



JAVASCRIPT INTRODUCTIE

05/04/2017

Gebaseerd op
«The Javascript Artist - English.pptx»

Doel

- Op een leuke manier leren programmeren met JavaScript!
- De basisblokken van een programma begrijpen.
- Je vertrouwd maken om te experimenteren met moeilijkere JavaScript code.
- Je creativiteit loslaten
- Een beetje experimenteren met wiskunde!
- Je aanmoedigen om je werk te presenteren.

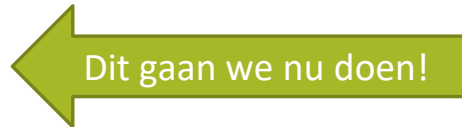
Wat is Javascript?



- Een programmeertaal die in alle browsers werkt.
- Zoals met elke spreektaal (zoals Frans, Engels, Nederlands) moeten we de juiste schrijfwijze respecteren.
- Zonder JavaScript zouden bijna alle websites niet cool zijn! Facebook, Gmail, Twitter gebruiken allemaal JavaScript om hun website gebruiksvriendelijker te maken.

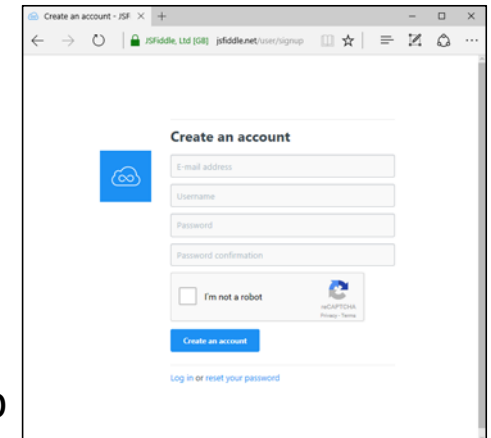
Wat kunnen we doen met Javascript?

- Muziek afspelen
- Acties toevoegen aan een pagina (bijv. Inzoomen op een foto, een e-mail in de vuilnisbak gooien, ...)
- Teken op een webpagina
- Spelletjes bouwen
- En nog veel meer!
Bekijk dit maar eens:
 - <http://lights.helloenjoy.com/>
 - <http://www.patatap.com/>
 - <https://www.cubeslam.com/>



Hoe gaan we programmeren?

- We zullen een online tool gebruiken.
- Surf naar
- Als je je werk wilt bewaren, kan je best een account aanmaken
 - <https://jsfiddle.net/>
- Enkele voorbeelden om mee aan de slag te gaan, kan je vinden op
 - <https://jsfiddle.net/user/FrankD/fiddles/>



De Editor

HTML editor

CSS editor

Uitvoeren

JavaScript editor

Resultaat

The screenshot shows the JSFiddle web editor interface. At the top, there is a navigation bar with icons for Run, Update, Fork, Set as base, Collaborate, and Embed. Below this, the editor is divided into four main sections: a left sidebar, an HTML editor, a JavaScript editor, and a CSS editor. The left sidebar contains a 'Fiddle Meta' section with a title input field, a 'No description' section, and a 'JSFiddle Roadmap' button. The HTML editor contains a single line of code: `<canvas id="canvas1" width="400" height="400"></canvas>`. The JavaScript editor contains a multi-line script that initializes a canvas context and draws a blue line from (10, 10) to (100, 100). The CSS editor contains a single rule: `body { background: #ff0000; }`. The result area at the bottom right is a solid red rectangle. Green arrows point from the labels 'HTML editor', 'CSS editor', 'JavaScript editor', and 'Resultaat' to their respective sections. A black arrow points from the 'Uitvoeren' label to the result area. A green arrow also points from the 'Fiddle Meta' section to the 'Uitvoeren' label.

