

JavaScript in webpagina's

Veerle Ongenae



Doel

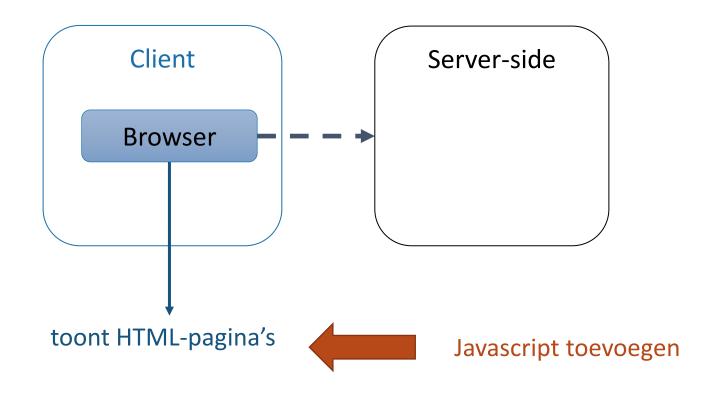


- Hoe gebruik je javascript in webpagina's?
- Wat is HTML DOM?
- Hoe werkt het eventmechanisme?
- Verschillende javascript API's in webpagina's kunnen gebruiken



Inhoud







JavaScript in webpagina's



- Javascript toevoegen aan een webpagina





Javascript toevoegen aan HTML



- Script-tag

```
<script src="timer.js"></script>
```

```
<script>
… // javascript code
</script>
```

- Waar?
 - Header
 - ➤ Op te roepen functies
 - Body
 - ➤ Onmiddellijk uit te voeren opdrachten



