

Javascript les 12

Classes & objecten deel 2
meer is beter

Definieer een kwartnoot
in P5js



Functies

- Tekenen
- Vallen

Eigenschappen

- grootte
- x-positie
- y-positie

Hoe doen we dit in code?

Classes



Objecten

Hoe doen we dit in code?

Classes

- omschrijving van iets

Objecten

- uit die omschrijving gemaakt object

Hoe doen we dit in code?

Classes

- omschrijving van een kwartnoot

Objecten

- een zichtbare kwartnoot op een positie in het canvas

```
class Note {  
    constructor() {  
        this.x = random(100,500);  
        this.y = random(100,500);  
        this.s = random(10,20);  
    }  
}
```



```
class Note {  
    constructor() {  
        this.x = random(100,500);  
        this.y = random(100,500);  
        this.s = random(10,20);  
    }  
}
```

//start van class-definitie

//geen haakjes ()

// wel een hoofdletter

```
class Note {  
    constructor() {  
        this.x = random(100,500);  
        this.y = random(100,500);  
        this.s = random(10,20);  
    }  
}  
  
//functie binnen class (method)  
//zonder 'function'  
//constructor = vergelijkbaar met setup()
```

```
class Note {  
    constructor() {  
        this.x = random(100,500);  
        this.y = random(100,500);  
        this.s = random(10,20);  
    }  
}
```

//this. refereert naar unieke variabele binnen aangemaakt object

//alleen bekend binnen class-omschrijving

//verder vergelijkbaar met variabele

```
class Note {  
    constructor(newX, newY) {  
        this.x = newX;  
        this.y = newY;  
        this.s = random(10,20);  
    }  
}
```

//constructor met argumenten kan ook!

//zo kan je vanuit buiten waarden meegeven aan je nieuwe object

```
class Note {  
    constructor(newX, newY) {  
        this.x = newX;  
        this.y = newY;  
        this.s = random(10,20);  
    }  
    teken() {  
        ellipse(this.x,this.y,this.s);  
    }  
}
```

//je kan zoveel methods binnen een class maken als je wil

```
class Note {  
    constructor(newX, newY) {  
        this.x = newX;  
        this.y = newY;  
        this.s = random(10,20);  
    }  
    teken() {  
        ellipse(this.x,this.y,this.s);  
    }  
}
```

//functies van een object

```
class Note {  
    constructor(newX, newY) {  
        this.x = newX;  
        this.y = newY;  
        this.s = random(10,20);  
    }  
    teken() {  
        ellipse(this.x,this.y,this.s);  
    }  
}
```

//eigenschappen van een object

```
let note1;

function setup() {
    note1 = new Note(width / 2, height / 2);
}

function draw() {
    note1.teken();
}

//vanuit class nieuw object aanmaken met argumenten (op het midden van het
scherm op alle assen)

//methods van dat object uitvoeren
```



```
let note1;
```

```
function setup() {
```

```
    note1 = new Note(width / 2, height / 2);
```

```
}
```

```
function draw() {
```

```
    note1.teken();
```

```
}
```

```
//variabele waar nieuw object in wordt opgeslagen
```

```
//kan ook array zijn (gaan we zo in voorbeeld uitwerken)
```

```
let note1;

function setup() {
    note1 = new Note(width / 2, height / 2);
}

function draw() {
    note1.teken();
}

//Hier wordt object aangemaakt vanuit classbeschrijving 'Note'
//in de variabele note1
//Op dit moment wordt constructor-method uitgevoerd met argumenten van
buitenaf
```

```
let note1;

function setup() {
    note1 = new Note(width / 2, height / 2);
}

function draw() {
    note1.teken();
}

//Je kunt methods uit de class elk moment aanroepen met
//variabelenaam.methodnaam()

//hiermee voeren we dus de functie teken() uit de classomschrijving uit
```

```
class Note {  
    constructor(newX, newY) {  
        this.x = newX;  
        this.y = newY;  
        this.s = random(10,20);  
    }  
    teken(noteColor) {  
        fill(noteColor);  
        ellipse(this.x,this.y,this.s);  
    }  
}
```

//je kan een method ook argumenten geven om van buiten het object in te vullen

```
let note1;
let note2;
function setup() {
    note1 = new Note(width / 2, height / 2);
    note2 = new Note(width / 4, height / 4);
}
function draw() {
    note1.teken(255);
    note2.teken(0);
}
//ook methods kan je met argumenten uitvoeren
```

Verdere uitwerking in code

- We maken een classomschrijving om een kwartnoot te tekenen
- Bij opstarten van de code verschijnen er een x-aantal op het scherm
- Bij klikken op de muis verschijnt een nieuwe op de positie van de muis
- Bij een druk op een toets vallen de noten naar beneden
- (Als de noot valt klinkt er een toon)