

Les 1 Creative coding introductie

Lesdoelen:

- Leerlingen maken kennis met het begrip 'creative coding'
- Leerlingen leren werken in P5.js editor
- Leerlingen maken eerste sketch in P5.js editor.



Wat is 'creative coding' ?

Om antwoord te kunnen geven op deze vraag bestuderen leerlingen kort onderstaande linken:

https://en.wikipedia.org/wiki/Creative_coding

<https://processing.org/>

[home | p5.js \(p5js.org\)](https://p5js.org/)

<https://editor.p5js.org/>

les 1 A onderzoek naar vormen en kleuren

leerdoelen:

- Gebruiken van commentaar in codes
- Function setup gebruiken
- Draw functie gebruiken
- Vorm en kleur mbv reference site van P5.js

Welkom in de wereld van p5js, waar je Javascript-code kunt gebruiken om o.a. kunst, interactieve apps en graphics kunt maken .

Met de volgende oefeningen ontdek je de basis van P5j.

1 Code-opmerkingen

Het eerste stukje code dat je moet kennen, doet eigenlijk helemaal niets: commentaar. Opmerkingen zijn er in twee varianten:

- enkele lijn
- meerdere lijnen

Commentaar op één regel begint met twee schuine strepen. Overal waar er twee schuine strepen vlak naast elkaar staan, wordt de rest van de tekst op die regel genegeerd.

Commentaar dat uit meerdere regels bestaat, begint met een schuine streep, onmiddellijk gevolgd door een asterisk (ook wel sterkarakter genoemd). Zoals de naam al aangeeft, kan een commentaar van meerdere regels over meerdere regels worden verspreid, en het eindigt pas als er nog een asterisk is gevolgd door een schuine streep. (Opmerking: commentaar met meerdere regels kan ook op één regel worden gebruikt, met als voordeel dat er meer code kan komen na het einde van het commentaar.

Opmerkingen doen niets wanneer een schets wordt uitgevoerd, maar ze zijn nuttig om uit te leggen wat de code eromheen doet, en je zult er veel van zien in deze tutorial.

mySketch

```
1  // This is a single line comment
2
3  /*
4  This is a multi-line comment
5  */
```

2 function setup {}

Elke p5js-schets heeft een setup-functie nodig.

Een functie is een manier om een reeks JavaScript-instructies te groeperen en deze een naam te geven. U kunt uw eigen functies maken die u later kunt oproepen, wat een handige manier is om dezelfde regels code opnieuw te gebruiken zonder ze herhaaldelijk op verschillende plaatsen te

typen. Maar in het geval van de setup-functie is dit een speciaal genoemde functie die p5js zal aanroepen wanneer de schets voor het eerst begint.

Een functie in javascript bestaat uit de volgende onderdelen:

- Het functie trefwoord
- De naam van de functie (moet beginnen met een letter en mag alleen letters, cijfers en onderstrepingstekens bevatten).
- Een lijst met parameters tussen haakjes. De haakjes zijn vereist, zelfs als er geen parameters zijn.
- De 'body' van de functie begint met een open accolade { gevolgd door een reeks uitspraken.
- Deze instructies worden elke keer uitgevoerd wanneer de functie wordt aangeroepen. Eindigt met een sluitende accolade }

mySketch

```
1 // This function will automatically be called once
2 // when we run our sketch
3 function setup() {
4   // Our setup code will go here.
5 }
```

3 CreateCanvas

Het belangrijkste dat in function setup {} gebeurt, is het maken van een canvas, waar de afbeeldingen worden weergegeven. Om dit te doen noemen we de functie p5js createCanvas.

De code createCanvas heeft twee parameters nodig*: de breedte van het canvas en de hoogte van het canvas. Om te beginnen maak je een canvas van 400 vierkante pixels.

mySketch

```
1 // This function will automatically be called once
2 // when we run our sketch
3 function setup() {
4   createCanvas(400, 400);
5 }
```

4 Het canvas vullen met een kleur

Momenteel is je schets behoorlijk saai, het is gewoon een effen wit canvas op een effen witte pagina (vrijwel onzichtbaar). Maak het zichtbaar door het te vullen met een enkele effen kleur.

