

Javascript les 7

funcities

functie

=

mini-systeem binnen je code

input -> transformatie -> output

voorbeelden?

functie
waarom?

functie

waarom?

- herhalende code makkelijk gebruiken
 - code leesbaar houden

Maken

```
function naam(param) {  
  //functie-code  
  
}
```

Gebruiken

```
naam(param);
```

Maken

functienaam

Gebruiken

```
function naam(param) {  
    //functie-code  
  
}
```

```
//naam mag alles zijn.  
//Beschrijft wat de  
//functie doet.
```

```
naam(param);
```

Maken

functienaam

Gebruiken

```
function square(val) {  
    val = val * val;  
    console.log(val);  
}
```

```
//bijvoorbeeld om het  
//kwadraat uit te  
//rekenen
```

```
square(10);  
//resultaat = 100
```

Maken

functienaam

Gebruiken

```
function square(val) {  
  val = val * val;  
  console.log(val);  
}
```



//ook een functie


```
square(10);  
//resultaat = 100
```


Maken

functienaam

Gebruiken

```
function chord(notes) {  
  makeNote(notes[0],0.5,250);  
  makeNote(notes[1],0.5,250);  
  makeNote(notes[2],0.5,250);  
}  
  
//look functies
```



```
chord([60,64,68]);  
//resultaat, er worden  
//drie noten  
//afgespeeld
```

Maken

parameters
(input)

Gebruiken

```
function note(pitch) {  
  makeNote(pitch, 0.5, 250);  
}
```

```
//de 60 die is meegegeven  
//aan de functie, komt  
//hier terug.
```

```
note(60);
```

```
note(66);
```


```
//resultaat, er wordt  
//een noot afgespeeld
```

Maken

parameters
(input)

Gebruiken

```
function telop(a,b) {  
  let result = a + b  
  console.log(result);  
}
```



```
//je kan meerdere parameters  
//meegeven
```

```
telop(60,62);  
//resultaat: er wordt  
//122 naar de console  
//gelogd.
```

Maken

parameters
(input)

Gebruiken

```
function chord(notes) {  
  makeNote(notes[0],0.5,250);  
  makeNote(notes[1],0.5,250);  
  makeNote(notes[2],0.5,250);  
}
```

```
chord([60,64,68]);  
//resultaat, er worden  
//drie noten  
//afgespeeld
```

//arrays zijn ook mogelijk

Maken

transformatie

Gebruiken

```
function chord(notes) {  
  //code in de functie  
  //kan alles zijn  
  //wat je maar wil  
}
```

```
chord([60,64,68]);  
//resultaat, dat wat er  
in de functie wordt  
gedaan
```

Maken

output

Gebruiken

```
function note(pitch) {  
  makeNote(pitch, 0.5, 250);  
}
```

```
note(60);  
//resultaat: de functie  
doet iets en dan is het  
klaar (speelt noot af in  
dit geval)
```

Maken

output

Gebruiken

```
function telop(a,b) {  
  let result = a + b  
  return result;  
}  
//return stuurt het resultaat  
terug
```

```
let resultaat = telop(6,7);  
//resultaat: de functie  
doet iets en geeft het  
resultaat terug naar de  
plek waar de functie is  
aangeropen
```

Maken

output

Gebruiken

```
function telop(a,b) {  
  let result = a + b;  
  
  let result2 = a - b;  
  console.log(result2);  
  return [result,result2];  
}
```

//let op!

na return is de functie klaar
dus we zien result2 niet

```
let resultaat = telop(6,7);  
//resultaat: de functie  
doet iets en geeft het  
resultaat terug naar de  
plek waar de functie is  
aangeropen
```


Syntax

```
function functieNaam(argument1, argument2) {  
  //schrijf hier code van de functie  
  console.log("Functie is uitgevoerd met argumenten:");  
  console.log(argument1 + " en " + argument2);  
  
  //geef iets terug aan hetgeen dat de functie aanriep  
  return "hallo, ik heb gewerkt!";  
}  
  
let resultaat = functieNaam("arg1","arg2")
```

Syntax

```
function telOp(num1, num2) {  
    //schrijf hier code van de functie  
    let som = num1+num2;  
  
    //geef iets terug aan hetgeen dat deze functie aanriep  
    return som;  
}  
  
let resultaat = telOp(20,30);  
console.log(resultaat) //in de variabele resultaat staat 50
```

Syntax

```
function telOp(num1, num2) {  
    //schrijf hier code van de functie  
    let som = num1+num2;  
  
    //ga hier los met handige functionaliteiten  
    if (som < 0) {  
        console.log("De som is minder dan nul!");  
    }  
  
    //geef iets terug aan hetgeen dat deze functie aanriep  
    return som;  
}
```

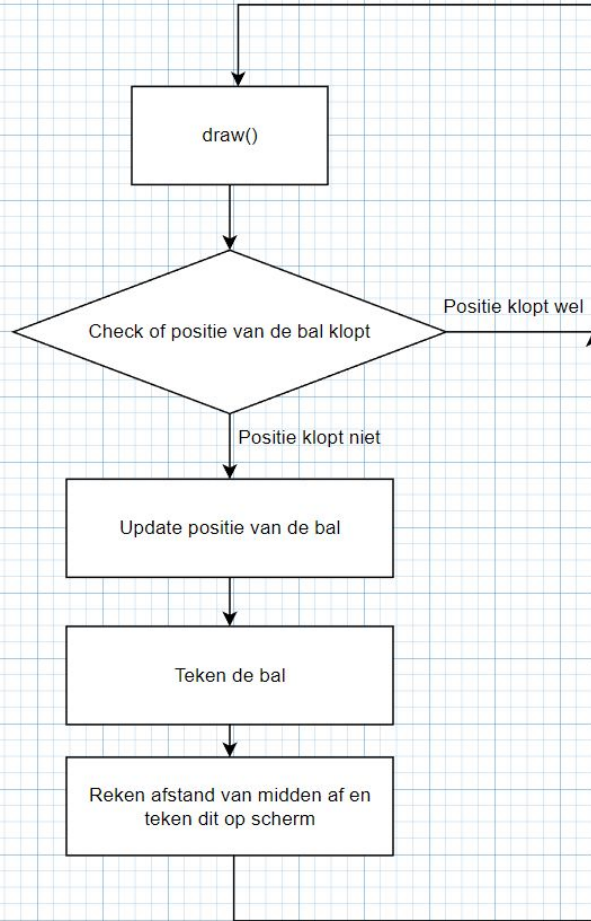
Syntax

```
function makeMusic(note) {  
    //schrijf hier code van de functie  
  
    //ga hier los met handige functionaliteiten  
    if (note >= 60 && note <= 71) {  
        makeNote(note, 0.5, 100);  
    }  
}
```

```
makeMusic(67); //noot wordt afgespeeld  
makeMusic(73); //noot wordt niet afgespeeld
```

Plan:

- Stuk code maken dat laat zien hoe ver links of rechts het balletje in Click to Pan[©] is
- Dit stuk code vangen in een functie (met argumenten)



Opdracht