

# Veebiarendus

Martti Raavel

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

# Tarkvaraarendus

- VS Code laiendused ja nipid
- Käsurea kasutamine (*terminal*)

## VS Code klahvikombinatsioonid - Otsing

- `Ctrl + P` - Avab kiire otsingu, kus saad otsida faile projekti kaustast
- `Ctrl + F` - Avab otsingu praeguses failis
- `Ctrl + Shift + F` - Avab globaalse otsingu kogu projekti kaustast
- `Ctrl + H` - Avab asendamise praeguses failis
- `Ctrl + Shift + H` - Avab globaalse asendamise kogu projekti kaustast

## VS Code klahvikombinatsioonid - Kommentaarid

- `Alt + Shift + A` - Valitud ridade 'välja' ja 'sisse' kommenteerimine

## VS Code klahvikombinatsioonid - Kopeerimine/Lõikamine/Kustutamine/Valimine

- `Alt + Shift + Nool üles` - Valitud ridade ülespoole kopeerimine
- `Alt + Shift + Nool alla` - Valitud ridade allapoole kopeerimine
- `Ctrl + C` - Kopeerib valitud rea (ka siis, kui rida pole valitud)
- `Ctrl + X` - Lõikab valitud rea (ka siis, kui rida pole valitud)
- `Ctrl + Shift + K` - Kustutab valitud rea, kus kursor asub (ei lähe lõikelauale)
- `Ctrl + Shift + L` - Valib kõik esinemised, kus kursor asub
- `Ctrl + D` - Valib järgmise esinemise, kus kursor asub

## VS Code klahvikombinatsioonid - Muud

- `Ctrl + Shift + E` - Avab küljepaneeli, kus saad näha kõiki projekti faile ja kaustu
- `Ctrl + Shift + X` - Avab küljepaneeli, kus saad näha kõiki laiendusi
- `Ctrl + Shift + P` - Avab käsurea, kus saad otsida kõiki VS Code'i funktsioone (*Command Palette*)
- `Ctrl + Alt + Nool üles/Nool alla` - Mitu kursorit üles/alla
- `Ctrl + Alt + Nool üles/Nool alla` - Mitu kursorit üles/alla

# VS Code laiendused

- [Markdown All in One](#) - Markdowni lihtsamaks redigeerimiseks ja eelvaateks
- [Prettier](#) - Koodi automaatseks vormindamiseks
- [Git Graph](#) - Git ajaloo graafiline kuvamine
- [Better Comments](#) - Kommentaaride värvimine vastavalt nende tüübile
- [Live Server](#) - Arendusserveri käivitamine ja veebilehtede automaatne värskendamine
- [Live Share](#) - Reaalajas koostöö teiste arendajatega
- [Error Lens](#) - Koodis olevate vigade ja hoiatuste esiletõstmine

Käsurea kasutamine



# Käsureda kasutamine - Windows

- `cd kaustanimi` - Liigub ühest kaustast teise;
- `cd ..` - Liigub ühe kausta võrra ülespoole;
- `dir` - Näitab kausta sisu;
- `mkdir kaustanimi` - Loob uue kausta;
- `del` - Kustutab faili;
- `echo` - Kirjutab teksti konsooli;
- `type failinimi` - Näitab faili sisu;
- `cls` - Puhastab konsooli;
- `exit` - Sulgeb konsooli;

# Käsureda kasutamine - MacOS/Linux

- `ls` - Näitab kausta sisu;
- `cd kaustanimi` - Liigub ühest kaustast teise;
- `cd ..` - Liigub ühe kausta võrra ülespoole;
- `mkdir kaustanimi` - Loob uue kausta;
- `rm failinimi` - Kustutab faili;
- `mv failinimi uusfailinimi` - Muudab faili nime;
- `mv failinimi kaustanimi` - Liigutab faili teise kausta;
- `cp failinimi kaustanimi` - Kopeerib faili teise kausta;
- `cat failinimi` - Näitab faili sisu;
- `touch failinimi` - Loob uue faili;
- `clear` - Puhastab konsooli;
- `exit` - Sulgeb konsooli;

