

Programmeerimise algõpetus Javascripti baasil

Martti Raavel

martti.raavel@tlu.ee

Tänaſed teemad

- Milleſt rääkisime eelmisel korral?
- Funktsioonid
- Massiiv
- Tegevused massiividega

Millest rääkisime eelmisel korral?

Funktsioon - Mis?

Funktsioon on korduvkasutatav koodiplokk, mis on mõeldud teatud ülesande täitmiseks.

Funktsioone kasutatakse koodi korraldamiseks loogilistesse üksustesse, mida saab programmi teistes osades uuesti kasutada.

Funktsioonid

Milliseid funktsioone oleme juba kasutanud?

- `console.log()`
- `alert()`
- `prompt()`
- `confirm()`
- `Number()`
- `String()`
- ...

Funktsiooni - koosneb

- märksõna funktsiooni deklareerimiseks
- funktsiooni nimi
- funktsiooni parameetrid (valikuline), mis on sulgude sees
- funktsiooni sisu/keha, mis on loogeliste sulgude `{}` vahel
- tagastusavaldus funktsiooni kehas (valikuline) - `return` märksõna
- tagastusväärtus `return` järel (valikuline)

Funktsioon - Näide

```
function add(a, b) {  
  const sum = a + b;  
  return sum;  
}
```

Funktsiooni deklareerimine

Javascriptis saame funktsioone deklareerida kahel viisil:

- funktsiooni deklaratsioon;
- funktsiooni avaldis.

Funktsiooni deklaratsioon

```
function functionName(parameter1, parameter2) {  
    // funktsiooni sisu  
    return value;  
}
```

Funktsiooni avaldis

```
const add = function(a, b) {  
  const sum = a + b;  
  return sum;  
}
```

Funktsiooni väljakutsumine (käivitamine)

```
const sum = add(1, 2);  
  
console.log(sum); // 3
```

Funktsioonid - tasub meeles pidada

- Funktsiooni nimi peab olema unikaalne. Kui funktsioonil on sama nimi, mis mõnel teisel funktsioonil, siis see eelnev funktsioon kirjutatakse üle
- Funktsioon tagastab alati mingi väärtuse. Kui funktsioonil ei ole `return` lauset, siis tagastab funktsioon `undefined` väärtuse
- Funktsiooni poolt tagastatava väärtuse saame salvestada muutujasse

Harjutused funktsioonidega

Funktsioon - Harjutus 1

Loo funktsioon, mis võtab sisendiks kaks arvu ja tagastab nende summa.

```
// Testi funktsiooni  
console.log(sum(3, 5)); // Oodatav väljund: 8  
console.log(sum(-2, 7)); // Oodatav väljund: 5
```

Funktsioon - Harjutus 2

Funktsioon, mis tervitab kasutajat

Loo funktsioon, mis võtab sisendiks kasutaja nime ja tagastab tervitussõnumi.

```
console.log(greetUser('Alice')); // Oodatav väljund: Hello, Alice! Welcome to our site.  
console.log(greetUser('Bob'));
```

Funktsioon - Harjutus 3

Funktsioon, mis arvutab kolmnurga pindala

Loo funktsioon, mis võtab sisendiks kolmnurga aluse ja kõrguse ning tagastab kolmnurga pindala. Pindala arvutamise valem on $(\text{alus} * \text{kõrgus}) / 2$

```
console.log(triangleArea(5, 10)); // Oodatav väljund: 25  
console.log(triangleArea(8, 4)); // Oodatav väljund: 16
```


Funktsioon - Harjutus 4

Funktsioon, mis kontrollib paaris ja paaritu arvu

Loo funktsioon, mis võtab sisendiks arvu ja tagastab sõnumi, kas see arv on paaris või paaritu.

```
console.log(isEvenOrOdd(4)); // Oodatav väljund: The number is even.  
console.log(isEvenOrOdd(7));
```

Massiivid

Massiiv on andmestruktuur, mis võimaldab salvestada ühe muutuja alla palju erinevaid väärtuseid.

Massiiv on väärtuste kogum, mis on salvestatud ühte muutujasse. Tavaliselt kasutatakse massiive omavahel seotud andmekogumite salvestamiseks.

Massiivi loomine

```
const numbers = [1, 2, 3];  
const fruits = ['õun', 'pirn', 'banaan'];  
const mixed = [1, 'õun', true];
```

Kuidas massiivist väärtuseid kätte saab?

Massiiv on indekseeritud kollektsioon, mis tähendab, et igal elemendil massiivis on oma indeks, ehk järjekorranumber .

Massiivis olevaid elemente saab kätte nende järjekorranumbri (`index`) järgi, seejuures peab arvestama, et järjekorranumbrid algavad `0` -ga. Järjekorranumbrit, ehk indeksit saab kasutada kujul: `massiiviNimi[index]` ;

Kuidas massiivist väärtuseid kätte saab?

```
const days = ['Esmaspäev', 'Teisipäev', 'Kolmapäev', 'Neljapäev', 'Reede', 'Laupäev', 'Pühapäev'];  
console.log(days[0]); // Väljastatakse massiivi esimene element, ehk 'Esmaspäev'  
console.log(days[3]); // Väljastatakse massiivi neljas element, ehk 'Neljapäev'  
console.log(days[6]); // Väljastatakse massiivi seitsmes element, ehk 'Pühapäev'
```

Massiivide meetodid

Massiividel on 'kaasas' hulk funktsioone ja omadusi, mille abil saab massividesse näiteks elemente juurde lisada, elemente eemaldada, elemente otsida jne.

Massiivide meetodid ja omadused

```
const days = ['Esmaspäev', 'Teisipäev', 'Kolmapäev', 'Neljapäev', 'Reede', 'Laupäev', 'Pühapäev'];

console.log(days.length); // (Omadus) Massiivis olevate elementide arv - kuvatakse 7
console.log(days.push('Puhkepäev')); // Lisab elemendi massiivi lõppu - funktsioon tagastab uue massiivi elementide arvu - 8
console.log(days.pop()); // Eemaldab massiivi lõpust elemendi - tagastatakse eemaldatud elemendi väärtus - 'Puhkepäev',
console.log(days.shift()); // Eemaldab esimese elemendi massiivist - tagastatakse eemaldatud elemendi väärtus - 'Esmaspäev'
console.log(days.indexOf('Kolmapäev')); // Tagastab otsitava väärtuse indeksi - 1 (sest Esmaspäeva ei ole enam massiivis)
```

Massiiv ja tsükkel

```
let days = [  
  'Esmaspäev',  
  'Teisipäev',  
  'Kolmapäev',  
  'Neljapäev',  
  'Reede',  
  'Laupäev',  
  'Pühapäev'  
];  
  
for (let i = 0; i < days.length; i++) {  
  console.log(`Indeksiga ${i} element on massiivis ${days[i]}`);  
}
```


Ülesanne 1

Kirjuta JavaScripti kood, mis loob massiivi, mis sisaldab arvud 1 kuni 10. Seejärel väljasta konsooli massiivi iga element, üksteise järel.

Ülesanne 2

Kirjuta JavaScripti kood, mis küsib kasutajalt 10 arvu. Iga arvu sisestamise järel lisa arv massiivi. Kui kasutaja on kõik arvud sisestanud, siis kuva konsoolis kõik arvud sisestamise järjekorras. Arvuta ka arvude summa ja aritmeetiline keskmine ning kuva need kasutajale.

Kodune ülesanne

- Kodune ülesanne