



Webscripting

Hoofdstuk 15

Eventhandling

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be – www.pxl.be/facebook



Eventhandling

W3C Eventhandling

addEventListener:

voeg een eventlistener (function) aan een element
(of heel het window) toe

removeEventListener

verwijder een eventlistener

Nooit gebruik maken van inline eventhandling:

```
<a href="somewhere.html" onClick="alert('I've been clicked!')">
```

(lelijk: html en javascript door elkaar)



Eventhandling

```
<!doctype html>
<html>
  <head>
    <title>Events</title>
  </head>
  <body>
    <p>Click this document to activate the handler.</p>
    <script>
      window.addEventListener("click", () => {
        console.log("You knocked?");
      });
    </script>
  </body>
</html>
```

Eventhandler toegevoegd aan het scherm



Eventhandling

```
<!doctype html>
<html>
  <head>
    <title>Events</title>
  </head>
  <body>
    <button>Click me</button>
    <p>No handler here.</p>
    <script>
      let button = document.querySelector("button");
      button.addEventListener("click", () => {
        console.log("Button clicked.");
      });
    </script>
  </body>
</html>
```

Eventhandler toegevoegd aan 1e <button>



Event-object

Informatie over het type event

```
<!doctype html>
<html>
  <head>
    <title>Events</title>
  </head>
  <body>
    <button>Click me any way you want</button>
    <script>
      let button = document.querySelector("button");
      button.addEventListener("mousedown", event => {
        if (event.button == 0) {
          console.log("Left button");
        } else if (event.button == 1) {
          console.log("Middle button");
        } else if (event.button == 2) {
          console.log("Right button");
        }
      });
    </script> </body> </html>
```



Event-propagation

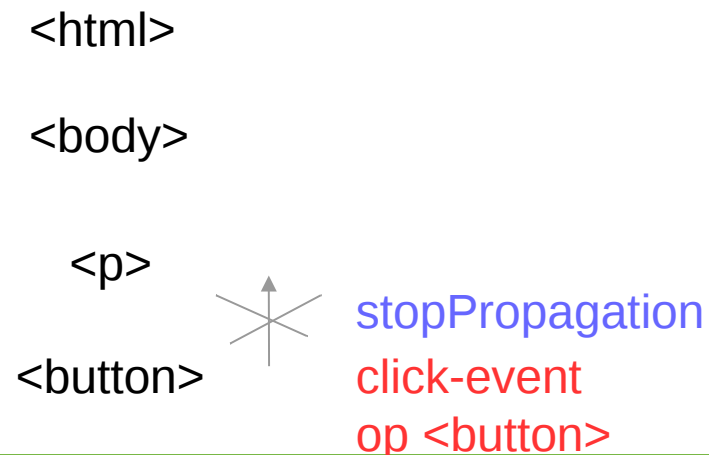
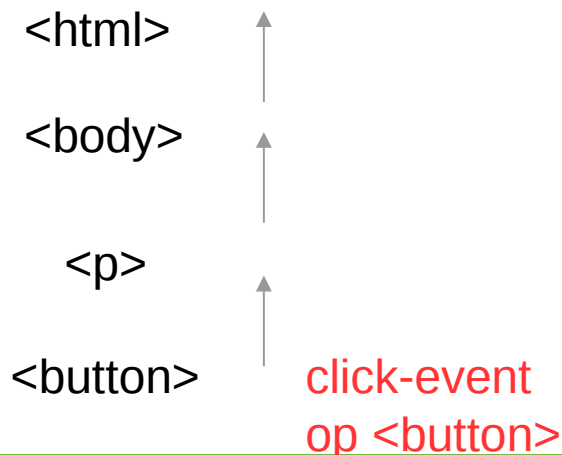
Meerdere eventlisteners in DOM-tree

Bijvoorbeeld listener naar click-events op <p> en op <button>

<p>A paragraph with a<button>button</button>.</p>

propagation (bubbling): event reist van laagste element naar boven
Eventhandler voor <button> wordt eerst behandeld dan die voor <p>

event.stopPropagation(): propagation wordt onderbroken



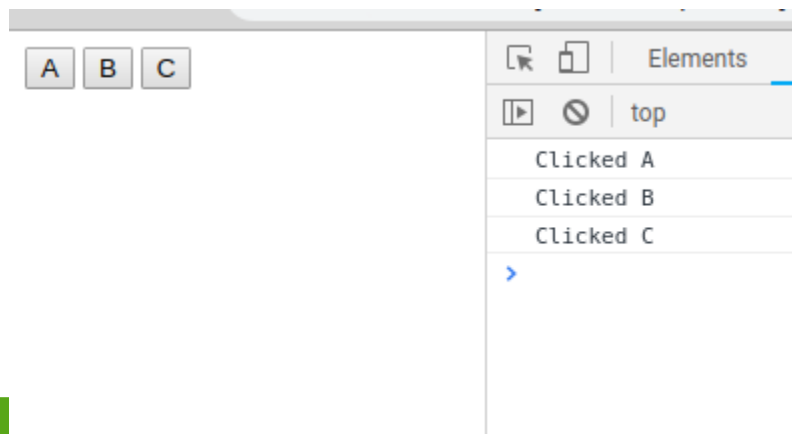
```
<!doctype html>
<html>
  <head>
    <title>My home page</title>
  </head>
  <body>
    <p>A paragraph with a
      <button>button</button>.
    </p>
    <script>
      let para = document.querySelector("p");
      let button = document.querySelector("button");
      para.addEventListener("mousedown", () => {
        console.log("Handler for paragraph.");
      });
      button.addEventListener("mousedown", event => {
        console.log("Handler for button.");
        if (event.button == 2) {
          event.stopPropagation();
        }
      });
    </script>
  </body>
</html>
```



Target

target: welk element triggerde het event?

```
<button>A</button>
<button>B</button>
<button>C</button>
<script>
  document.body.addEventListener("click", event => {
    if (event.target.nodeName == "BUTTON") {
      console.log("Clicked", event.target.textContent);
    }
  });
</script>
```



Event: target vs currentTarget

target: welke domnode veroorzaakte event?

