümbritsetud loogeliste sulgudega `{}`. Loogeliste sulgude sees saame määratleda ühe või mitu CSS-reeglit. Iga CSS-reegel koosneb omadusest ja väärtusest. Omadus järgneb koolonile `:` ja väärtus järgneb semikoolonile `;`.

```
selector {  
  property: value;  
}
```

# Märgendi selektorid

Märgendi selektoreid kasutatakse HTML-elementide valimiseks nende märgendi nime järgi. Selektorid ei ole tõstutundlikud, mis tähendab, et `p` ja `P` on samad selektorid. Näiteks kui soovime valida kõik `<p>` elemendid meie veebilehel, saame kasutada järgmist selektorit:

```
p {  
  color: red;  
}
```

# Klassi selektorid

Kui soovime valida kõik elemendid klassiga `class="my-class"`, saame kasutada järgmist selektorit:

```
.my-class {  
  /* CSS-reeglid */  
}
```

Klassi selektorid algavad punktiga `.`, millele järgneb klassi nimi. Selles näites on klassi nimi `my-class`.

# ID selektorid

Kui soovime valida kõik elemendid ID-ga `id="my-id"`, saame kasutada järgmist selektorit:

```
#my-id {  
  /* CSS-reeglid */  
}
```

ID selektorid algavad räsikärgiga `#`, millele järgneb ID nimi. Selles näites on ID nimi `my-id`.



# CSS omadused

CSS omadusi kasutatakse valitud HTML-elementide stiilimiseks. Igal CSS omadusel on nimi ja väärtus. Nimele järgneb koolon `:` ja väärtusele järgneb semikoolon `;`. Näiteks kui soovime muuta oma veebilehe teksti värvi, saame kasutada järgmist CSS-reeglit:

```
p {  
  color: red;  
}
```

Selles näites on omadus `color` ja väärtus `red`. Omadust järgneb koolon `:` ja väärtust järgneb semikoolon `;`.

## CSS omadused - 2

On palju erinevaid CSS omadusi. Mõned kõige levinumad CSS omadused on:

- `color` - määrab teksti värvi;
- `font-size` - määrab teksti fondi suuruse;
- `font-family` - määrab teksti fondipere;
- `background-color` - määrab elemendi taustavärvi;
- `border-color` - määrab elemendi piirivärvi;
- `border-width` - määrab elemendi piiri laiuse;
- `border-style` - määrab elemendi piiri stiili;
- `border` - määrab elemendi raamjoone;

## CSS omadused - 3

- `width` - määrab elemendi laiuse;
- `height` - määrab elemendi kõrguse;
- `margin` - määrab elemendi veerise;
- `padding` - määrab elemendi polsterduse;
- `text-align` - määrab teksti horisontaalse joonduse;
- `vertical-align` - määrab teksti vertikaalse joonduse;
- `display` - määrab elemendi kuvamiskäitumise;
- `position` - määrab elemendi positsiooni;
- jne

# CSS väärtused

CSS väärtusi kasutatakse CSS omaduse väärtuse määramiseks. Näiteks kui soovime muuta oma veebilehe teksti värvi, saame kasutada järgmist CSS-reeglit:

```
p {  
  color: blue;  
  font-size: 40px;  
}
```

Selles näites on CSS-reegel kolme omadusega, mis on rakendatud kõigile veebilehe lõikudele: `color` ja `font-size`. `color` omaduse väärtus on `blue`. `font-size` omaduse väärtus on `40px`.

## CSS väärtused - mitu väärtust

Mõned CSS omadused võivad nõuda või võimaldada mitut väärtust.

```
p {  
  border: 1px solid black;  
}
```

Selles näites on `border` omadusel kolm väärtust: `1px`, `solid` ja `black`. Esimene väärtus ( `1px` ) määrab piiri laiuse. Teine väärtus ( `solid` ) määrab piiri stiili. Viimane väärtus ( `black` ) määrab piiri värvi.

# Ühikud

CSS ühikuid kasutatakse elemendi suuruse määramiseks. On kaks tüüpi CSS ühikuid:

- absoluutsed ühikud
- suhtelised ühikud .

# Absoluutsed ühikud - 1

Absoluutsed ühikud on fikseeritud ühikud. Need ei ole suhtelised millegi suhtes.