# Git-i käsud

- `git clone` - Kloonib *remote* serverist projekti enda arvutisse
- `git add .` - Lisab kõik muudatused *staging* alasse
- `git commit -m "Muudatuste kirjeldus"` - Salvestab muudatused *commit* ajalukku
- `git push` - Saadab muudatused *remote* serverisse
- `git pull` - Tõmbab muudatused *remote* serverist
- `git status` - Näitab muudatuste olekut
- `git log` - Näitab *commit* ajalugu
- `git branch` - Näitab olemasolevaid *branch*´e
- `git checkout -b uusBranch` - Loob uue *branch*´i ja liigub sinna
- `git merge teineBranch` - Liidab praeguse *branch* i teise *\*branch\** iga

**Programmeerimine**

**Moodulid**

## Moodulid - Mis?

Moodul on Javascripti fail, mis sisaldab koodi, mida saab taaskasutada teistes Javascripti failides.

Moduleid kasutatakse koodi organiseerimiseks loogilistesse üksustesse, mida saab taaskasutada programmi teistes osades.

Moodulid võivad sisaldada funktsioone, objekte, klasside, muutujaid ja muid Javascripti koodi osi.

## Moodulid - Tüübid

- Iseenda kirjutatud moodulid
- Sisseehitatud moodulid
- Kolmanda osapoole moodulid

## Iseenda kirjutatud moodulid

Nagu ka nimetus ütleb, on need moodulid, mille oleme ise kirjutanud näiteks koodi struktureerimiseks ja taaskasutamiseks.

Selleks, et kasutada iseenda kirjutatud mooduleid, tuleb need kõigepealt eksportida ja seejärel vastavasse faili importida.



## Iseenda kirjutatud moodulid - Eksportimine

```
const myModule = {  
  myFunction() {  
    // funktsiooni sisu  
  }  
}  
  
module.exports = myModule;
```

## Iseenda kirjutatud moodulid - importimine

```
const myModule = require('./moduleName');
```

## Iseenda kirjutatud moodulid - kasutamine

```
const myModule = require('./moduleFileName');  
myModule.myFunction();
```

## Sisseehitatud moodulid

Sisseehitatud moodulid on Javascripti enda poolt pakutavad moodulid, mida saab kasutada ilma neid eraldi installimata.

# Sisseehitatud moodulid - nimekiri

- `fs` - failisüsteemi moodul
- `path` - tee moodul
- `os` - operatsioonisüsteemi moodul
- `util` - utiliidi moodul
- `events` - sündmuste moodul
- `http` - HTTP moodul
- `crypto` - krüptograafia moodul
- `zlib` - zlib moodul
- `stream` - voogude moodul
- ...

## Sisseehitatud moodulid - importimine

```
const fs = require('fs');
```

## Sisseehitatud moodulid - kasutamine

```
const fs = require('fs');  
  
const file = fs.readFileSync('file.txt', 'utf8');
```

## Kolmanda osapoole moodulid

Kolmanda osapoole moodulid on moodulid, mida on kirjutanud keegi teine ja mida saab kasutada oma projektis.



# NPM - Node Package Manager

- Node module register
- <https://www.npmjs.com/>

## Kolmanda osapoole moodulid - `package.json` fail

`package.json` fail on fail, kus on kirjas kõik projektiga seotud info, sh projektis kasutatavad kolmanda osapoole moodulid ehk sõltuvused (*dependencies*).

Kolmanda osapoole moodulid - `package.json` faili loomine

```
npm init -y
```

## Kolmanda osapoole moodulid - installimine

```
npm install mooduliNimi
```

Näiteks:

```
npm install prompt-sync
```

## Kolmanda osapoole moodulid - kasutamine

```
const prompts = require('prompt-sync');  
  
const prompt = prompts();  
  
const nimi = prompt('Sisesta oma nimi: ');  
  
console.log(`Tere, ${nimi}!`);
```

## `node_modules` kaust

Kõik kolmanda osapoole moodulid paigutatakse `node_modules` kausta.

## `.gitignore` fail

`node_modules` kaust tuleks lisada `.gitignore` faili, et seda ei lisataks *remote* serverisse.

**.gitignore** faili loomine

```
node_modules/
```



# Sõltuvuste taaspaigaldamine

Kui kopeerid oma projekti teise arvutisse, tuleb kõik sõltuvused taaspaigaldada.

```
npm install
```

