

Veebiarendus

Front-End arendus

Martti Raavel

martti.raavel@tlu.ee

Teemad

- Meenutame eelmist loengut
- CSS Position
- CSS Frameworks
- Bootstrap
- Andmete pärimine välisest allikast
 - Axios

Meenutame eelmist loengut

CSS Position

CSS-i position omadus võimaldab arendajatel määrata elemendi paigutuse meetodi. Selle omaduse abil saate määrata, kuidas element paigutatakse suhtes oma tavapärase vooga või teiste elementidega.

CSS Position väärtused

Peamised CSS-i position omaduse väärtused on:

- static
- relative
- absolute
- fixed
- sticky

CSS Position: static

Static väärtus on `position` omaduse vaikeseade, mis jätab elemendi tavapärasesse voogu. `top`, `right`, `bottom`, ja `left` omadused ei mõjuta static elemendi asukohta.

```
div.static {  
  position: static;  
  border: 3px solid #73AD21;  
}
```

CSS Position: relative

Relative väärtus paigutab elemendi suhteliselt tema tavapärasest asukohast. See võimaldab elemendil liikuda ilma teisi elemente mõjutamata.

```
div.relative {  
  position: relative;  
  left: 30px;  
  border: 3px solid #73AD21;  
}
```

CSS Position: absolute

Absolute väärtus eemaldab elemendi dokumendi elementide tavapärasest voost ja paigutab selle suhtes lähima positsioneeritud vanema elemendiga. Kui sellist esivanemat pole, siis paigutatakse element suhtes HTML body elemendiga.

```
div.absolute {  
  position: absolute;  
  right: 0;  
  border: 3px solid #73AD21;  
}
```


CSS Position: fixed

Fixed väärtus paigutab elemendi nähtava ala suhtes ja see jääb paigale isegi kerimisel.

```
div.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 200px;  
  border: 3px solid #73AD21;  
}
```

CSS Position: sticky

sticky: väärtus on kombinatsioon `relative` ja `fixed` paigutusest. Element on `relative` kuni teatud punktini, kus see muutub `fixed` elemendiks. Seda kasutatakse näiteks navigatsioonimenüüde jaoks.

```
div.sticky {  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 3px solid #73AD21;  
}
```

CSS Frameworks

CSS-i raamistikud on eelnevalt kirjutatud CSS-i koodikogud, mis pakuvad mugavaid stiilide, paigutuste ja komponentide malle. Need on mõeldud veebiarenduse kiirendamiseks, pakkudes järjepidevat stiili ja toetust mitmesugustele disainilahendustele.

Populaarsed CSS-i raamistikud

- Bootstrap
- Foundation
- Tailwind CSS
- Bulma
- jne

CSS-i raamistike eelised

- **Kiirus:** Valmiskomponendid ja klassid vähendavad aega, mis kulub veebilehtede visuaalse osa arendamisele.
- **Järjepidevus:** Raamistikud aitavad säilitada kujunduse järjepidevust kogu projekti või ettevõtte veebilehtede vahel.
- **Reageerivus:** Enamik raamistikke pakub sisseehitatud tuge reageerivale disainile, mis tagab veebilehtede korrektse kuvamise erinevatel seadmetel.

CSS-i raamistike piirangud

- **Kohandatavuse puudumine:** Mõned raamistikud võivad olla üleliia piiravad, jättes arendajatele vähem võimalusi kohandamiseks.
- **Õppimiskõver:** Iga raamistik nõuab spetsiifiliste klasside ja süsteemide õppimist, mis võib alguses tunduda keeruline.
- **Täiendav koormus:** Raamistike kasutamine võib suurendada veebilehe laadimisaega, eriti kui kasutatakse palju sisseehitatud funktsioone, mida tegelikult ei vajata.

Bootstrap

Bootstrap on üks kõige laialdasemalt kasutatavaid raamistikke, mis pakub rikkalikku valikut komponente nagu nupud, vormid ja navigeerimisribad. Bootstrap on tuntud oma paindliku *grid*-süsteemi poolest, mis toetab reageerivat veebidisaini.

Bootstrap-i paigaldamine

Bootstrap-i saab paigaldada mitmel viisil:

- Bootstrap-i CDN
- Bootstrap-i allalaadimine
- Bootstrap-i npm-i kaudu
- ...

Bootstrap-i CDN

```
<!-- Bootstrapi CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/5.3.2/css/bootstrap.min.css">

<!-- Valikuline JavaScript -->
<!-- jQuery, Popper.js ja Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/5.3.2/js/bootstrap.min.js"></script>
```

Bootstrap-i grid-süsteem

Bootstrapi grid-süsteem on 12-veeruline süsteem, mis võimaldab arendajatel luua reageerivaid ja dünaamilisi paigutusi. Grid-süsteem jagab ekraani 12 võrdseks veeruleks, mida saab kasutada elementide paigutamiseks vastavalt nende suurusele.