De achtergrondfunctie vult het canvas met een kleur. Er zijn verschillende manieren om aan te geven welke kleur je wilt gebruiken.

Genoemde kleuren

Bepaalde kleuren kunnen worden opgegeven met behulp van een tekstwaarde die de naam van de kleur specificeert. Een tekstwaarde in JavaScript, ook wel een string genoemd, is een reeks tekens tussen 'enkele' of 'dubbele' aanhalingstekens*.

```
// fill the canvas with light blue  
background( 'LightBlue' );
```

Je kunt een lijst met benoemde kleuren vinden op wikipedia

https://en.wikipedia.org/wiki/Web_colors#Extended_colors , of gebruik deze tool

<https://enes.in/sorted-colors/> om de namen te vinden kleuren met een bepaalde tint.

* Let op bij het plakken van code uit bepaalde toepassingen; sommige programma's veranderen normale aanhalingstekens in "linker en rechter" aanhalingstekens. Javascript zal dit niet leuk vinden.

Tinten grijs

Grijstinten kunnen ook worden opgegeven met een enkele gehele waarde van 0 tot 255, waarbij 0 zwart is en 255 wit.

```
// fill the canvas 50% gray  
background(127);
```

RGB-waarden

Kleuren kunnen ook worden opgegeven met behulp van drie cijfers die de hoeveelheid van de drie primaire kleuren vertegenwoordigen: rood, groen en blauw. Voor meer informatie over dit kleurmodel zie Wikipedia.

```
// the red green and blue values for light blue  
background(173, 216, 230);
```

Hex-codes

In HTML, CSS en Javascript is het gebruikelijk om hexadecimale notatie voor kleuren te gebruiken. Hexadecimaal is een compacte manier om getallen weer te geven met niet alleen de cijfers 0-9, maar ook de letters A-F. Hierdoor kunnen getallen met grondtal 16 worden geschreven in plaats van met grondtal 10. Om deze notatie met p5js te gebruiken, gebruikt u een tekenreekswaarde die begint met een hekje (#), gevolgd door 3 of 6 cijfers (d.w.z. '#add8e6').

mySketch

```
1 // This function will automatically be called once  
2 // when we run our sketch  
3 function setup() {  
4   createCanvas(400, 400);  
5   // Let's make our canvas light blue  
6   background('LightBlue');  
7 }
```

5 De tekenfunctie

In theorie zou je al je tekeninstructies in de setup-functie kunnen plaatsen. De setup-functie wordt echter maar één keer uitgevoerd, en dat geldt ook voor een statische, niet-interactieve schets.

De tekenfunctie wordt automatisch herhaaldelijk aangeroepen, zodat uw schets in de loop van de tijd kan veranderen, of als reactie op gebruikersinteractie (klikken, muisbewegingen, enz.).

In dit voorbeeld wordt elke keer dat de tekenfunctie wordt aangeroepen, een cirkel getekend op de plek waar de muis zich momenteel bevindt.

Merk op dat, omdat we het canvas alleen met een effen kleur vullen in de instellingsfunctie (door achtergrond aan te roepen), alle cirkels die we tekenen blijven bestaan.

```
mySketch
1 // Runs only once
2 function setup() {
3   createCanvas(400, 400);
4   // Let's make our canvas light blue
5   background('LightBlue');
6 }
7
8 // Runs repeatedly
9 function draw() {
10  // Draw a circle centered on the current mouse
11  // location, with a diameter of 20
12  circle(mouseX, mouseY, 20);
13 }
```

6 Vormen tekenen

P5js bevat veel verschillende ingebouwde functies voor het tekenen van basisvormen:

arc - Tekent een segment van een cirkel.

circle - Tekent een cirkel.

