

Programmeerimise algõpetus Javascripti baasil

Martti Raavel

martti.raavel@tlu.ee

Esimene loeng

- Sissejuhatus
- Mis on programmeerimine?
- Mis on algoritm?
- Mis on Javascript?
- Kuidas alustada Javascripti koodi kirjutamist?
- Programmi sisend ja väljund
- Muutuja
- Andmetüübid
- Operaatorid
- Tingimuslause

Sissejuhatus

- Saame tutvavaks
 - Kes, kust, miks?
- Aine ülesehitus
- Kodused tööd

Aine ülesehitus

- Loengud
 - 22.05 kell 10:00 - 13:00
 - 29.05 kell 10:00 - 13:45
 - 05.06 kell 10:00 - 13:45
 - 12.06 kell 10:00 - 13:45
 - 19.06 kell 10:00 - 13:45
- Materjalid Github'is
- Loengute salvestused
- Kodused tööd

Mis on programmeerimine?

Programmeerimine on protsess, kus luuakse ja rakendatakse **algoritme**, kasutades konkreetset **programmeerimiskeelt** (koodi), et arvutit mingisuguse ülesande täitmiseks juhtida.

Mis on algoritm?

Algoritm on täpne juhend või protseduur probleemi lahendamiseks või ülesande täitmiseks.

Mis on JavaScript?

JavaScript on 1990ndate keskpaigas loodud kõrgetasemeline, tõlgendatav programmeerimiskeel, mida peamiselt kasutatakse veebilehtede interaktiivseks muutmiseks.

JavaScript on üks kolmest põhitehnoloogiast, mida kasutatakse veebilehtede ehitamiseks. Teised kaks on **HTML** (*HyperText Markup Language*), mida kasutatakse veebilehe struktuuri kirjeldamiseks, ja **CSS** (*Cascading Style Sheets*), mida kasutatakse veebilehe kujundamiseks.

Kuidas alustada JavaScripti koodi kirjutamist?

- Koodieditor
- HTML-dokument
- JavaScripti kood
 - inline
 - internal
 - external

Koodieditor

Koodieditor aitab kirjutada ja redigeerida koodi.

- koodi esiletõstmine
- automaatne täiendamine
- veateated
- *snippets*
- ...

HTML-dokument

HTML-i on meil vaja selleks, et me saaksime veebilehel JavaScripti koodi käivitada.

HTML dokumendile on vaja teada anda, et me kasutame JavaScripti ja kus see asub.

Kuidas lisame JavaScripti HTML-dokumendile?

- *inline* (reasisene) skript
- *internal* (sisemine) skript
- *external* (väline) skript

| `<script>` element

Inline (reasisene) skript

```
<!DOCTYPE html>
<html lang="et">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScripti Õppimine</title>
</head>
<body>
  <h1 onclick="alert('Tere!')">Klikka siia</h1>
</body>
</html>
```

Internal (sisemine) skript

```
<!DOCTYPE html>
<html lang="et">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScripti Õppimine</title>
  <script>
    alert('Tere tulemast JavaScripti maailma!');
  </script>
</head>
<body>
  <h1>Tere tulemast JavaScripti maailma!</h1>
</body>
</html>
```

External (väline) skript

script.js fail:

```
alert('Tere tulemast JavaScripti maailma!');
```

HTML fail:

```
<!DOCTYPE html>
<html lang="et">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScripti Õppimine</title>
</head>
<body>
  <h1>Tere tulemast JavaScripti maailma!</h1>
  <script src="script.js"></script>
</body>
</html>
```

Programmi sisend ja väljund

- `alert();`
- `confirm();`
- `prompt();`
- `console.log();`

Muutuja

- Deklareerimine
- Nimetamine

Andmetüübid

- String
- Number
- Boolean
- Null
- Undefined

Operaatorid ja avaldised

- Mis on operaator?
- Mis on avaldis?

Mis on operaator?

Operaator on sümbol või sõna, mida kasutatakse teatud arvutuste tegemiseks. Näiteks matemaatilistes avaldistes on operaatorid nagu pluss (**+**), miinus (**-**), korrutamine (*****) ja jagamine (**/**).

Mis on avaldis?

Avaldis on koodijupp, millel on väärtus. Avaldised koosnevad tavaliselt muutujatest, väärtustest, operaatoritest ja funktsioonikutsetest, mis on kombineeritud nii, et neil on mingi väärtus.

Näiteks `5 + 3` on avaldis, mis annab tulemuseks `8`. Toodud näites on `5` ja `3` väärtused ning `+` on operaator, mis lisab need kaks väärtust kokku.

Operaatorite tüübid

- Aritmeetilised operaatorid
- Määramisoperaatorid
- Võrdlusoperaatorid
- Loogikaoperaatorid

Aritmeetilised operaatorid

- $+$ - liitmine
- $-$ - lahutamine
- $*$ - korrutamine
- $/$ - jagamine

Määramisoperaatorid

- `=` - omistamine
- `+=` - liida ja omista
- `-=` - lahuta ja omista
- `*=` - korruta ja omista

Võrdlusoperaatorid

- `==` - võrdlus
- `===` - range võrdlus
- `!=` - mittevõrdne
- `!==` - rangelt mittevõrdne
- `\>` - suurem
- `\>=` - suurem või võrdne
- `\<` - väiksem
- `\<=` - väiksem või võrdne

Loogikaoperaatorid

- `&&` - loogiline JA
- `||` - loogiline VÕI
- `!` - loogiline EI

Tingimuslause

Tingimuslause on programmeerimise kontrollstruktuur, mis võimaldab meil teha otsuseid ja käituda vastavalt sellele, kas tingimus on tõene või väär.

Tingimuslause tüübid

- `if`
- `else`
- `else if`

if tingimuslause

```
let x = 10;  
  
if (x > 5) {  
    console.log('x on suurem kui 5');  
}
```

else tingimuslause

```
let x = 10;

if (x > 5) {
  console.log('x on suurem kui 5');
} else {
  console.log('x on väiksem või võrdne 5-ga');
}
```

else if tingimuslause

```
let x = 10;

if (x > 5) {
  console.log('x on suurem kui 5');
} else if (x < 5) {
  console.log('x on väiksem kui 5');
} else {
  console.log('x on võrdne 5-ga');
}
```

Tüübiteisendus / type casting

- Mis on tüübiteisendus?
- Implicit (ehk vaikimisi) tüübiteisendus
- Explicit (ehk otsene) tüübiteisendus

Kuidas tüübiteisendusega toime tulla?

- `==` ja `!=` operaatorid
- `===` ja `!==` operaatorid
- `Number()`
- `String()`

Kodune ülesanne

- [Kodune ülesanne](#)