

Web Essentials





JAVASCRIPT

- Programmeertaal uitgevoerd door de browser
- Dynamische webpaginas
- Interactie



LINKEN NAAR EEN JAVASCRIPT-BESTAND VANUIT EEN HTML-PAGINA

```
<DOCTYPE html>
<html>
  <head>
    <title>Constructive & Co</title>
    <link rel="stylesheet" href="c01.css" />
  </head>
  <body>
    <h1>Constructive & Co</h1>
    <p>For all orders and enquiries,
    please call <em>555-3344</em></p>
    <script src="js/some-script.js"></script>
  </body>
</html>
```

JAVASCRIPT
WORDT
UITGEVOERD
WAAR HET WORDT
GEVONDEN IN DE
HTML

Wanneer de browser een
`<script>` element tegenkomt,
zal hij:

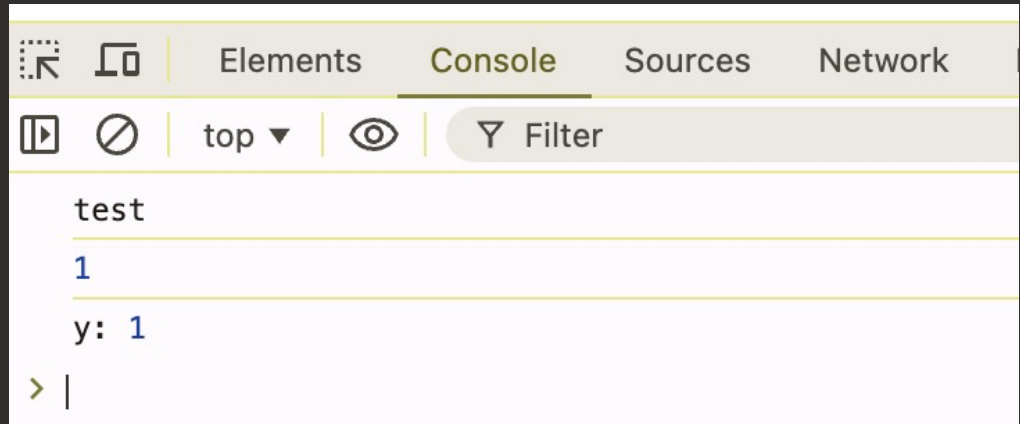
Stoppen om het script te laden
Controleren of hij een actie moet
uitvoeren

LOG NAAR DE CONSOLE

```
console.log('test')
```

```
let y = 1;  
console.log(y);
```

```
let x = 1  
console.log("y:", y)
```



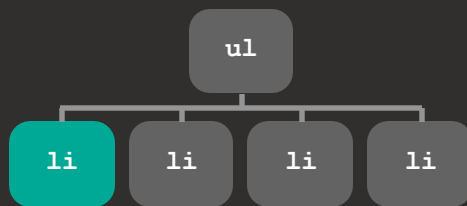


JAVASCRIPT
DOM

ELEMENTEN SELECTEREN & UPDATEN IN HET DOM

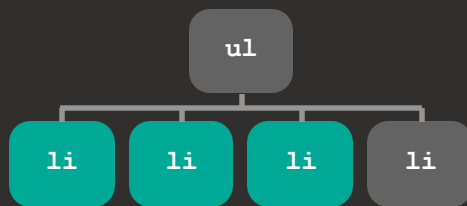
ELEMENTEN SELECTEREN

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



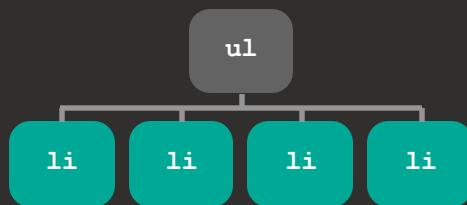
```
document.getElementById('one');
```

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



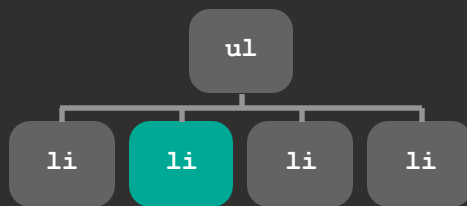
```
document.getElementsByClassName('hot');
```

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



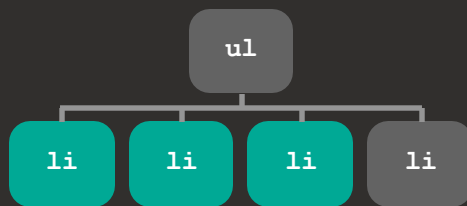
```
document.getElementsByTagName('li');
```

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



```
document.querySelector('#two');
```

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



```
document.querySelectorAll('li.hot');
```


HTMLCollections & NODELISTS

Als een DOM query meerdere elementen terug geeft, is het bekend als een **NodeList** of **HTMLCollection**.

