

# Javascript les 10

Classes & objecten  
meer is beter

Dingen in de wereld hebben:

functies

&

eigenschappen

Definieer de algemene functies en eigenschappen van

- een mengpaneel
- een delay-plugin
- een alien uit space invaders



# Welke functies heeft een mengpaneel?

- volume regelen
- panning
- equalizer
- aux-sends
- effecten
- ...

# Welke eigenschappen heeft een mengpaneel?

- analoog / digitaal
- aantal inputs
- aantal pre-amps
- aantal sends
- type eq
- ...

# Welke functies heeft een delay plugin?

- input vertragen
- feedback toevoegen
- modulatie toevoegen
- panning
- ...

# Welke eigenschappen heeft een delay plugin

- Aantal kanalen
- Type emulatie (analoog / (vintage) digitaal / echo)
- aantal instelbare parameters
- ...

# Welke functies heeft een alien (bij space invaders)?

- bewegen
- schieten
- doodgaan
- ...



# Welke eigenschappen heeft een alien

- positie
- vorm
- kleur
- ...

# Maak mixer

- analoog / digitaal = analoog
- aantal inputs = 16
- aantal pre-amps = 8
- aantal sends = 2
- type eq = drie-bands

en nu programmeren

definieer een kwartnoot  
in P5js



# functies kwartnoot

- tekenen

# eigenschappen kwartnoot

- 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() {  
        this.x = random(100,500);  
        this.y = random(100,500);  
        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() {  
        this.x = random(100,500);  
        this.y = random(100,500);  
        this.s = random(10,20);  
    }  
    teken() {  
        ellipse(this.x,this.y,this.s);  
    }  
}
```

```
//functies van een object
```

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

**//eigenschappen van een object**



```
let note1;  
function setup() {  
    note1 = new Note();  
}  
function draw() {  
    note1.teken();  
}  
  
//vanuit class nieuw object aanmaken  
//methods van dat object uitvoeren
```

```
let note1;
```

```
function setup() {
```

```
    note1 = new Note();
```

```
}
```

```
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();
}
function draw() {
    note1.teken();
}
//Hier wordt object aangemaakt vanuit classbeschrijving 'Note'
//in de variabele note1
//Op dit moment wordt constructor-method uitgevoerd
```

```
let note1;  
function setup() {  
    note1 = new Note();  
}
```

```
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

# 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