

Javascript les 1

P5.js, pixels & variabelen

JavaScript

programmeertaal / script

werkt in browser / als app / in Max

P5.js

bibliotheek met extra functies bovenop JavaScript

Audiovisueel

P5.js

index.html

&

sketch.js

P5.js

index.html

open je in de browser

&

sketch.js

in dit bestand pas je de code aan

P5.js

index.html

open je in de browser

&

sketch.js

in dit bestand pas je de code aan

csdosc

een webserver op je eigen computer

geschikt om audio toe te voegen aan p5

&

te communiceren met Max

P5.js - sketch.js

```
function setup(){}
```

wordt één keer uitgevoerd

```
function draw(){}
```

wordt continu herhaald

P5.js - setup()

```
createCanvas(800,600);
```

```
functie(waarde (x),waarde(y));
```

```
background(255);
```

0 (zwart) - 255 (wit)

P5.js - draw()

```
frameRate();
```

animatie

P5.js - canvas

x-as (horizontaal)

y-as (verticaal)

0,0 - width-1,height-1

P5.js - canvas

Voorbeeld

P5.js - iets tekenen

```
point(x,y);
```

```
line(x1,y1,x2,y2);
```

```
square(x,y,s);
```

[en meer](#)

commentaar in javascript

```
//deze code wordt niet uitgevoerd
```

```
/*deze
```

```
regels
```

```
ook niet */
```

commentaar in javascript

Beschrijven wat de code doet

code aan- en uitzetten

sneltoets:

cmd/ctrl + /

commentaar in javascript

“Commenting your code is writing a love letter
to your future self”

Gregory Taylor - Cycling 74

Foutmeldingen

console

[sneltoetsen](#)

variabelen in Javascript

Stukjes in het RAM-geheugen met een naam

Waardes opslaan en terughalen

types

- booleans (true/false)
- getallen (1, 100, 15.6, 3.1415)
 - tekst (“hallo”)
- complexere objecten
(foto's, audiobestanden)

aanmaken

```
let y = 32;
```

let = maak variabele

y = naam

32 = waarde

voorbeelden

boolean

```
let connected = true;
```

```
let sequencerState = false;
```

voorbeelden

int / float

```
let xPosition = 500;
```

```
let xSpeed = 1.2;
```

```
let newPosition = xPosition + xSpeed;
```

voorbeelden

tekst

```
let name = "Roald";
```

```
let greeting = "hallo, mijn naam is ";
```

```
let sentence = name + greeting;
```

voorbeelden

tekst

```
let name = "Roald";
```

```
let greeting = "hallo, mijn naam is ";
```

```
let betterSentence = greeting + name;
```

volgorde

Van boven naar beneden

scope

Geldigheid van een variabele

() = functie

{ } = code block

scope

```
1
2 let y = 0; //dit is global scope
3
4 function setup() {
5   createCanvas(800,600);
6   background(255);
7 }
8
9 function draw() {
10
11
12 }
13
```

y is overal bekend

scope

```
1  function setup() {  
2    let y = 0; |  
3    createCanvas(800,600);  
4    background(255);  
5  
6  
7  
8  }  
9  
10 function draw() {  
11  
12 }  
13
```

y is bekend binnen `setup()`
vanaf regel 2 tot regel 8

scope

```
1  function setup() {  
2    createCanvas(800,600);  
3    background(255);  
4    let y = 0;  
5  
6  
7  
8  }  
9  
10 function draw() {  
11  
12 }  
13
```

y is bekend binnen `setup()`
vanaf regel 4 tot regel 8

scope

```
1  function setup() {  
2    createCanvas(800,600);  
3    background(255);  
4  
5  
6  
7  }  
8  
9  function draw() {  
10   let y = 0;  
11  
12 }  
13
```

} **y** is bekend binnen **draw()**
vanaf regel 10 tot regel 12
en niet daarbuiten

opdracht

online