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="speelgoed.js" type="module"></script>
```



Modules gebruiken in een webpagina



- Modules

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

<script src="speelgoed.js" type="module"></script>

<script nomodule defer src="fallback.js"></script>

Scrinting

oudere browsers

► FACUITEIT INGENIEURSWETENSCHAP	PFN	parser fetch execution	runtime →
<pre><script async="" type="module"></pre></th><th>Scripting: HTML Parser:</th><th></th><th></th></tr><tr><td><pre><script type="module"></pre></td><td>Scripting: HTML Parser:</td><td></td><td></td></tr><tr><td><script async></td><td>Scripting: HTML Parser:</td><td></td><td></td></tr><tr><td><script defer></td><td>Scripting: HTML Parser:</td><td></td><td></td></tr><tr><td><script></td><td>HTML Parser:</td><td></td><td></td></tr></tbody></table></script></pre>			

JavaScript in webpagina's



- Javascript toevoegen aan een webpagina
- HTML DOM API





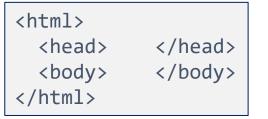
HTML DOM



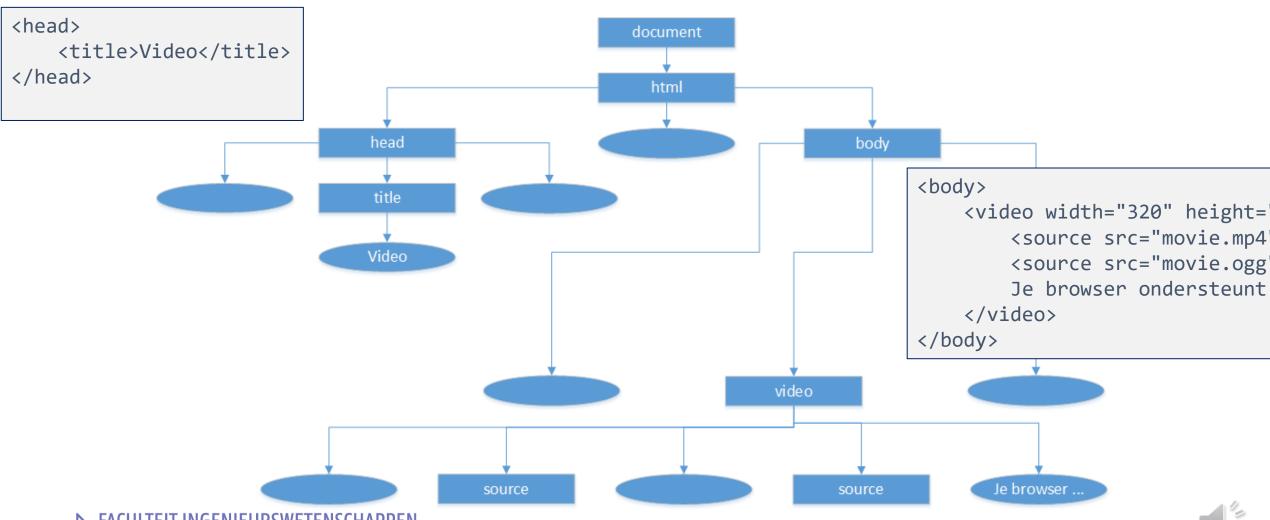
- Webpagina
 - Boom van objecten
- Document Object Model (DOM)
 - Model
 - API manipulatie webpagina
- Deel van DOM
 - Specificatie W3C
 - DOM level 3

HTML DOM

EN ARCHITECTUUR







HTML DOM API



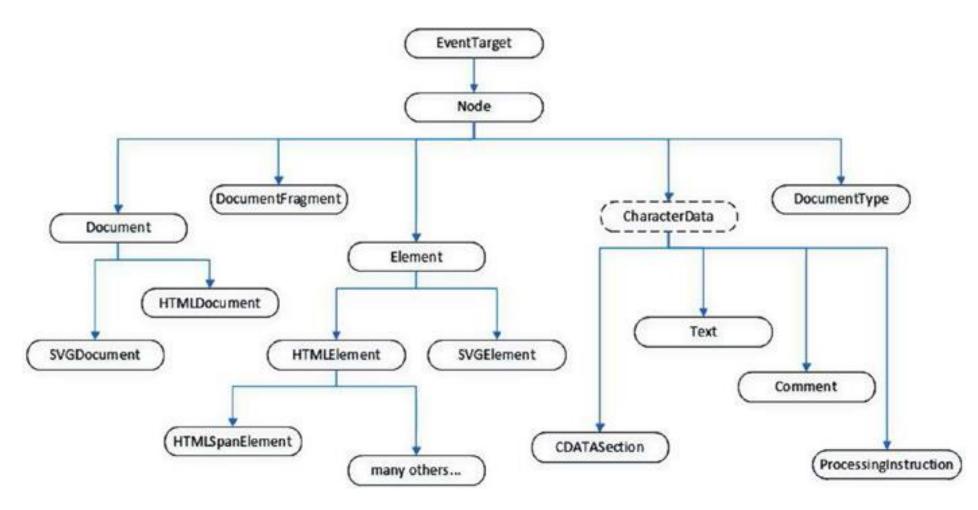
- Boom bestaat uit knopen (node)
 - Document zelf (document)
 - Elementen
 - Attributen
 - Tekst
 - ...







DOM - nodes







Toegang tot HTML – elementen selecteren

webpagina-object

```
document.getElementById(...)
document.getElementsByClassName(...)
document.getElementsByName(...)
document.getElementsByTagName(...)
```

```
const alineas = document.querySelectorAll("p");
```

CSS-selector

```
const eersteAlinea = document.querySelector("p");
```







```
HTML
```

```
<div id="div1">Hi Champ!</div>
```

JS

```
const div1 = document.getElementById('div1'); //=> <div id="div1">Hi Champ!</div>
const exampleAttr= div1.getAttribute('id'); //=> "div1"
```

waarde attribuut opvragen na

naam attribuut

ALTERNATIEF

element.attrNaam

JS

