

Week 1 - 2

Javascript Deel 1



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be



Inleiding

Wat is Javascript

- Derde webtaal die we gaan leren!
- Scripttaal om webpagina's interactief en dynamisch te maken.

Javascript gebruiken

- Formuliervalidatie
- Single-page applicatie
- Ajax-webapplicaties
- Aanpassingen aan de opmaak
- Aanpassingen aan de inhoud
- Animaties
- ...

Benodigde software

- Webstorm *(of een andere teksteditor)*
- Node.JS
- Google Chrome
- Google Chrome DevTools
- Git Bash
- Github

Starten met Javascript

Javascript toevoegen

Twee manieren om Javascript toe te voegen:

- We plaatsen het script integraal in de HTML-code
- We plaatsen het script in een apart bestand met de extensie .js

Integraal in de HTML-code

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="utf-8">
  <title>Dit is een webpagina met een script</title>
  <script>
    alert("Hello World - vanuit javascript");
  </script>
</head>
```


In een apart bestand

```
...  
</footer>  
<script src="apartbestand.js"></script>  
</body>
```

*Dit gaan wij **ALTIJD** doen!*

Javascript locatie

- Je kan het script eender waar toevoegen aan de HTML-code, maar *de locatie heeft invloed op de werking van het script.*
 - Plaats je het script helemaal bovenaan in de head, wordt eerst de JS-code en daarna pas de HTML- en CSS-code ingeladen.
 - Plaats je het script onderaan de body. Worden eerst de HTML- en CSS-code ingeladen en daarna pas de JS-code.
- *Alternatief om problemen te vermijden is het gebruik van 'defer' en 'async'!*

Defer

- De webbrowser moet niet wachten op het script en mag de HTML-pagina al verwerken.
- Het script laad in de achtergrond en wordt uitgevoerd zodra de HTML-pagina verwerkt is.
- Defer zorgt ervoor dat, ongeacht de locatie van het script, er geen blokkade is.

Async

- *Het script wordt asynchroon (tegelijkertijd) met de HTML-pagina gedownload en uitgevoerd.*
- Het script wacht niet op eventuele andere scripts of op de HTML-pagina. Het script start wanneer het is ingeladen.
- 'Async' maakt het script volledig onafhankelijk.

```
<script defer src="js/apartscriptbestand.js"></script>
```

```
<script async src="js/apartscriptbestand.js"></script>
```

Javascript output

Javascript output

Javascript kan op verschillende manieren data 'tonen'.

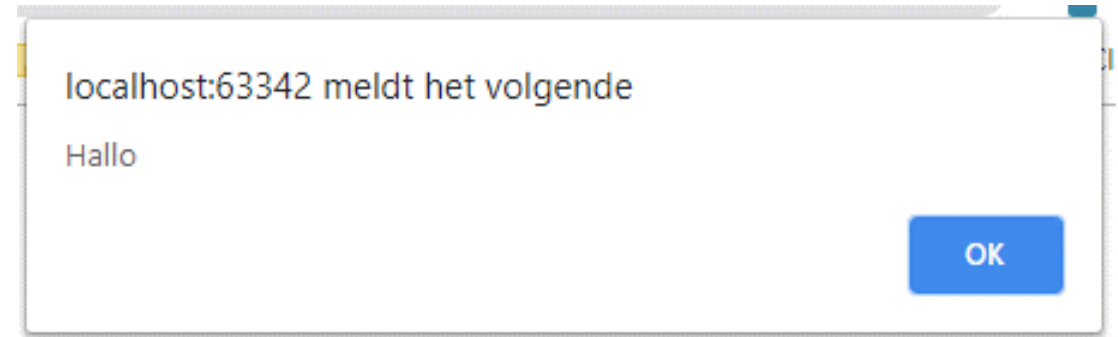
Drie manieren die we in het begin regelmatig gaan gebruiken:

- In een alert-box, via 'alert()'.
- In een browser console, via 'console.log()'.
- In een HTML-element, via 'innerHTML'.

Via alert()

Via de alert() methode kunnen we data weergeven in een alert-venster.

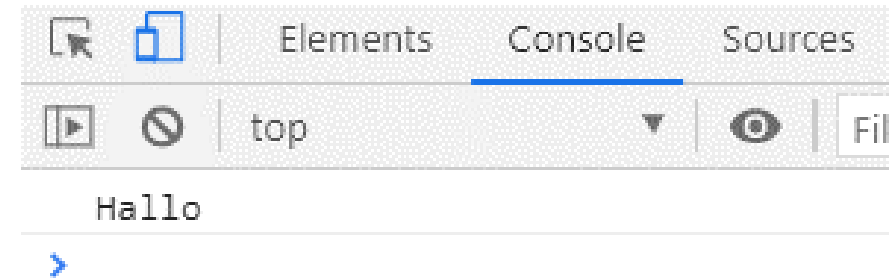
```
alert("Hallo");
```



Via console.log()

- De console.log() methode is ideaal om te debuggen!
- De data is niet meteen zichtbaar voor de bezoeker van de webpagina, maar is wel te raadplegen via de DevTools (tab 'Console').