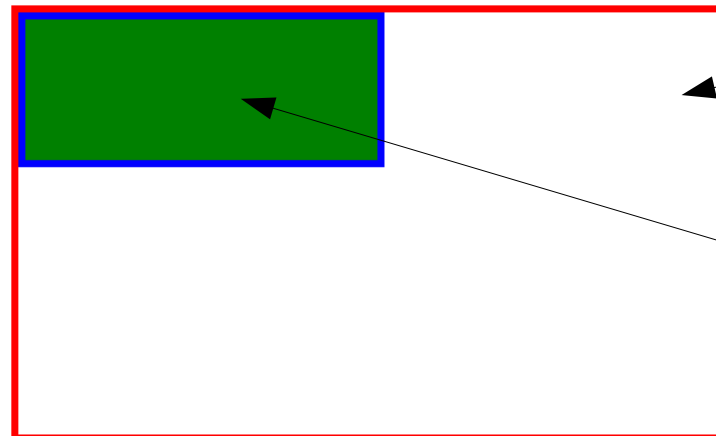
currentTarget: aan welke domnode werd eventlistener toegevoegd?

```
<div id="p">  
  <div id="c">  
  </div>  
</div>
```

```
window.addEventListener("load", handleWindowLoad);
```

```
function handleWindowLoad() {  
  let parent=document.getElementById("p");  
  parent.addEventListener("click", handleClick);  
}
```

```
function handleClick(event) {  
  event.target.style.backgroundColor="green";  
}
```



eventlistener
hier toegevoegd
currentTarget

hier geklikt:
target

Default actions

Veel elementen hebben een default action

click op <a>, default actie naar de link gaan

preventDefault: default actie wordt niet uitgevoerd

```
<a href="https://developer.mozilla.org/">MDN</a>
<script>
  let link = document.querySelector("a");
  link.addEventListener("click", event => {
    console.log("Nope.");
    event.preventDefault();
  });
</script>
```



Event: preventDefault

Doe de standaard-actie niet
submit -> procesform.php

loginname	<input type="text"/>
password	<input type="text"/>
submit	<input type="submit" value="Submit"/>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Form</title>
  <link rel="stylesheet" href="form.css">
</head>
<body>
<form action="procesform.php" method="post" id="form">
  <div class="row">
    <label for="loginname">loginname</label>
    <input type="text" id="loginname"></input>
  </div>
  <div class="row">
    <label for="password">password</label>
    <input type="text" id="password" name="password"></input>
  </div>
  <div class="row">
    <label for="submit">submit</label>
    <input type="submit" id="submit"></input>
  </div>
  <script type="text/javascript" src="form.js"></script>
</form>
</body>
</html>
```

Event: preventDefault

```
window.addEventListener("load", handleWindowLoad);

function handleWindowLoad() {
    let form=document.getElementById("form");
    form.addEventListener("submit", handleSubmit);
}

function handleSubmit(event) {
    let
inputLoginname=document.getElementById("loginname");
    let loginName=inputLoginname.value.trim();
    if(loginName.length < 5 ||
        !loginName.match(/^[a-zA-Z0-9_]+$/)) {
        inputLoginname.value="";
        event.preventDefault();
    }
}
```



Key events

keyup / keydown events
.key

```
<p>This page turns violet when you hold the V key.</p>
<script>
  window.addEventListener("keydown", event => {
    if (event.key == "v") {
      document.body.style.background = "violet";
    }
  });
  window.addEventListener("keyup", event => {
    if (event.key == "v") {
      document.body.style.background = "";
    }
  });
</script>
```



Focus event

Element krijgt focus (bijvoorbeeld klikken op text-input)

Blur: verliest focus

```
<p>Name: <input type="text" data-help="Your full name"></p>
<p>Age: <input type="text" data-help="Your age in years"></p>
<p id="help"></p>
<script>
  let help = document.querySelector("#help");
  let fields = document.querySelectorAll("input");
  for (let field of Array.from(fields)) {
    field.addEventListener("focus", event => {
      let text = event.target.getAttribute("data-help");
      help.textContent = text;
    });
    field.addEventListener("blur", event => {
      help.textContent = "";
    });
  }
</script>
```

Name:

Age:

Your full name

Load event

Tot nu toe:

<script> tag onderaan in html document
alle html-elementen zijn geladen voor het script uitgevoerd wordt
dus eventlisteners kunnen aan elementen toegevoegd worden

beter: wachten tot de pagina geladen is (load event) en dan
pas andere eventlisteners toevoegen



Load event

```
<!doctype html>
<html>
  <head>
    <title>Events</title>
  </head>
  <body>
    <button>click</button>
    <script>
      const handleLoad = () => {
        let button = document.querySelector("button");
        button.addEventListener("click",handleClick );
      }
      const handleClick = (event) => {
        console.log(event.target.textContent);
      }
      window.addEventListener("load", handleLoad);
    </script>
  </body>
</html>
```



Besluit

W3C eventhandling

addEventListener / removeEventListener

verschillende soorten eventlistener: click, focus, ...

Event-object

- target welk element triggerde het event

- stopPropagation

- preventDefault