```
const div1 = document.getElementById('div1'); //=> <div id="div1">Hi Champ!</div>
div1.setAttribute('id', 'ID1'); // HTML aangepast <div id="ID1">Hi Champ!</div>
```

waarde attribuut aanpassen naam attribuut nieuwe waarde









```
HTML
```

```
<div id="divA">This is <span>some</span> text!</div>
```

JS

```
let tekst = document.getElementById('divA').textContent; //=> This is some text!
```

tekstinhoud van een element

```
document.getElementById('divA').textContent = 'This text is different!';
// HTML aangepast <div id="divA">This text is different!</div>
```









HTML

```
<input type="text" name="naam" size="20">
<textarea id="textOutput" ...> ... </textarea>
```

JS

waarde van een invoerveld

```
let invoer = document.getElementsByName('naam')[0].value; // inhoud
tekstvak
document.getElementById('textOutput').value = invoer;
```





nieuwe knopen maken

```
document.createAttribute(...)
document.createComment(...)
document.createElement(...)
document.createTextNode(...)
```

knopen toevoegen aan boomstructuur

```
element.appendChild(...)
document.appendChild(...)
```

knopen verwijderen

```
element.removeChild(...)
element.replaceChild(newChild, oldChild)
```

knopen vervangen



Voorbeeld

Kies dag: maandag •

HTML



- Lijst opvullen met dagen van de week

JS

```
<html>
  <head>
    <script src="js/dagenWeek.js">
    </script>
  </head>
  <body>
    Kies dag:
    <select id="keuzeDag"></select>
    <script>
      vulLijst();
    </script>
  </body>
</html>
```

JavaScript in webpagina's



- Javascript toevoegen aan een webpagina
- HTML DOM API
- Events





Events



- Browser
 - Laden documenten, figuren
- Gebruiker
 - Muis
 - Toetsenbord
- Attributen
 - onload, onunload
 - onfocus, onchange, onblur
 - onkeydown, onkeyup
 - onmouseover, onmousemove, onmouseout
 - onsubmit
 - onclick
 - ...





Afhandelen events



- Reageren op events: drie opties
 - Attribuut toevoegen in HTML deprecated
 - Eventhandler toevoegen in JavaScript
 - ➤ Met property

```
<button id="knop">...
```

```
function doeIets() {...}

let knop = document.getElementById("knop");
knop.onclick = doeIets;
```

event





eventhandler

Afhandelen events



- Drie opties
 - Eventhandler toevoegen in JavaScript
 - ➤ Via functie
 - Meerdere functies mogelijk
 - Events kunnen ook verwijderd worden
 - Kan ook rechtstreeks aan document, ...

```
<button id="knop">...
```

```
function doeIets() {...}

let knop = document.getElementById("knop");
knop.addEventListener("click", doeIets);
```

event



Voorbeeld





```
<script src="js/som.js" defer></script>
function init() {
                                                   <form>
   // events aan tekstvakken
                                                     <input id="getal1" type="text"> +
   let getal1 = document.getElementById("getal1");
                                                     <input id="getal2" type="text"> =
   let getal2 = document.getElementById("getal2");
                                                     <input id="som" type="text" readonly>
   getal1.onchange=berekenSom;
   getal2.onchange=berekenSom;
                                                   </form>
function berekenSom() {
   let getal1 = document.getElementById("getal1");
   let getal2 = document.getElementById("getal2");
   let som = parseInt(getal1.value) + parseInt(getal2.value);
   let somTekst = document.getElementById("som");
   somTekst.value = som;
init();
```

Event bubbling en event capture



- Volgorde waarin eventhandlers uitgevoerd worden
- Event bubbling
 - Standaard
 - Eerst event "meest binnenste element"

