

Samenvatting Web Development

Functie JS

- Elementen toevoegen/ verwijderen
- CSS-classes of CSS-properties van een element wijzigen
- HTTP-requests versturen op de achtergrond en de response verwerken

Waar implementeren we JS-files?

Net onder de body tag

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles/style.css">

  <title>demo</title>
</head>
<body>

<script type="text/javascript" charset="utf-8"
        src="scripts/code.js"></script>
</body>
</html>
```

Globale vs lokale variabelen

Lokale variabelen in een functie, globale niet (zo weinig mogelijk gebruiken)

Enkel let gebruiker, var

Standaardcode js bestand

```
const setup = () => { // deze code wordt pas uitgevoerd als de pagina volledig is ingeladen }
window.addEventListener("load", setup);
```

Array aanmaken

```
let leeg = [];

let teksten = ["een","twee","drie","twee"];

let getallen = [1, 2, 3];

let gemixt = [true, "hallo", 4];           //verschillende elementen opvragen
```

Lengte opvragen

```
console.log(teksten.length)           //toont 4
```

Individuele elementen opvragen

```
console.log(teksten[0]);               //toont "een"
```

For-loop array

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

`indexOf(element, index)` *index parameter is optioneel*

```
console.log(teksten.indexOf('twee')); //toont 1, neemt standaard het eerste item hiervan
```

```
console.log(teksten.indexOf('twee',2)); //toont 3, begint vanaf index 2
```

```
console.log(teksten.indexOf('vier')); //toont -1, item zit niet in de array
```

`lastIndexOf(element,index)` *index parameter is optioneel*

```
console.log(teksten.lastIndexOf('twee')); //toont 3, neemt standaard het laatste item hiervan
```

`push(element)` *geeft de nieuwe lengte terug*

```
getallen.push(4); //voegt 4 toe aan het einde van de array getallen
```

`pop()` *geeft het verwijderde item terug*

```
getallen.pop() // toont 4, verwijdert het laatste item
```

`unshift(element)` *geeft de nieuwe lengte terug*

```
getallen.unshift(9) // [9,1,2,3] item toevoegen aan het begin
```

`shift()` *geeft het verwijderde item terug*

```
getallen.shift() //9 wordt terug verwijderd, eerste item in array
```

`slice(index1, index2)` *geeft nieuwe array terug vanaf index 1 tot index2 (excl)*

```
teksten.slice(1,3) //toont "twee","drie"
```

`join(scheidingsteken)` *scheidingsteken is optioneel, geeft String terug*

```
teksten.join() //toont "een,twee,drie,twee", komma standaard
```

```
teksten.join("") //toont "eentweedrie"
```

```
teksten.join("-") //toont "een-twee-drie"
```

`popups` *voor ontwikkelaars, niet voor eindgebruikers*

```
window.alert("dit is een mededeling"); //mededeling met enkel OK button
```

```
window.confirm("Weet u het zeker?"); //vraag in popup met ok en cancel button
```

