

Veebiarendus

Martti Raavel

martti.raavel@tlu.ee

Tarkvaraarendus

- Eelmise loengu meeldetuletus
- Kodutööde ajal tekkinud probleemide lahendamine
- Github Issue
- .gitignore
- Harud, Tõmbetaotlus ja ühendamine
- Git ja Githubi parimad tavad

Millest rääkisime eelmisel korral?

Kodutööde ajal tekkinud probleemid

- Githubi Markdowni iseärasused
- Koodi käivitamine õigest kaustast
- Veel midagi?

Github Issue

- Githubi kontekstis on Issue omadus, mis võimaldab kasutajatel jälgida konkreetse hoidla ülesandeid, vigu ja funktsioonitaotlusi.
- Issue saab olla avatud või suletud.
- Issue -le saab määrata silte, vastutajaid, kommentaare ja muid atribuute.

Kuidas **Issue**-t luua

- Lisa oma hoidlale **collaborator**
- Loo oma hoidlale **Issue**
- Määra **Issue**-le **Assignee**-ks **collaborator**

`.gitignore`

- `.gitignore` fail on fail, mis sisaldab hoidla jaoks ignoreeritavaid faile ja kaustu.

Mida lisatakse `.gitignore` faili?

- Kõik failid ja kaustad, mida ei soovita hoidlasse lisada.
- Näiteks:
 - `.env` failid
 - `node_modules` kaustad
 - `dist` kaustad
 - Saladusi sisaldavad failid

`.gitignore` harjutus

- Loo oma hoidlale `.gitignore` fail
- Lisa faili kirje `draft.md`
- `commit` -i ja `push` -i oma muudatused
- lisa fail `draft.md` oma hoidlasse
- `commit` -i ja `push` -i oma muudatused
- Mis juhtus?

Harud, Tõmbetaotlus ja ühendamine

- Harude loomine
- Harusse vahetamine
- Töö harus
- Tõmbetaotluse loomine
- Tõmbetaotluse ülevaatamine ja ühendamine
- Haru kustutamine

Harude loomise protsessi harjutus

- Loo oma hoidlale uus haru
- Tee mõned muudatused uues harus
- `commit` -i ja `push` -i muudatused
- Loo tõmbetaotlus
- Ühenda tõmbetaotlus peamise haruga
- Kustuta haru

Git ja Githubi parimad tavad

- `commit` -i tihti
- Kirjuta mõistlikud `commit` -i sõnumid
- Kasuta `.gitignore` faili
- Ära lisa saladusi sisaldavaid faile hoidlasse
- Kasuta harusid
- Sünkroniseeri tihti
- Lisa alati hoidlasse `README.md` fail (ka alamkaustadesse)

Programmeerimine

- Eelmise loengu meeldetuletus
- Kodutööde ajal tekkinud probleemide lahendamine
- Operaatorid ja avaldised
- Tingimuslaused

Millest rääkisime eelmises loengus?

Milliseid probleeme tekkis kodutööde ajal?

Operaatorid ja avaldised 1

- Operaator on sümbol, mida kasutatakse ühe või mitme väärtusega toimingu sooritamiseks
- Avaldis on väärtuste, muutujate, operaatorite ja funktsioonide kombinatsioon, mille abil arvutatakse uus väärtus

Operaatorid ja avaldised 2

```
let sum = 5 + 10;
```

- `5 + 10` on avaldis
- `sum` on muutuja nimi, kuhu salvestatakse avaldise tulemus
- `+` on operaator
- `5` ja `10` on operandid

Operaatorid ja avaldised 3

- Operaatorid:
 - Aritmeetilised operaatorid
 - Määramisoperaatorid
 - Võrdlusoperaatorid
 - Loogilised operaatorid
 - Stringi operaatorid

Arithmeetilised operaatorid

- `+` - liitmine
- `-` - lahutamine
- `*` - korrutamine
- `/` - jagamine
- `%` - jääk

Määramisoperaatorid

- `=` - määramine, omistamine
- `+=` - liitmine ja omistamine
- `-=` - lahutamine ja omistamine
- `*=` - korrutamine ja omistamine
- `/=` - jagamine ja omistamine
- `%=` - jäägi omistamine
- `++` - suurendamine
- `--` - vähendamine
- `**` - astendamine
- `**=` - astendamine ja omistamine

Võrdlusoperaatorid

Võrdlusoperaatoreid kasutatakse kahe väärtuse võrdlemiseks ja tõeväärtuse tagastamiseks - tulemuseks on `true` või `false`.

- `==` - võrdub
- `===` - võrdub ja on sama tüüpi
- `!=` - ei võrdu
- `!==` - ei võrdu ja ei ole sama tüüpi
- `>` - suurem kui
- `<` - väiksem kui
- `>=` - suurem või võrdne
- `<=` - väiksem või võrdne

Loogilised operaatorid

- `&&` - ja (AND)
- `||` - või (OR)
- `!` - mitte (NOT)

Stringi operaatorid

- `+` - liitmine

Tingimuslaused

Tingimuslause on kontrollmehanism, mida kasutatakse tingimuse alusel valikute tegemisel.

Tingimuslausetega seotud mõisted

- Tõeväärtus avaldis

Tõeväärtus avaldis

Tõeväärtus avaldis on avaldis, mis hindab avaldise väärtust tõeseks (`true`) või vääraks (`false`). Selleks kasutatakse võrdlusoperaatoreid.

Tõeväärtusavaldis võib olla ka muutuja, mille väärtus on `true` või `false` .

Avaldist kasutatakse tingimuslausetes otsustamaks, kas koodiplokk täidetakse või mitte.

Tingimuslausete tüübid

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

if

if lauset kasutatakse koodiploki täitmiseks, kui tingimus on tõene.

if lausel on järgmine süntaks:

```
if (tingimus) {  
    // kood, mida täidetakse, kui tingimus on tõene  
}
```

if näide

```
let weather = 'päikeseline';  
  
if (weather === 'päikeseline') {  
  console.log('Lähen randa!');  
}
```

if-else

`if-else` lauset kasutatakse koodiploki täitmiseks, kui tingimus on tõene, ja teise koodiploki täitmiseks, kui tingimus on väär.

`if-else` lausel on järgmine süntaks:

```
if (tingimus) {  
    // kood, mida täidetakse, kui tingimus on tõene  
} else {  
    // kood, mida täidetakse, kui tingimus on väär  
}
```

if-else näide

```
weather = 'päikeseline';

if (ilm === 'päikeseline') {
  console.log('Lähen rand!');
} else {
  console.log('Lähen kinno!');
}
```

if-else-if

if-else-if lauset kasutatakse juhul, kui on vaja kontrollida mitut tingimust.

if-else-if lausel on järgmine süntaks:

```
if (tingimus1) {  
    // kood, mida täidetakse, kui tingimus1 on tõene  
} else if (tingimus2) {  
    // kood, mida täidetakse, kui tingimus2 on tõene  
} else {  
    // kood, mida täidetakse, kui ükski tingimus ei ole tõene  
}
```


if-else-if näide

```
let number = 5;

if (number > 0) {
  console.log('Number on positiivne');
} else if (number < 0) {
  console.log('Number on negatiivne');
} else {
  console.log('Number on null');
}
```

Tänane kodutöö

- Loe läbi tänase loengu materjalid
- Tee läbi materjalides olevad harjutused
- Laadi tehtud harjutuste kood Githubi