```
click-event button click-event p
```

- Event capture
 - Eerst event "meest buitenste element"

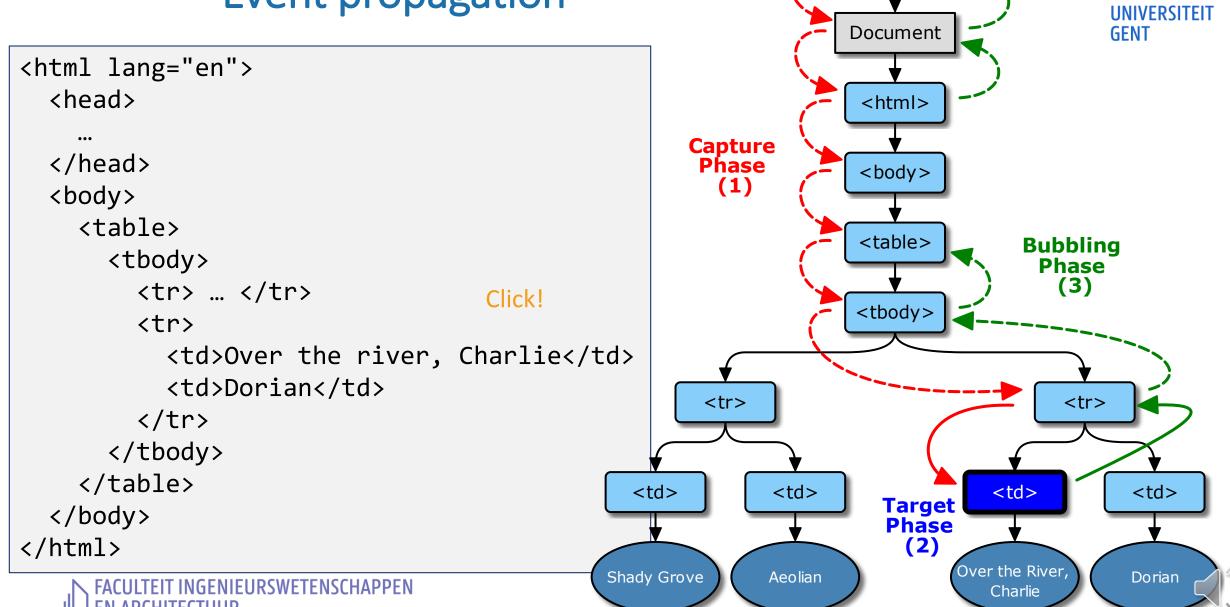
```
click-event p click-event button
```

addEventListener(event, function, useCapture);

```
<button id="knop">...</button>
```



Event propagation



Window

Event-object



- Wordt meegegeven met eventhandler
- Voorzie een parameter voor eventhandler
- Eigenschappen
 - type
 - target
 - currentTarget
 - bubbles
 - •

```
function schrijf(event) {
    ...
}
```

Canceling events



Naam methode	Omschrijving
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





Validatie



- Controle invoer gebruiker
- Formulier wordt niet ingediend indien controle mislukt
 - Event is "cancelable": standaardactie (indienen formulier) kan stopgezet worden
 - > onsubmit, onclick
 - Eventhandler: logische functie



JavaScript in webpagina's



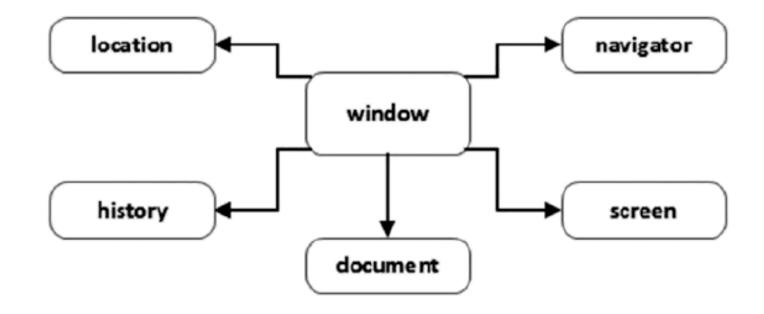
- Javascript toevoegen aan een webpagina
- HTML DOM API
- Events
- Browser model



Browser Object Model



- Geen officiële standaard









één keer een functie uitvoeren