Run Update Fork Set as base Collaborate Embed

Fiddle Meta

Untitled fiddle

No description

Add title to make the fiddle public

External Resources

AJAX Requests

Legal, Credits and Links

[JSFiddle Roadmap](#)
suggest and vote for features

1 `<canvas id="canvas1" width="400" height="400"></canvas>` HTML

1 `body`

2 `{`

3 `background: #ff0000;`

4 `}`

1 `// het tekendoek (= canvas) ophalen` JAVASCRIPT

2 `var c = document.getElementById('canvas1').getContext('2d');`

3

4 `// lijn tekenen op het canvas`

5 `c.lineWidth = 2;`

6 `c.fillStyle = "#0000ff";`

7 `c.lineWidth = 5;`

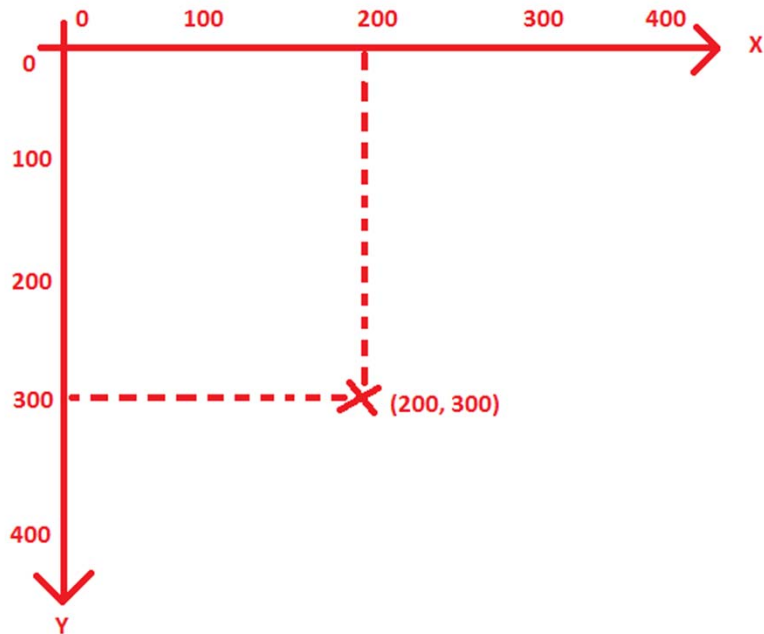
8

9 `c.beginPath();`

10 `c.moveTo(10, 10); // Start positie`

11 `c.lineTo(100, 100); // Eind positie`

12 `c.stroke();`



Het Canvas

- Het gebied waarin we tekenen
- Bijv. 400 x 400 punten
- Gebruikt een (x, y) coördinaten systeem
- X – van links naar rechts
- Y – van boven naar onderen
- Om te oefenen, waar liggen deze punten?
 - (300, 100)
 - (50, 400)
 - (0,0)
 - (400,400)
 - (399, 17)

Ons eerste programma

HTML

```
<canvas id="canvas1" width="400" height="400"></canvas>
```

- Wat krijg je te zien?
- Verander de waardes in moveTo en LineTo
- Kan je een vierkant tekenen?

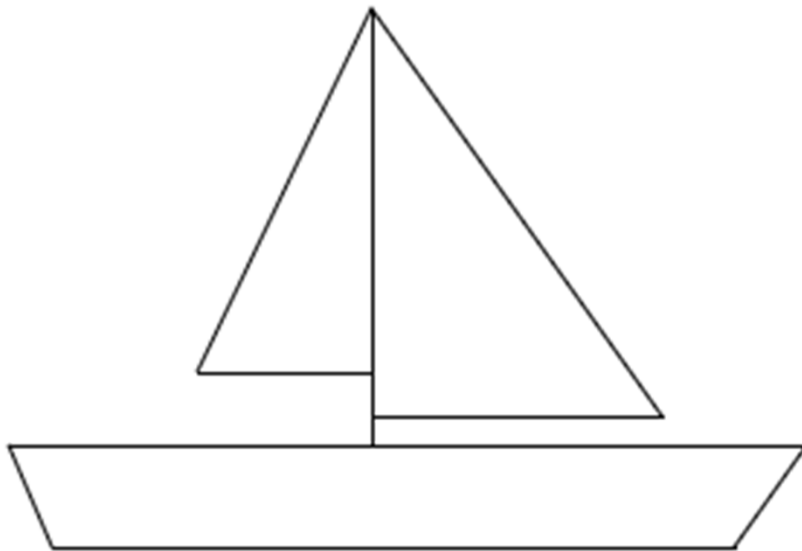
Tip: extra lineTo lijnen nodig...

JavaScript

```
// het tekendoek (= canvas) ophalen
var c = document.getElementById('canvas1').getContext('2d');

// lijn tekenen op het canvas
c.lineWidth = 2;
c.fillStyle = "#0000ff";
c.lineWidth = 5;

c.beginPath();
c.moveTo(10, 10); // Start positie
c.lineTo(100, 100); // Eind positie
c.stroke();
```

Een kleine oefening...

- Kan je een boot tekenen met wat je nu hebt geleerd?
- Vergeet niet je werk te bewaren als je er later wilt verder op werken, of als je het aan andere wil tonen.

