Veebiarendus

Martti Raavel

martti.raavel@tlu.ee

Sissejuhatus

- Saame tuttavaks
- Kursuse ülesehitus
- Kodused tööd
- Hindamine

Saame tuttavaks

- Kes ma olen?
- Eelnev kokkupuude seoses tarkvaraarendusega ja programmeerimisega
- Miks ma siin olen?

Materjalid

https://github.com/HK-Mikrokraadid/Veebiarendus

Discord

Kursuse ülesehitus

- 10 loengut
  - viis loengut kohapeal
  - viis loengut Zoomis
- Kodused tööd iga loengu järel
- Materjalide lugemine enne loengut
- Otsesuhtlus Discordis

Tarkvaraarendus

- Tarkvara ja tarkvaraarendus
- Tööriistad
- Versioonikontroll
- Git
- Github
- Markdown

Tarkvara ja tarkvaraarendus

- Mis on tarkvara?
- Avatud vs Suletud koodiga tarkvara
- Kust tarkvara saab?
- Mis on tarkvaraarendus?

Mis on tarkvara?

Tarkvara on juhiste või programmide kogum, mis on loodud arvutisüsteemis konkreetsete ülesannete või funktsioonide täitmiseks. Põhimõtteliselt on see juhiste kogum, mis ütleb arvutile, mida ja kuidas teha.

- Süsteemitarkvara
- Rakendustarkvara

Avatud vs Suletud koodiga tarkvara

Tarkvara saab samuti klassifitseerida avaliku lähtekoodiga või suletud lähtekoodiga tarkvaraks. Avaliku lähtekoodiga tarkvara on tarkvara, mis on vabalt kättesaadav ning mida võib (üldjuhul) muuta ja levitada igaüks. Suletud lähtekoodiga on tarkvara, mis kuulub ettevõttele või isikule ja mida ei saa ilma loata kasutada, muuta ega levitada.

Kust tarkvara saab?

Mis on tarkvaraarendus - arutelu

Mis on tarkvaraarendus?

Tarkvaraarendus on tarkvara loomise protsess. See hõlmab:

- kasutajavajaduste analüüsimist;
- tarkvaralahenduste kavandamist;
- koodi kirjutamist;
- tarkvara testimist;
- tarkvara juurutamist
- jne

> Tarkvaraarendus EI VÕRDU programmeerimisega!

Tööriistad

Tarkvaraarenduses üldiselt kasutatakse erinevaid tööriistu, mis aitavad arendusprotsessi sujuvamaks muuta. Sellel kursusel alustame järgmiste tööriistadega:

- Visual Studio Code
  - Markdown All in One pistikprogramm
- git
- Github Desktop
- ...

Versioonihaldus

- Mis on versioonihaldus?
- Miks see oluline on?
- Keskne vs hajutatud versioonihaldus

Mis on versioonihaldus?

Versioonihaldus, tuntud ka kui Lähtekoodi haldus (SCM - Source Code Management) või Versioonikontroll (VCS - Version Control System), on süsteem, mis jälgib ja haldab muudatusi failides, kataloogides ja koodibaasides aja jooksul. See võimaldab mitmel inimesel töötada ühe projekti kallal ilma üksteise kirjutatud koodiga konflikti tekitamata ning säilitab iga muudatuse ajaloo.

Miks versioonihaldus on oluline?

- Ajalugu
- Koostöö
- Tagasipööramine
- Harud ja harude ühendamine
- Dokumentatsioon
- Koodi ülevaatus
- Väljalasete haldus
- Paralleelne arendus
- ...

Keskne vs hajutatud versioonihaldus

Keskne versioonihaldus

- Üks hoidla
- Kõik muudatused salvestatakse ühte kohta

Hajutatud versioonihaldus

- Iga kasutaja kloonib hoidla endale
- Iga kasutaja saab muudatusi teha oma kloonis

Git

- Mis on Git?
- Git-i põhiarhitektuur
- Git-i sõnavara
- Kuidas Git-i kasutada?

Mis on Git?

