JavaScript in webpagina's

Hoe toevoegen?

- Script-tag
- In header: op te roepen functies, initialisatie

```
<header>
    <script src="bestandsnaam.js"></script>
</header>
```

- In body: onmiddellijk uit te voeren opdrachten

```
<body>
    <script>
        // JavaScript code
    </script>
</body>
```

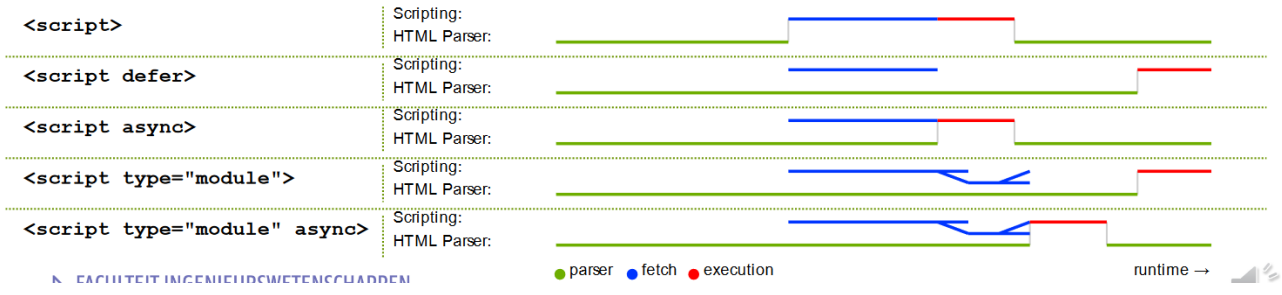
Uitvoeren JavaScript in HTML

- Laden HTML:
    - Browser stopt het laden bij script-tag
    - Script wordt opgehaald en uitgevoerd
    - Browser gaat verder met het laden van de pagina

```
<!-- Script asynchroon laden en daarna onmiddellijk uitvoeren -->
<script async src="timer.js"></script>
<!-- Script pas uitvoeren nadat de paginastructuur (=DOM-boom) opgesteld is -->
<script defer src="timer.js"></script>
<!-- Modules -> standaard defer -->
<script src="timer.js" type="module"></script>
```

Modules gebruiken in een webpagina

- Standaard pas uitgevoerd na het parsen van de pagina (=defer)



HTML DOM

- Webpagina
  - Boom van objecten
  - Boom bestaat uit knopen
- Elementen selecteren

```
document.getElementById(...);
document.getElementsByClassName(...);
document.getElementsByName(...);
document.getElementsByTagName(...);
// CSS-selector
const alineas = document.querySelectorAll("p");
const eersteAlinea = document.querySelector("p");
```

- Toegang krijgen tot attributen

```
const exampleAttribute = div1.getAttribute("id");
exampleAttribute = div1.id;
```

- Tekst
  .textContent
- Invoervelden
  .value

Documentstructuur aanpassen

```
document.createAttribute(...);
document.createComment(...);
document.createElement(...);
document.createTextNode(...);
element.removeChild(...);
element.replaceChild(newChild, oldChild);
element.appendChild(...);
document.appendChild(...);
```

Events

- Drie opties:
  - Attribuut toevoegen in HTML - deprecated
  - Eventhandler toevoegen in JavaScript
    - Met property
      knop.onclick = functie
    - Via functie
      knop.addEventListener("click", functie);

Event bubbling en event capture

- Volgorde waarin eventhandlers uitgevoerd worden
- Event bubbling
    - Standaard
    - Eerst event "meest binnenste element"
- Event capture
    - Eerst event "meest buitenste element"
    - addEventListener(event, function, useCapture);

Canceling events

event.preventDefault(): standaardactie voor event wordt niet uitgevoerd
event.stopImmediatePropagation(): andere eventhandlers op het element worden niet uitgevoerd en het event bubbelt niet verder naar (voor)ouderelementen
event.stopPropagation(): zorgt ervoor dat het event niet verder bubbelt naar (voor)ouderelementen

Timer

- Eén keer uitvoeren: setTimeout(functie, tijdsinterval in ms);
- Herhaaldelijk uitvoeren: setInterval(functie, tijdsinterval in ms);

Canvas

- Gebied om op te tekenen in browser
- Tekenen in script (JavaScript)

Vormen tekenen

- Rechthoeken
    - fillRect(x,y,width,height)
    - strokeRect(x,y,width,height)
    - clearRect(x,y,width,height)
- Lijnen en curves
    - beginPath()
    - moveTo(x, y)
    - lineTo(x, y)
    - arc(x, y, radius, startAngle, endAngle, anticlockwise)
    - closePath()
    - fill()
    - stroke()

Canvas versus SVG

- SVG
    - Figuur in XML-formaat
    - Elke vorm is een object
        - Onafhankelijke resolutie
- Canvas

    - Tekening wordt niet onthouden
    - Betere performantie indien veel "objecten"

Webstorage

- Info bewaren op client
- Twee opties
    - localStorage
        - Permanent
    - sessionStorage
        - Zolang tabblad niet gesloten wordt

Data bewaren

```
let gameData = {
    playerName: "Jan",
    levelCompleted: 1,
    score: 10
};
// Opslaan
let gameDataJSON = JSON.stringify(gameData);
localStorage.setItem("gameData", gameDataJSON);
// Ophalen
loadedData = localStorage.getItem("gameDate");
let data = JSON.parse(loadedData);
```