```
// timer instellen
let timer = setTimeout(fct,tijdsinterval);
...
// timer uitschakelen
clearTimeout(timer);
```

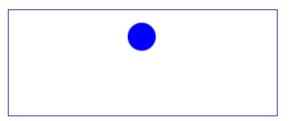
herhaaldelijk een functie uitvoeren

```
// timer instellen
let timer = setInterval(fct,tijdsinterval);
...
// timer uitschakelen
clearInterval(timer);
```

methodes window-object
uit te voeren functie
tijdsinterval in ms
id timer



Voorbeeld







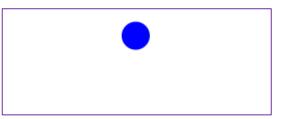
HTML

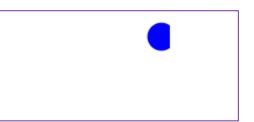
- Blauwe cirkel die over scherm schuift

functiedeclaratie

uitvoeren functie

Voorbeeld







animatieCirkel.js

```
export default function () {
   let x = 0;
   let timer = setInterval(tekenCirkel, 100);
   function tekenCirkel() {
        // x verhogen
        x = x + 1;
       // einde
       if (x >= 200) {
            clearInterval(timer);
        //svg-cirkel
        let cirkel = document.getElementById("cirkel");
        cirkel.setAttribute("cx", x);
```

startAnimatie.js

```
import animatieCirkel from "./animatieCirkel.js";
animatieCirkel();
```







```
window.alert("Naam invoeren!");
ALTERNATIEF
alert("Naam invoeren!");
```

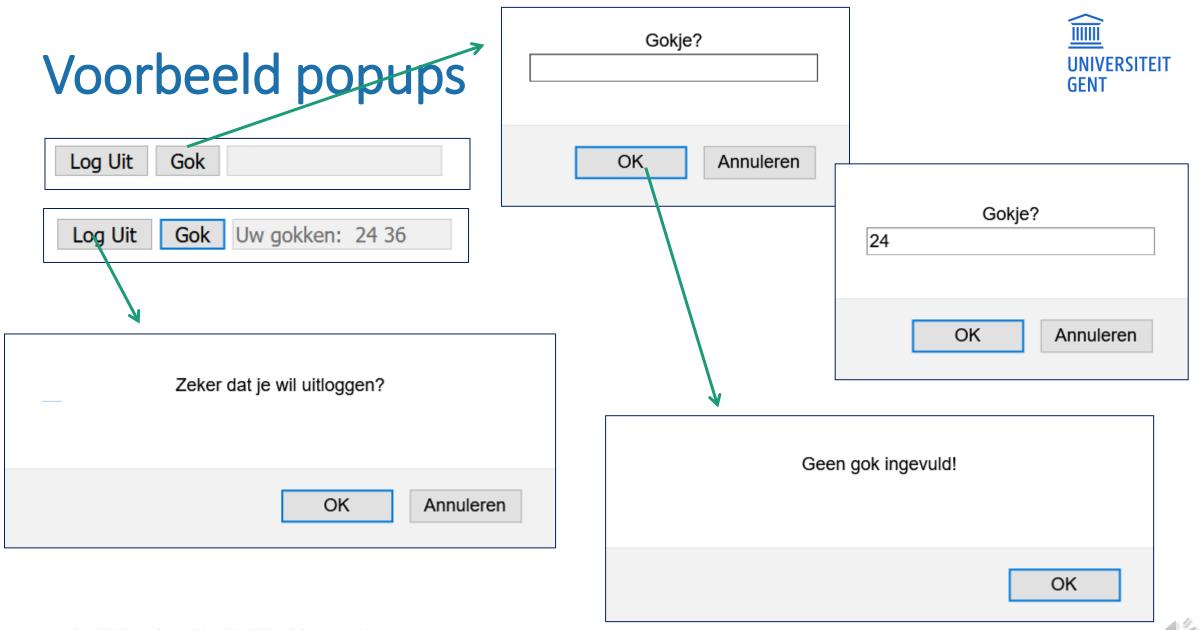
```
let antwoord = confirm("...");
    true/false
```