ellipse - Tekent een ellips (zoals een cirkel, maar met variërende breedte en hoogte).

line - Tekent een enkel lijnsegment van het ene punt naar het andere.

point - Tekent een enkel punt.

quad - Tekent een willekeurige vierzijdige vorm die vier punten verbindt.

rect - Tekent een rechthoek.

square - Tekent een vierkant.

triangle - Tekent een drizijdige vorm die drie punten verbindt.

mySketch

```
1 function setup() {
2   createCanvas(400, 400);
3   // By default angles in p5js are specified in radians.
4   // This switches to degree mode so we can specify
5   // angles from 0 to 360 instead of 0 to 2 * PI
6   angleMode(DEGREES);
7   // Let's make our canvas light blue
8   background('LightBlue');
9 }
10
11 function draw() {
12   // arc(centerX, centerY, width, height, startAngle, endAngle)
13   arc(67, 67, 50, 50, 45, 360 - 45);
14
15   // circle(centerX, centerY, diameter)
16   circle(200, 67, 50);
17
18   // ellipse(centerX, centerY, width, height)
19   ellipse(334, 67, 30, 50);
20
21   // point(x, y)
22   strokeWeight(5); // we need to increase the thickness of the stroke to make the point visible
23   point(67, 200);
24   strokeWeight(1);
25
26   // line(x1, y1, x2, y2)
27   line(175, 225, 225, 175);
28
29   // quad(x1, y1, x2, y2, x3, y3, x4, y4)
30   quad(309, 225, 309, 200, 359, 175, 359, 200);
31
32   // rect(leftX, topY, w, h)
33   rect(42, 314, 50, 40);
34
35   // square(leftX, topY, size)
36   square(176, 309, 50);
37
38   // triangle(x1, y1, x2, y2, x3, y3)
39   triangle(315, 340, 342, 310, 350, 355);
40 }
```

7 Fill en strokes

Tot nu toe zijn alle vormen die we hebben getekend gevuld met wit en omlijnd met zwart (behalve punt en lijn die geen gebied hebben om in te vullen). Om in p5js te veranderen met welke kleur een vorm wordt ingevuld of omlijnd, gebruikt je de vul- en lijnfuncties voordat je de verschillende

vormfuncties aanroept. De kleur die je opgeeft, wordt voor alle volgende vormen gebruikt, totdat je opnieuw een beroep doet op fill of stroke.

```
mySketch
1 function setup() {
2   createCanvas(400, 400);
3   // By default angles in p5js are specified in radians.
4   // This switches to degree mode so we can specify
5   // angles from 0 to 360 instead of 0 to 2 * PI
6   angleMode(DEGREES);
7   // Let's make our canvas light gray
8   background('LightGray');
9   strokeWeight(4);
10 }
11 function draw() {
12   fill('Crimson');
13   stroke('RoyalBlue');
14   arc(67, 67, 50, 50, 45, 360 - 45);
15
16   fill('ForestGreen');
17   stroke('Violet');
18   circle(200, 67, 50);
19
20   fill('RoyalBlue');
21   stroke('Gold');
22   ellipse(334, 67, 30, 50);
23
24   fill('Violet');
25   stroke('DarkTurquoise');
26   strokeWeight(8); // we need to increase the thickness of the stroke to make the point more visible
27   point(67, 200);
28   strokeWeight(4);
29
30   fill('Gold');
31   stroke('DarkOrange');
32   line(175, 225, 225, 175);
33
34   fill('DarkTurquoise');
35   stroke('PaleGreen');
36   quad(309, 225, 309, 200, 359, 175, 359, 200);
37
38   fill('DarkOrange');
39   stroke('SkyBlue');
40   rect(42, 314, 50, 40);
41
42   fill('PaleGreen');
43   stroke('Crimson');
44   square(176, 309, 50);
45
46   fill('SkyBlue');
47   stroke('ForestGreen');
48   triangle(315, 340, 342, 310, 350, 355);
49 }
```