```
console.log("Hallo");
```



Via innerHTML

Via 'innerHTML' kunnen we inhoud toevoegen aan een HTML-element.

```
<p id="inhoud"></p>
```

```
document.getElementById("inhoud").innerHTML = "Hallo";
```

Commentaar in Javascript

Twee manieren

We kunnen op twee manieren commentaar toevoegen:

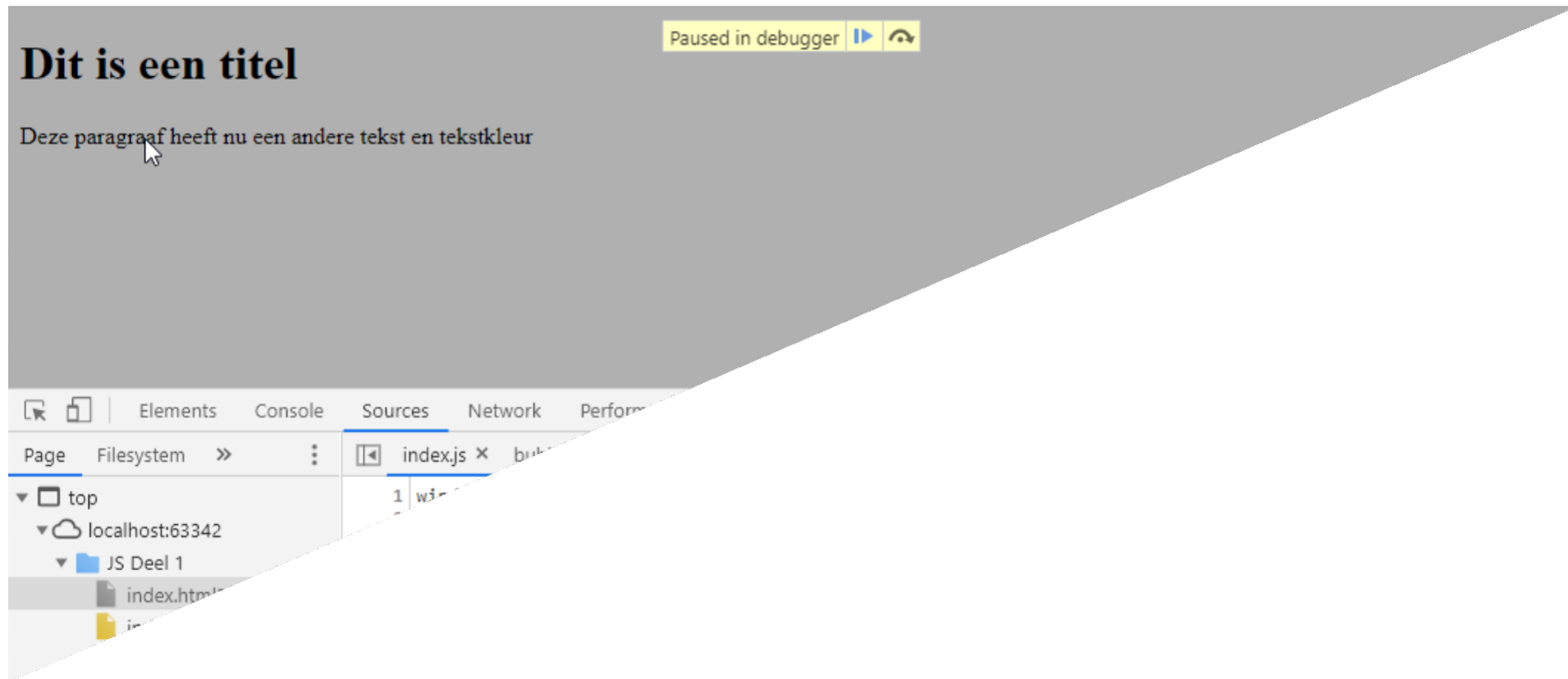
- Op één regel.
- Op meerdere regels.

```
// Dit is commentaar op één regel.
```

```
/* Dit is commentaar over  
meerdere regels heen */
```

Javascript Debuggen

In Google Chrome



Via Node.JS