Items in een NodeList of HTMLCollection zijn genummerd en geselecteerd **als** een array:

```
let elements;  
elements = document.getElementsByClassName('hot');  
let firstItem = elements[0];
```

Je kan controleren of er elementen in de List/Collection zitten voor je deze gebruikt

```
if (elements.length >= 1) {  
  
    let firstItem = elements[0];  
  
}
```

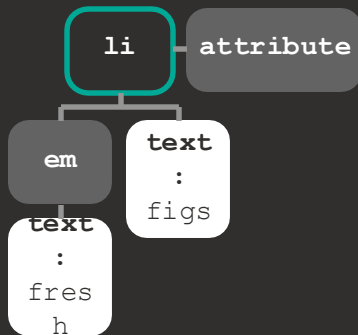
Loop over items

```
for (let i = 0; i < elements.length; i++)  
{  
  
    console.log(elements[i]);  
  
}
```

WERKEN MET ELEMENTEN

textContent verzamelt tekst inhoud

```
</li>  
<em>fresh</em>figs</li>
```

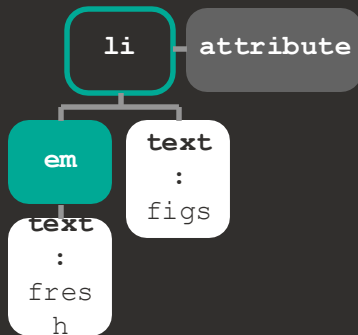


```
document.getElementById('one').textContent;
```

returns: fresh figs

innerHTML neemt de text en markup

`freshfigs`

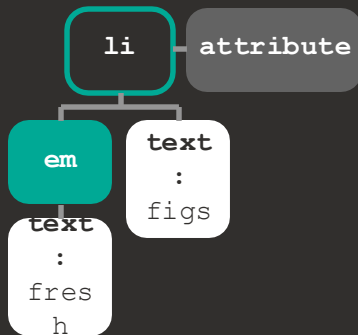


```
document.getElementById('one').innerHTML;
```

returns: `fresh figs`

Naast lezen kan je ook schrijven in het DOM

`freshfigs`



`<i>Spicy</i> Peppers`

```
let content =document.getElementById('one');
content.innerHTML = <i>Spicy</i> Peppers
```

Na uitvoering is de inhoud : `<i>Spicy</i> Peppers`

DOM MANIPULATIE

VS

innerHTML

```
createElement()  
createTextNode()  
appendChild()  
removeChild()
```

- Bouwt een string
- Bevat markup
- Update elementen

DOM MANIPULATIE

Voorbeeld DOM Manipulatie

```
let p = document.createElement('p');  
let text = document.createTextNode(p.textContent = "Dom manipulatie werkt");  
p.appendChild(text);  
div.appendChild(p);
```

```
let p = document.createElement('p');  
p.textContent = "Dom manipulatie werkt";  
div.appendChild(p);
```

WERKEN MET ATTRIBUTEN

TOEGANG TOT EEN ATTRIBUUT

1. Een DOM query gebruiken om element te selecteren:

```
let el = document.getElementById('one');
```

2. Een methode haalt een attribuut op van een element:

```
el.getAttribute('class');
```

ATTRIBUUT UPDATEN

Controleer op attribuut en update;

```
let el = document.getElementById('one');  
  
if (el.hasAttribute('class')) {  
    el.setAttribute('class', 'cool');  
}
```

ATTRIBUUT VERWIJDEREN

Controleer op attribuut en verwijder

```
let el = document.getElementById('one');  
  
if (el.hasAttribute('class')) {  
    el.removeAttribute('class');  
}
```

Manipuleren van classes

```
element.classList.add('cool')
```

```
element.classList.remove('boring');
```

```
element.classList.replace('boring', 'cool')
```

```
element.classList.toggle('hide-me');
```


EVENTS IN JAVASCRIPT

EVENTS IN JAVASCRIPT

- De gebruiker selecteert, klikt of houdt de cursor boven een bepaald element.
- De gebruiker kiest een toets op het toetsenbord.
- De gebruiker verkleint het browservenster of sluit het.
- Een webpagina wordt geladen.
- Een formulier wordt ingediend.
- Een video wordt afgespeeld, gepauzeerd of beëindigd.
- Er treedt een fout op
-

VERSCHILLENDE EVENT TYPES

USER INTERFACE EVENTS

load
unload
error
resize
scroll
...

KEYBOARD EVENTS

keydown
keyup
keypress

MOUSE EVENTS

click
dblclick
mousedown
mouseup
mouseover
mouseout

FOCUS EVENTS

focus / focusin
blur / focusout

MOUSE EVENTS

click
dblclick
mousedown
mouseup
mouseover
mouseout

EEN EVENT KOPPELEN AAN EEN ELEMENT

HTML EVENT HANDLER ATTRIBUUT

Verouderde code die we **NIET** meer gebruiken

```
<input type="text" id="username"  
      onblur="checkUsername()">
```



EVENT

Nee echt!
Niet gebruiken.

EVENT LISTENERS

Correcte manier die we **ALTIJD** gebruiken

```
el.addEventListener('blur', checkUsername, false);
```


ELEMENT

dat we in DOM
selecteren


EVENT


FUNCTION

Functie moet
aangeropen worden

PARAMETERS MET EVENT LISTENERS

```
el.addEventListener('blur', function() {  
    checkEmail(input);  
}, false);
```

Als 2e argument gebruiken we een anonieme functie.

ANONIEME FUNCTIE RECHSTREEKS IN EVENT LISTENER

```
el.addEventListener('click', function() {  
    // uit te voeren javascript code hier  
}, false);
```