Cirkels

- Met deze commando's kan je een cirkel tekenen
- Experimenteer met de straal en hoek om te kijken hoe alles verandert

```
// het tekendoek ophalen
var c = document.getElementById('canvas1').getContext('2d');

// cirkel tekenen
var straal = 75; // straal
var startHoek = 0 * Math.PI;
var eindHoek = 2 * Math.PI;
var tegenDeKlok = false;

c.lineWidth = 5;
c.strokeStyle = "blue";

c.beginPath();
c.arc(100, 100, straal, startHoek, eindHoek, tegenDeKlok);
c.stroke();
```

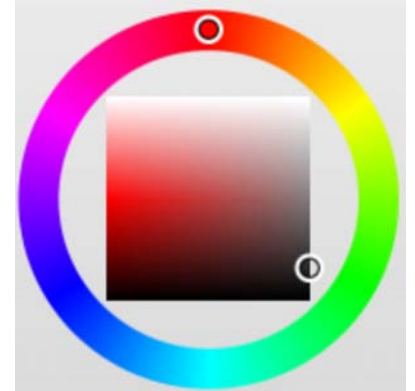
Colors

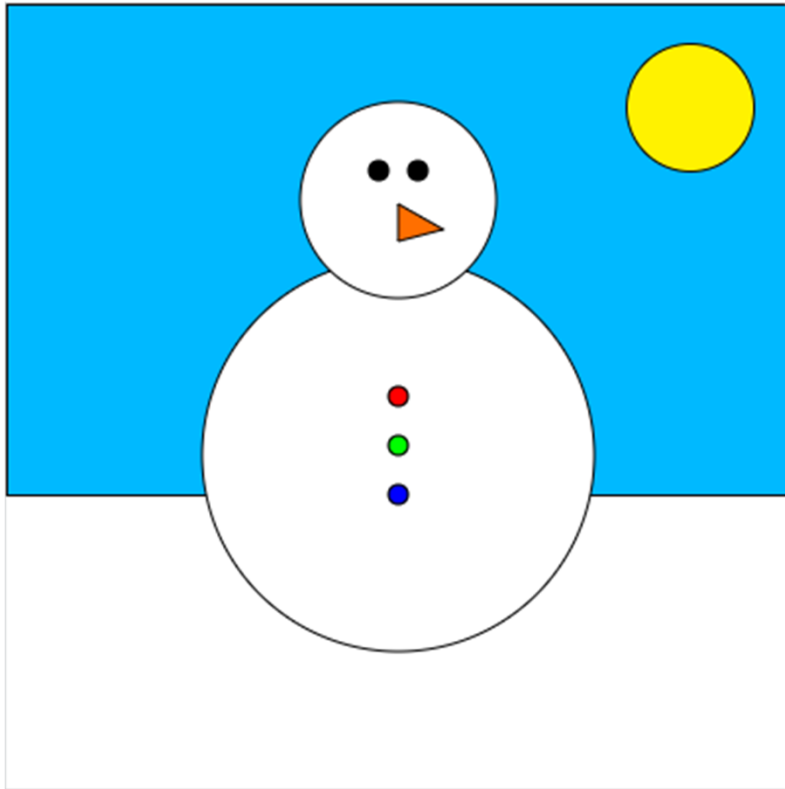
- Internet kleuren kan je makkelijkst maken met HEX kleuren
- Daarmee mix je **ROOD**, **GROEN** and **BLAUW** om een kleur te maken.
In het kort = RGB
- Je kan per kleur variëren van 0 tot 255.
- Bijvoorbeeld een rode rechthoek

```
// vierkant tekenen  
c.fillStyle = "rgb(255,0,0)";  
c.fillRect(200, 0, 60, 120);
```



R = 255, G = 0, B = 0





Een kleine oefening...

- Probeer deze sneeuwman in de zon te tekenen.
- Gebruik alles wat je tot nu toe geleerd hebt.
- Wees creatief en maak er je eigen versie van!

Herhalingen

- Herhalingen worden gebruikt om dezelfde code te herhalen.
- Bijv. 10 lijnen tekenen

```
// het tekendoek ophalen
var c = document.getElementById('canvas1').getContext('2d');

// lijnen tekenen
c.fillStyle = "#0000ff";
c.lineWidth = 5;

var i = 1;
while (i <= 10) {

    c.beginPath();
    c.moveTo(i * 10, i * 10); // Beginpunt
    c.lineTo(100, i * 10); // Eindpunt
    c.stroke();

    i = i + 1;

}
```

Bij start zet de waarde $i = 1$
Zolang i kleiner of gelijk is aan 10:

- Teken een lijn
- Verhoog i met 1