Git on hajutatud versioonihaldussüsteem (DVCS), mida kasutatakse tarkvaraarenduses lähtekoodi muudatuste jälgimiseks. See on loodud nii väikeste kui ka väga suurte projektidega toimetulekuks kiiresti ja tõhusalt. Git võimaldab mitmel arendajal koostööd teha sama koodibaasi kallal üksteist segamata.

Git-i põhiarhitektuur

- Hoidla (Repository)
- Blob - faili sisu
- Puu (Tree) - kataloogid
- *Commit* - muudatuste salvestamine
- *Branch* - haru
- ...

Git-i sõnavara

- Hoidla (Repository)
- *Commit* - muudatuste salvestamine
- *Branch* - haru
  - *Master* või *Main*
- *Clone* - hoidla koopia
- *Pull* - muudatuste toomine
- *Push* - muudatuste saatmine
- *Merge* - harude ühendamine
- *Pull Request* - tõmbetaotlus
- ...

Kuidas Git-i kasutada?

Github

- Mis on Github?
- Miks kasutada Github-i?
- Github Desktop
- Harjutused

Mis on Github?

GitHub on veebipõhine platvorm, mis pakub majutust tarkvaraarenduseks ja versioonikontrolliks Git'i abil. See pakub Git'i hajutatud versioonikontrolli võimekust koos Github-i poolt lisatud funktsionaalsustega.

Miks kasutada Github-i?

Github Desktop

Harjutused

- Oma repositooriumi loomine
- Kloonimine
- README.md
- Commi
- Push
- ...

Markdown

- Mis on Markdown?
- Markdown-i plussid
- Markdown-i miinused
- Markdown-i süntaks
- Harjutused

Mis on Markdown?

Markdown on kerge märgendikeel, millel on lihtne tekstivormingu süntaks. See loodi John Gruber-i poolt 2004. aastal. Markdowni peamine eesmärk oli muuta inimestel teksti kirjutamine ja vormindamine lihtsaks viisil, mis on vabalt loetav ja mida saab konverteerida HTML-iks (või teisteks väljundvorminguteks).

Markdown-i plussid

- Lihtne
- Kiire
- Loetav
- Konverteeritav
- Ei vaja spetsiaalset tarkvara
- ...

Markdown-i miinused

- Piiratud stiilimine
- Variatsioonid
- Ei pruugi sobida väga suurte dokumentide jaoks
- ...

Markdown-i süntaks

- Pealkirjad
- Loendid
- Rõhutamine
- Lingid
- ...

Markdown-i harjutused

Programmeerimine

- Sissejuhatus
- Tööriistad
- Javascript
- NodeJS
- Hello World
- Muutujad
- Andmetüübid
- Koodi laadimine Githubi

Sissejuhatus (Programmeerimine)

- Mis on selle aine eesmärk?
- Mis on programmeerimine?
- Mida me selles aines õpime?
- Millist keelt me kasutame?
- Milliseid tööriistu me kasutame?

Mis on selle aine eesmärk?

Selle aine eesmärk on tutvuda programmeerimisega. Õpime programmeerimise aluseid, kuidas planeerida ja kirjutada koodi kasutades JavaScripti.

Mis on programmeerimine?

Programmeerimine on protsess, mille käigus kirjutatakse juhiseid arvutile ülesande täitmiseks. Neid juhiseid kirjutatakse programmeerimiskeeles, mis on formaalne keel, mida saab tõlkida masinkoodiks (st binaariks) ja mida arvuti saab täita.

Programmeerimine on loov protsess, mis hõlmab algoritmide kavandamist ja koodi arendamist probleemide lahendamiseks.

Mida me selles aines õpime?

- Andmetüübid
- Juhtstruktuurid
- Operaatorid ja avaldised
- Sisendid ja väljundid
- Tsüklid
- Funktsioonid
- Lihtsate algoritmide kirjutamist
- Koodi silumist
- ...

Tööriistad (Programmeerimine)

- Visual Studio Code
- NodeJS
- Terminal

Javascript

- Mis on Javascript?
- Miks Javascript?
- Javascript-i eelised
- Javascript-i puudused

Mis on Javascript?