Grid veerud lisatakse tavaliselt `<div class="row">` elemendi sisse, mis omakorda on paigutatud `<div class="container">` või `<div class="container-fluid">` elemendi sisse.

Grid-klassid

Grid-süsteemi kasutamiseks lisatakse HTML-elementidele klassid, mis määravad nende laiuse vastavalt 12-veerulisele süsteemile.

Näiteks:

- `.col-6` : Element võtab 6 veergu (pool ekraanist).
- `.col-4` : Element võtab 4 veergu (kolmandik ekraanist).
- `.col-3` : Element võtab 3 veergu (veerand ekraanist).
- `.col-12` : Element võtab kogu laiuse.

Bootstrap nupud

- `.btn` : Põhiline nupu klass.
- `.btn-primary` : Primaarne nupp.
- `.btn-secondary` : Teisene nupp.
- `.btn-success` : Edukas nupp.
- `.btn-danger` : Ohtlik nupp.
- `.btn-warning` : Hoiatusnupp.
- `.btn-info` : Infonupp.

Bootstrap Vormid

- `.form-group` : Ümbritseb vormielemendid ja lisab nende vahele ruumi.
- `.form-control` : Määrab vormielemendi stiili.
- `.form-check` : Vormielementide kontrollimiseks kasutatav klass.
- `.form-check-input` : Kontrollimiseks kasutatava sisendi klass.
- `.form-check-label` : Kontrollimiseks kasutatava sildi klass.
- `.form-text` : Vormielemendi abiteksti klass.
- `.form-inline` : Vormi paigutuse klass, mis asetab elemendid üksteise kõrvale.
- `.form-horizontal` : Vormi paigutuse klass, mis asetab elemendid horisontaalselt.
- jne.

Bootstrap Navigeerimisriba

- `.navbar` : Põhiline navigeerimisriba klass.
- `.navbar-expand` : Määrab navigeerimisriba laiendamise reeglid.
- `.navbar-light` : Heleda taustaga navigeerimisriba.
- `.navbar-dark` : Tume taustaga navigeerimisriba.
- `.navbar-brand` : Navigeerimisriba logo või brändi nimi.
- `.navbar-nav` : Navigeerimisriba menüü klass.
- `.nav-item` : Navigeerimisriba menüü elemendi klass.
- `.nav-link` : Navigeerimisriba menüü lingi klass.
- jne.

Andmete pärimine välisest allikast

Andmete pärimine välisest allikast on oluline osa veebiarendusest, kuna see võimaldab veebilehtedel saada dünaamilisi andmeid ja suhelda serveritega. Üks populaarsemaid viise andmete pärimiseks on kasutada `Axios` -*library*-t.

Axios

Axios on lubaduspõhine (*Promise based*) HTTP-klient JavaScriptile, mida kasutatakse andmete saatmiseks ja vastuvõtmiseks veebiserverist. See toetab nii brauseri kui ka Node.js keskkondi, muutes selle mitmekülgseks valikuks erinevates arendusstsenaariumites.

Axios-e paigaldamine

Nagu paljusid teisi *library*-sid, saab Axios-i paigaldada mitmel viisil, kuid kuna hetkel kasutame veebilehitsejas töötavat JavaScripti, siis on kõige lihtsam viis kasutada Axios-i CDN-i.

```
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
```

Axios-e kasutamine

Axios-e kasutamine on üsna lihtne ja loogiline. Siin on näide, kuidas teha GET-päring ja käsitleda vastust *async/await* abil:

```
async function fetchPosts() {  
  try {  
    const response = await axios.get('https://jsonplaceholder.typicode.com/todos');  
    console.log(response.data);  
  } catch (error) {  
    console.error('Viga päringus:', error);  
  }  
}  
  
fetchPosts();
```

JS funktsioonide käivitamine HTML-is

Kui siiani oleme kuulnud sündmusi ja käivitanud funktsioone JavaScripti kaudu, siis tegelikult on võimalik käivitada JavaScripti funktsioone ka otse HTML-is.

```
<button onclick="myFunction()">Käivita funktsioon</button>
```

```
const myFunction = () => {  
  alert('Hello World!');  
};
```

Sündmuse objekti edastamine

Nagu eelmine loeng rääkisime, siis kaasneb sündmusega objekt, mis sisaldab palju kasulikku teavet sündmuse kohta. Selle objekti edastamine HTML-ist Javascripti käib nii:

```
<button onclick="myFunction(event)">Käivita funktsioon</button>
```

```
const myFunction = (event) => {  
  console.log(event);  
};
```

Kodune töö

- Tee blogi lehekülg kasutades Bootstrap-i.
- Lisa blogisse Axios-i abil andmete pärimine välisest allikast.
 - Näiteks kasuta [JSONPlaceholder](#) API-t
 - Kuva blogi postitused leheküljele.
 - Proovi kuvada ka kommentaarid postituste kohta.