Näiteks kui soovime määrata elemendi laiuse absoluutse ühiku abil, saame kasutada järgmist CSS-reeglit:

```
p {  
  width: 100px;  
}
```

Selles näites on elemendi laius `100px`. Laius ei ole suhteline millegi suhtes. See on fikseeritud `100px`.

# Absoluutsed ühikud - 2

Mõned kõige levinumad absoluutsed ühikud on:

- `cm` - sentimeetrid -  $1\text{cm} = 37,8\text{px}$ ;
- `mm` - millimeetrid -  $1\text{mm} = 3,78\text{px}$ ;
- `Q` - veerand-millimeetrid -  $1\text{Q} = 0,95\text{px}$ ;
- `in` - tollid -  $1\text{in} = 96\text{px}$ ;
- `pc` - pikad -  $1\text{pc} = 16\text{px}$ ;
- `pt` - punktid -  $1\text{pt} = 1,33\text{px}$ ;
- `px` - pikslid -  $1\text{px} = 1/96$  tolli;

Tavaliselt kasutatakse absoluutse ühikuna `px`.



# Suhtelised ühikud - 1

Suhtelised ühikud on suhtelised millegi muu suhtes (vanemaelemendi, juurelemendi jne suhtes). Näiteks kui soovime määrata elemendi laiuse suhtelise ühiku abil, saame kasutada järgmist CSS-reeglit:

```
p {  
  width: 100%;  
}
```

Selles näites on elemendi laius `100%`. Laius on suhteline vanemaelemendi laiusele. Kui vanemaelemendi laius on `100px`, on elemendi laius `100px`. Kui vanemaelemendi laius on `200px`, on elemendi laius `200px`. Kui vanemaelemendi laius on `300px`, on elemendi laius `300px`. Ja nii edasi.

## Suhtelised ühikud - 2

Mõned kõige levinumad suhtelised ühikud on:

- % - protsent;
- em - vanemaelemendi fondi suurus;
- rem - juurelemendi fondi suurus;
- vw - 1% vaateakna laiupest;
- vh - 1% vaateakna kõrgusest;
- vmin - 1% vaateakna laiupest või kõrgusest, olenevalt sellest, kumb on väiksem;
- vmax - 1% vaateakna laiupest või kõrgusest, olenevalt sellest, kumb on suurem;
- jne.

## Väärtused, mis ei nõua ühikuid - 1

Mõned väärtused ei nõua ühikuid. Näiteks kui soovime määrata elemendi läbipaistvuse, saame kasutada järgmist CSS-reeglit:

```
p {  
  opacity: 0.5;  
}
```

Selles näites on elemendi läbipaistvus `0.5`. Läbipaistvus ei ole suhteline millegi suhtes. See on fikseeritud `0.5`.

## Väärtused, mis ei nõua ühikuid - 2

Mõned kõige levinumad väärtused, mis ei nõua ühikuid, on:

- `opacity` - määrab elemendi läbipaistvuse;
- `z-index` - määrab elemendi virna järjekorra;
- `order` - määrab elemendi järjekorra;
- jne.

# CSS värvid

Värve kasutatakse elemendi värvi määramiseks. On palju erinevaid viise värvide määramiseks CSS-is. Mõned kõige levinumad viisid on:

- `color name` - näiteks `red`, `green`, `blue` jne;
- `rgb` - näiteks `rgb(255, 0, 0)`, `rgb(0, 255, 0)`, `rgb(0, 0, 255)` jne;
- `hexadecimal` - näiteks `#ff0000`, `#00ff00`, `#0000ff` jne;
- `rgba` - näiteks `rgba(255, 0, 0, 0.5)`, `rgba(0, 255, 0, 0.5)`, `rgba(0, 0, 255, 0.5)` jne;

# CSS värvid - nimed

CSS-is on palju värvinimesid. Mõned kõige levinumad värvinimed on:

- `red` - punane värv;
- `green` - roheline värv;
- `blue` - sinine värv;
- `yellow` - kollane värv;
- `orange` - oranž värv;
- `purple` - lilla värv;
- `pink` - roosa värv;
- `black` - must värv;
- `white` - valge värv;
- jne

# Kodune ülesanne

- Loe läbi loengus käsitletud teemad