Javascript on programmeerimiskeel, mis on algselt mõeldud interaktiivsete veebilehtede loomiseks. Minevikus kasutati Javascripti ainult kliendipoolselt (veebilehitsejas, Front-End). Tänapäeval on võimalus kasutada Javascripti ka serveripoolselt, kasutades NodeJS-i (Back-End). Selle kursuse raames kasutame Javascripti just sellepärast, et saaksime õppida nii kliendi- kui ka serveripoolset arendust kasutades sama programmeerimiskeelt.

Javascript-i eelised

- Lihtne õppida
- Lihtne kasutada
- Laialdaselt kasutatav
- Veebiarenduse jaoks hädavajalik
- Universaalne
- ...

Javascript-i puudused

- Tüüpide puudumine
- Ei kompileerita enne käivitamist
- ...

NodeJS

- Mis on NodeJS?
- Miks NodeJS?
- NodeJS-i eelised
- Kuidas NodeJS-i kasutada?

Mis on NodeJS?

NodeJS on Javascripti käituskeskkond, mis võimaldab meil käivitada Javascripti koodi väljaspool veebilehitsejat. NodeJS on serveripoolne keel, mis tähendab, et see töötab serveris ja vastupidiselt veebilehitsejas töötavale Javascriptile, on NodeJS-il ligipääs arvuti failisüsteemile. Failisüsteemile ligipääs võimaldab NodeJS-i abil näiteks lugeda, luua ja muuta arvuti andmekandjatelt faile, mida ei saa teha veebilehitsejas töötava Javascriptiga.

Kuidas NodeJS-i kasutada?

Hello World

- Mis on Hello World?
- Kuidas Hello World-i kirjutada?
- Kuidas NodeJS-is programme käivitada?
- Harjutus

Muutujad

- Mis on muutuja?
- Muutuja deklareerimine
- Muutujale väärtuse omistamine
- Harjutused

Mis on muutuja?

Muutuja on nimetatud piirkond arvuti mälus, mida saab kasutada andmete salvestamiseks. Muutujaid kasutatakse väärtuste salvestamiseks, mida saab programmi täitmise ajal kirjutada, lugeda ja muuta.

Põhimõtteliselt on muutuja nimetatud mäluala, millele on omistatud nimi ja millele saab omistada väärtuse.

Muutuja deklareerimine

- `var`
- `let`
- `const`

```
let firstName;
let lastName;
```

Muutujale väärtuse omistamine

Muutujatele saame väärtusi omistada kasutades omistamisoperaatorit `=`. Paremal pool omistamisoperaatorit olev väärtus omistatakse vasakul pool omistamisoperaatorit olevale muutujale.

```
let firstName = "John";
let lastName = "Doe";
const taxRate = 0.22;
```

Andmetüübid

- Mis on andmetüüp?
- Primitiivsed andmetüübid
- Andmestruktuurid
- Muutuja väärtuse tüübi muutmine

Mis on andmetüüp?

Nagu me juba teame, on muutuja nimetatud piirkond arvuti mälus, mida saab kasutada andmete salvestamiseks. Muutujas salvestatavatel andmetel on tüüp. Tüüp määrab, milliseid toiminguid saab andmetega teha. Näiteks saame liita kaks numbrit, kuid me ei saa liita numbrit ja stringi (sõnet).

Primitiivsed andmetüübid

- `String` (sõne)
- `Number` (number)
- `Boolean` (tõeväärtus)
- `Null` (tühi)
- `Undefined` (määratlemata)
- `Symbol` (sümbol)
- `BigInt` (suur täisarv)

Andmestruktuurid

- `Object` (objekt)
- `Array` (massiiv)

Muutuja väärtuse tüübi muutmine

Javascript on dünaamiliselt tüübitud keel, mis tähendab, et muutuja tüüpi saab programmi täitmise ajal muuta. Näiteks võime muutujale omistada numbri ja hiljem programmis omistada samale muutujale stringi. See erineb staatiliselt tüübitud keeltest nagu Java, kus muutuja tüüp on deklareerimisel fikseeritud.

Koodi üleslaadimine Githubi

- Harjutused

Kodune töö

- Lugege läbi esimese loengu materjalid
- Tehke läbi materjalides olevad harjutused
- Laadige tehtud harjutuste kood Githubi

Küsimused

Aitäh