```
window.prompt("Wat is uw naam", "onbekend") //zelfde met default waarde & input
```

Nodes opvragen

document.getElementById("abc");	
document.getElementsByClassName("xyz");	<i>//array met alle classes xyz</i>
document.getElementsByTagName("img");	<i>//array met alle images</i>
document.querySelector("p");	<i>//enkel eerste paragraaf</i>
document.querySelectorAll("p");	<i>//array met alle paragrafen</i>

inhoud opvragen/wegschrijven naar html *pas doen na node is opgevraagd*

txtOutput.innerHTML = "Welkom"	<i>//zowel HTML als tekst content</i>
txtOutput.innerText = "Welkom"	<i>//enkel de tekst van het element</i>
txtOutput.textContent	<i>//returnt content zoals ze in de HTML-pagina staat</i>
ook afstammelingen	

Tekstvelden uitlezen

document.getElementById("email").value	<i>value NIET vergeten, ALTIJD STRING</i>
--	---

EventListeners *document.addEventListener("event", functie)*

click	<i>van zodra gebruiker klikt op element</i>
submit	<i>van zodra formulier is verzonden</i>
mouseover	<i>van zodra pointer over element gaat</i>
change	<i>van zodra value van input element veranderd</i>

Huidig element opvragen via event & target

const testje = (event) =>{ const targetElement = event.target; }
--

Substring vs slice *slice beginnen tellen van achter naar voor*

let str = "Hello, world!"	
console.log(str.substring(0,5));	<i>Hello</i>
console.log(str.slice(0,5));	<i>Hello</i>
console.log(str.substring(-3,5));	<i>Hello</i>
console.log(str.slice(-3,5));	<i>"" , telt vanaf l tot komma & gaat niet</i>
console.log(str.substring(5,0));	<i>Hello, zelfde als str.substring(0,5)</i>
console.log(str.slice(5,0));	<i>"" , kan niet van achter naar voor tellen</i>

CSS-properties van element instellen

elementNode.style.backgroundColor="red";	//in css is dit background-color: red;
--	--

Classnames instellen van elementen

txtOutput.className = "invalid";	//1 klasse instellen
txtOutput.className = "";	//alle klassen verwijderen
txtOutput.classList.add("groen");	//een klasse toevoegen
txtOutput.classList.remove("groen");	//een klasse verwijderen
txtOutput.classList.toggle("enable");	//een klasse togglen (aan/uit)

Numbers in javascript

let getalAlsNumber = parseInt(getalAlsTekst,10);	//10 → decimaal, omzetten naar Number
let getalAlsTekst = getalAlsNumber.toString();	//omzetten naar String
let getalAfgerond = getalAlsNumber.toFixed(2)	//afronden tot 2 cijfers na de komma
let getalAfgerond = getalAlsTekst.toFixed(2)	//kan ook, produceert toch String default

GET vs POST

GET *meegestuurd data niet toevoegen op server, via adresbalk pagina opvragen*

POST *data wordt op server bewaard*

Name attribuut

Gebruikt bij doorsturen van data

Label element

Betekenis aanduiden van input element, verplicht

Hidden input velden

Handig voor code langs de serverkant

<input type="hidden" name="productCode" value="12345">
--

Radiobutton

Groep selecteren door elke radiobutton in die groep dezelfde name toe te kennen

Submitbuttons

Altijd een name attribuut nodig om goed te kunnen opvangen langs de serverkant

typeof

let s1 = "randomString";

typeof s1	//geeft waarde String terug
-----------	-----------------------------

length

`console.log(s1.length)` *//geeft waarde 12 terug*

`charAt(index)` *individuele karakters opvragen*

`s1.charAt(1) = "@";` *//beter dan `s1[1] = "@"`;*

`concat` *meerdere strings aan elkaar plakken*

`let s2 = s1.concat("extraTekst");` *//zelfde als `s2 = s1 + "extra tekst"`;*

String literals

`\n` *newline*

`\'` *single quote*

`\''` *double quote*

`\\` *backslash*

`\u89ab` *Unicode karakter 89ab*

Hoofdletters/Kleine letters

`.toUpperCase(), .toLowerCase()`

Trim

`let tekst = " Hello world ";`

`console.log(tekst.trim());` *verwijdert alle whitespaces*

`LocaleCompare` *Strings vergelijken met elkaar (lexografisch)*

`s1.localeCompare(s2);`

1 → *s2 komt voor s1*

-1 → *s1 komt voor s2*

0 → *equal*

Alfabetisch, kortere woorden eerst nemen, met hoofdletters voor zonder

Identity vs equality

Identity *gaat het al dan niet om één en hetzelfde ding* `a==b`

--wijzigen beide variabelen al dat niet naar hetzelfde object?--

Equality *stellen twee dingen hetzelfde voor* `a.equals(b)`

--beschouwt de auteur de objecten als gelijk?—

Arrays sorteren

Getallen sorteren:

```
const compare = (a,b) =>{  
    return a-b;  
}
```

Strings sorteren:

```
const compare = (a,b) =>{  
    return a.localeCompare(b);  
}
```

Geselecteerde items opvragen in <select>

```
let selectElement = document.getElementById('mySelect');  
let selectedIndex = selectElement.selectedIndex;           -1: niets geselecteerd  
let selectedOption = selectElement.options[selectedIndex];  
let selectedValue = selectedOption.value;  
let selectedText = selectedOption.text;  
for(option) option.selected                               geselecteerd of niet, bij meerdere
```

Bevat string een geldig getal?