```
let invoer = prompt("...");
ingevoerde tekst
```



Naam invoeren!





Voorbeeld popups





functiedeclaratie

uitvoeren functie

HTML

Voorbeeld popups

Log Uit Gok

```
export default function () {
    let gokken = "Uw gokken: ";
    let knopLogUit = document.getElementById("loguit");
    knopLogUit.onclick = bevestigUitloggen;
    let gokKnop = document.getElementById("gok");
   gokKnop.onclick = gok;
   function bevestigUitloggen() {
        let uitloggen = confirm("Zeker dat je wil uitloggen?");
        return uitloggen;
   function gok() {
       let gokje = prompt("Gokje?");
        if (gokje !== null && gokje !== "") {
            gokken += " " + gokje;
            document.getElementById("overzicht").value = gokken;
        } else {
            alert("Geen gok ingevuld!");
```

```
import gokjeNieuw from "./gokje.js";
gokjeNieuw();
```





- Javascript toevoegen aan een webpagina
- HTML DOM API
- Events
- Browser model
- Canvas API







Nieuwe API's sinds HTML5









Audio/video API



Drag and drop



History/Backup



Editeren



Offline



Data-opslag



Geolocatie



Cross-document messaging



Canvas





- Gebied om op te tekenen in browser

HTML afmetingen canvas

```
<canvas id="myCanvas" width="200" height="100">
Your browser does not support the canvas element.
</canvas>
```

- Tekenen in script (JavaScript)

id canvas

```
let c=document.getElementById("myCanvas");
let cxt=c.getContext("2d");
cxt.fillStyle="#FF0000";
cxt.beginPath();
cxt.arc(70,18,15,0,Math.PI*2,true);
cxt.closePath();
cxt.fill();
```

cirkelboog

- middelpunt (70,18)
- straal 15
- van 0 tot 2π
- tegenwijzerszin





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()



```
cxt.beginPath();
cxt.moveTo(50, 150);
cxt.lineTo(80, 180);
cxt.arc(80, 180, 20, 0.5, 2, false);
cxt.closePath();
cxt.fill();
cxt.stroke();
```

Opmaak



0: transparant

1: niet transparant

- Afbeeldingen
 - drawImage(image, x, y)
- Kleur
 - Eigenschappen
 - ➤ fillStyle
 - > strokeStyle
 - Waarden
 - > CSS-kleur: "#RRGGBB", rgb(red, green, blue), rgba(red, green, blue, alpha), ...
 - **→** Gradient
 - > Patroon





Opmaak



- Tekst
 - font cxt.font="30px Verdana";
 - textAlign
 - textBaseline
 - fillText(text, x, y [, maxWidth])
 - strokeText(text, x, y [, maxWidth])

Hallo

Hallo



Canvas



- Toestand
 - save()
 - restore()
- Canvas vervormen
 - translate(x, y)
 - rotate(angle)
 - scale(x, y)
 - transform(m11, m12, m21, m22, dx, dy)
 - setTransform(m11, m12, m21, m22, dx, dy)



Meer tekenen



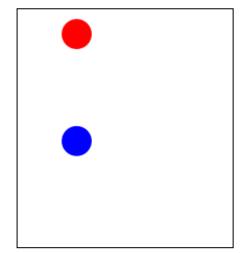
- Bezierkrommen
- Kwadratische krommen
- Gradiënten
- Patronen
- Overlappende vormen

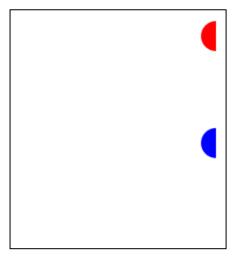


Voorbeeld



- Rode (canvas) en blauwe cirkel (svg) die over scherm schuift
 - cirkels.html





Canvas versus SVG



- SVG
 - Scalable Vector Graphics
 - Figuur in XML-formaat
 - Elke vorm is een object
 - ➤ Onafhankelijk resolutie

- Canvas

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