```
const isGetal = (tekst) => {  
    return !isNaN(tekst);  
}
```

parentNode

node.parentNode	<i>null indien er geen is</i>
<u>childNodes</u>	
node.childNodes	<i>lijst object dat alle children van node bevat, geen array, wel length en [idx]</i>

firstChild/lastChild

node.firstChild/node.lastChild	<i>null indien er geen is</i>
<u>nextSibling/previousSibling</u>	
node.nextSibling/node.previousSibling	<i>null indien er geen is</i>

ENKEL element nodes

n.children
n.firstChildElement/n.lastChildElement
n.nextElementSibling / n.previousElementSibling

Properties nodes

node.nodeName	<i>soort element als string indien element, #text indien textNode</i>
node.nodeType	<i>1 indien elementNode, 3 indien textNode</i>
node.nodeValue	<i>tekst indien textNode, anders null</i>

Element nodes aanmaken

document.createElement(e)	<i>bv. e=img,h1,...</i>
document.createTextNode(t)	<i>t=string alternatief... textContent</i>

Attributen nodes

e.getAttribute(a)	<i>a="href","src",...</i>
e.setAttribute(a,v)	<i>a="href", "src" v="waarde"</i>
e.hasAttribute(a)	<i>null indien leeg</i>

Data-attributen

Data opslaan, zelf verzinnen

AppendChild

<pre>let paragraph = document.createElement("p"); let textNode = document.createTextNode("This is a dynamically created paragraph."); paragraph.setAttribute("id", "dynamic-paragraph"); paragraph.setAttribute("class", "dynamic-paragraph"); paragraph.appendChild(textNode); document.getElementById("container").appendChild(paragraph);</pre>
--

target & currentTarget

```
<div id="parent">
  <button id="child">Child</button>
</div>

document.getElementById('parent').addEventListener('click', function(event) {
  console.log('event.target:', event.target);

  console.log('event.currentTarget:', event.currentTarget);
})
```

event.currentTarget = div.parent

EVENTLISTENER

event.target = button.child

HIEROP HEBBEN WE GEKLIKT

event.stopPropagation()

```
document.getElementById('childElement').addEventListener('click', function(event) {
  event.stopPropagation();

  console.log('Child element clicked');
});
```

Zorgt ervoor dat de parent event handlers niet worden getriggerd

event.preventDefault()

```
document.getElementById('myLink').addEventListener('click', function(event) {
  event.preventDefault();

  console.log('Link clicked, but default action prevented');
});
```

Zorgt ervoor dat de default acties niet worden uitgevoerd bij dat event. De link zal niet werken, maar er zal wel iets gelogd worden.

BELANGRIJK: return False in Event Listener = combinatie van stopPropagation & preventDefault

setInterval(funcitie,ms) *code herhaaldelijk laten uitvoeren om zoveel ms*

```
let taskId = setInterval(print,500);
const print = () =>{... clearInterval(taskId)}
```

setTimeout(funcitie,ms) *code 1x laten uitvoeren na wachttijd in ms*

```
let taskId = setTimeout(print,500); //liefst dit gebruiken ipv setInterval (vertraging vermijden)
clearTimeout(taskId); //geplande oproep annuleren
```


Random getallen

let getal = Math.random();	<i>kommagetal tussen 0 en 1</i>
let getal = Math.random() * 15;	<i>getal tussen 0 en 15</i>
let getal = 10 + Math.random() * 15;	<i>getal tussen 10 en 25</i>

Gehele getallen bekomen

Math.floor(kommagetal)	<i>rond getal af naar onder</i>
Math.ceil(kommagetal)	<i>rond getal af naar boven</i>
Math.round(kommagetal)	<i>rond af naar boven of onder</i>

Global gebruiken

let global = { score:0, players: [] PATH_PREFIX: "images\sprite" }	<i>//constante, daarom CAPS</i>
--	---------------------------------

Datums

let vandaag = new Date();	
let verjaardag = new Date(2023, 11,17);	<i>//OPLETTEN, december, maar -1 voor maand</i>

Eigen objecten aanmaken

let student = {};	
student.voornaam = "Arthur";	of student[voornaam] = "Arthur";
student.adres = {};	
student.adres.gemeente = "8790";	
delete student.voornaam;	<i>property verwijderen</i>

Indien je een property probeert te gebruiken die niet bestaat → undefined

Literals

JSON-object opbouwen in 1x ipv hierboven

JSON-omzettingen

let tekst = JSON.stringify(student);	<i>JSON → String</i>
let student = JSON.parse(tekst);	<i>String → JSON</i>

Web Storage API

Per domein data opslaan in browser

- Session storage *gekoppeld aan tabblad, behouden zolang tabblad open blijft*
- Local storage *mogelijks delen over meerdere tabbladen, ook na afsluiten b*

<pre>localStorage.setItem("myKey","myValue"); console.log(localStorage.getItem("myKey"); sessionStorage.setItem("myKey","myValue"); console.log(sessionStorage.getItem("myKey"); removeItem("myKey"); clear();</pre>	<p><i>item voor key verwijderen</i></p> <p><i>alle items verwijderen</i></p>
--	--