```
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
    style="fill:grey;stroke:maroon;stroke-width:5;fill-rule:evenodd;" />
    <rect x="0" y="220" height="100" width="200" />
    </svg>
```







- Javascript toevoegen aan een webpagina
- HTML DOM API
- Events
- Browser model
- Canvas API
- WebStorage API









Webstorage



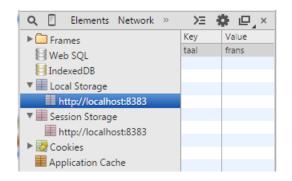


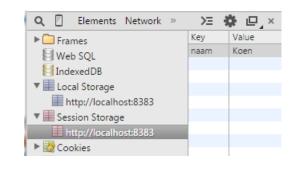
- Info bewaren op client
- Twee opties
 - localStorage
 - **>** Permanent

```
localStorage.setItem("taal","frans");
let taal = localStorage.getItem("taal");
```

- sessionStorage
 - ➤ Zolang tabblad niet gesloten wordt

```
sessionStorage.setItem("naam", "Koen");
let naam = sessionStorage.getItem("naam");
```





```
localStorage.taal = "frans";
let taal = localStorage.taal;
```

```
sessionStorage.naam = "Koen";
let naam= sessionStorage.naam;
```

- Naam/waarde-paren (strings)



Data bewaren



```
let gameData = {
    playerName: "Rex",
    levelCompleted: 5,
    score: 84,
    items: ["hat", "umbrella", "katana"]
};
let gameDataJSON = JSON.stringify(gameData);
localStorage.setItem("gameData", gameDataJSON);
loadedData = localStorage.getItem("gameData");
let data = JSON.parse(loadedData);
```



- Javascript toevoegen aan een webpagina
- HTML DOM API
- Events
- Browser model
- Canvas API
- WebStorage API
- JSON









JSON: Tekstuele voorstelling javascript-object (serialisatie)



```
"closet": {
   "id": 0,
   "description": "A dark coat closet.",
   "light": { "on": false },
   "contents": [],
   "exits": ["east"]
"livingRoom": {
   "id": 1,
   "description": "A living room in an old, rambling house.",
    "light": {
        "on": true
    "contents": ["fireplace", "sofa", "dagger"],
   "exits": ["west", "north", "south"]
```





```
fetch('json/rooms.json')
   .then(response => response.json())
   .then(json => {
      rooms = json;
   });
```

```
stuurt HTTP-bericht (GET)

antwoord verwerken

antwoord (tekst) → js-object (deserialiseren)

object bewaren
```





- Javascript toevoegen aan een webpagina
- HTML DOM API
- Events
- Browser model
- Canvas API
- WebStorage API
- JSON
- Frameworks, tools, ...









Frameworks, tools, ...



- Meeste browser hebben ingebouwde debug-tool (F12)
- JSLint: controle JS-code
- JavaScript-bibliotheken
 - JQuery
 - Angular
 - React
 - D3.js C3.js
 - •
- TypeScript: Programmeertaal uitbreiding Javascript
- Built tools: Grunt, Gulp
- Testing tools: Mocha&Chai, Karma
- Console/serverside: node.js





Doel



- Hoe gebruik je javascript in webpagina's?
- Wat is HTML DOM?
- Hoe werkt het eventmechanisme?
- Verschillende javascript API's in webpagina's kunnen gebruiken

