

Slideserie: Hva er JavaScript?

DS3103 Webutvikling

Høst 2022

Rolando Gonzalez

Temaer for slideserien

- Hva er JavaScript?
- Hva brukes JavaScript til?
- Hva er DOM?
- Sjekke feil i JavaScript

Hva er JavaScript?

- Utviklet av Netscape i 1995.
- JavaScript er De Facto klientside script-språk pga. sin gode portabilitet, dvs. virker i "alle" kjente nettlesere.
- JavaScript er objektorientert.
- Muliggjør dynamiske websider:
"A script can adapt the content based on input from the user or other variables, such as the time of day or the type of browser used by the client. Such web pages are said to be **dynamic**, as opposed to **static**, since their content has the ability to change."

(Deitel & Deitel, 2008, sd. 207)

Hva brukes JavaScript til?

- Endre HTML:
 - Legge til
 - Fjerne
- Endre attributter på HTML
- Endre CSS
- Stilsette elementer:
 - Direkte gjennom JS-metoder
 - Gjennom å sette klasser eller id
- Skape events:
 - click, mouseover, keyup osv.
 - Tidsintervaller
- Cookies

JavaScript - ECMAScript

- Basert på ECMAScript som er en definisjon av språket.
- ECMAScript 2015 (også kalt ECMAScript 6) og ECMAScript 2016 (ECMAScript 7) er under stadig implementasjon*
- Og videre ECMAScript 2017 - 2019
- ES.Next er navnet på neste (dynamisk) versjon av ECMAScript

*dvs. at nettleserne oppdateres til å forstå koden.

Hva er et script-språk?

- “A *scripting language* is a programming language that is used to manipulate, customize, and automate the facilities of an existing system. In such systems, useful functionality is already available through a user interface, and the scripting language is a mechanism for exposing that functionality to program control. In this way, the existing system is said to provide a host environment of objects and facilities, which completes the capabilities of the scripting language. A scripting language is intended for use by both professional and non-professional programmers.”
- (ECMA International, 2019)

Hva er DOM?

- The Document Object Model (DOM) is an application programming interface ([API](#)) for valid [HTML](#) and well-formed [XML](#) documents. It defines the logical structure of documents and the way a document is accessed and manipulated.
 - The HTML Document Object Model (HTML DOM) defines a standard way for accessing and manipulating HTML documents.
 - The DOM presents an HTML document as a tree-structure (a node tree), with elements, attributes, and text.
- (W3C, 2000)

DOM

- Det er flere måter å definere/forstå DOM
- Man kan definere DOM som nettsidens innhold som et objekt.
- Gjennom DOM kan HTML-elementer bli behandlet som objekter og deres egenskaper som attributter. JavaScript benyttes for å manipulere DOM-objekter.
- Metoden getElementById returnerer en DOM-node (som er et objekt).

Relasjonen mellom HTML-elementer

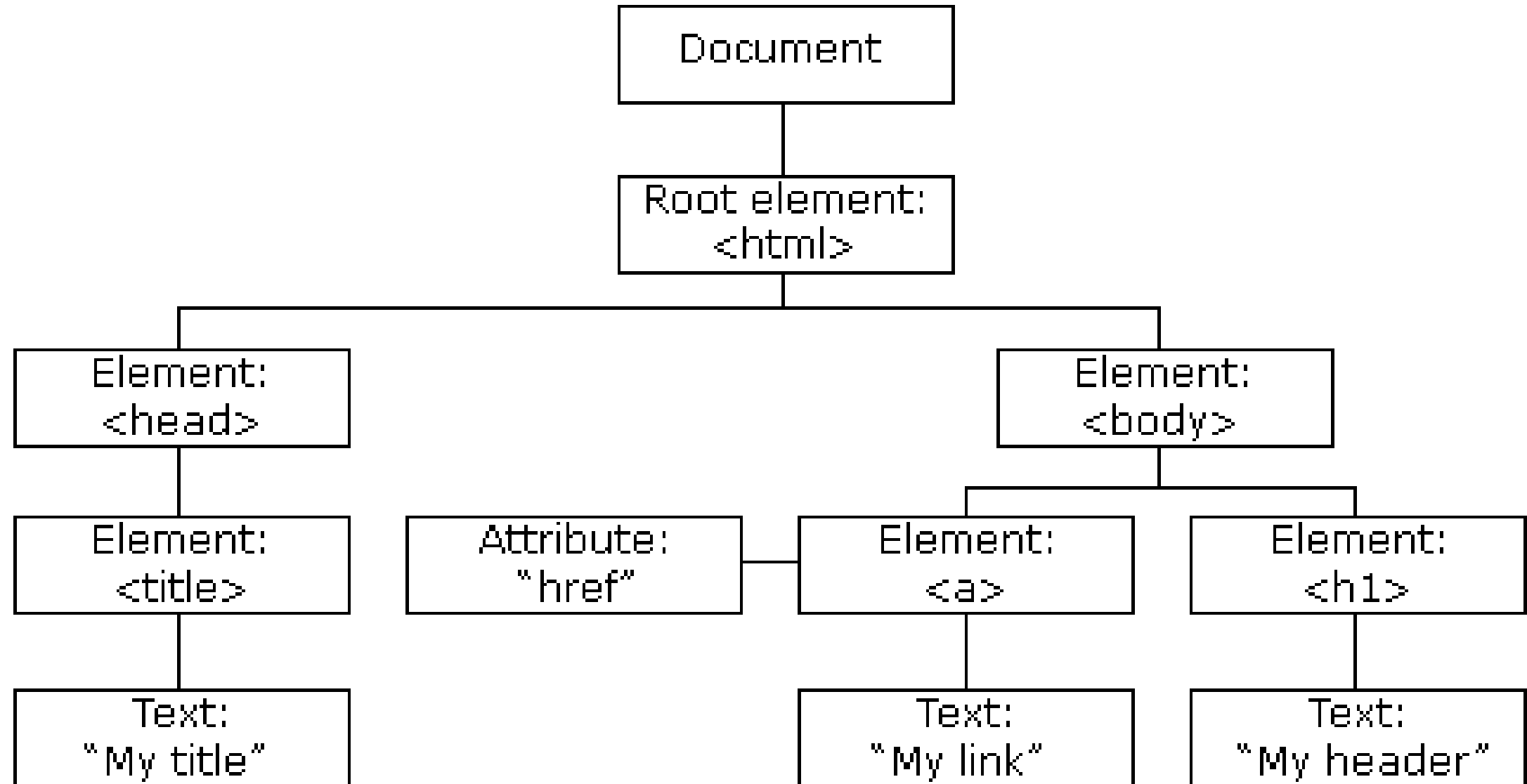
- Et HTML-element som befinner seg i et annet element er et barneelement:

```
<div> <!--Forelder til <p> og <h1>-->  
    <h1></h1> <!--Barn av <div> Søsken med <p>-->  
    <p></p> <!--Barn av <div> Søsken med <h1>-->  
</div>
```

<head> og <body> er søsken, og barn av <html>

- Elementer kan ha flere barn, men hvert barn har bare en forelder.

Grafisk presentasjon av DOM



Grafisk presentasjon av DOM

<table>

<tbody>

<tr>

<td>Shady Grove</td>

<td>Aeolian</td>

</tr>

<tr>

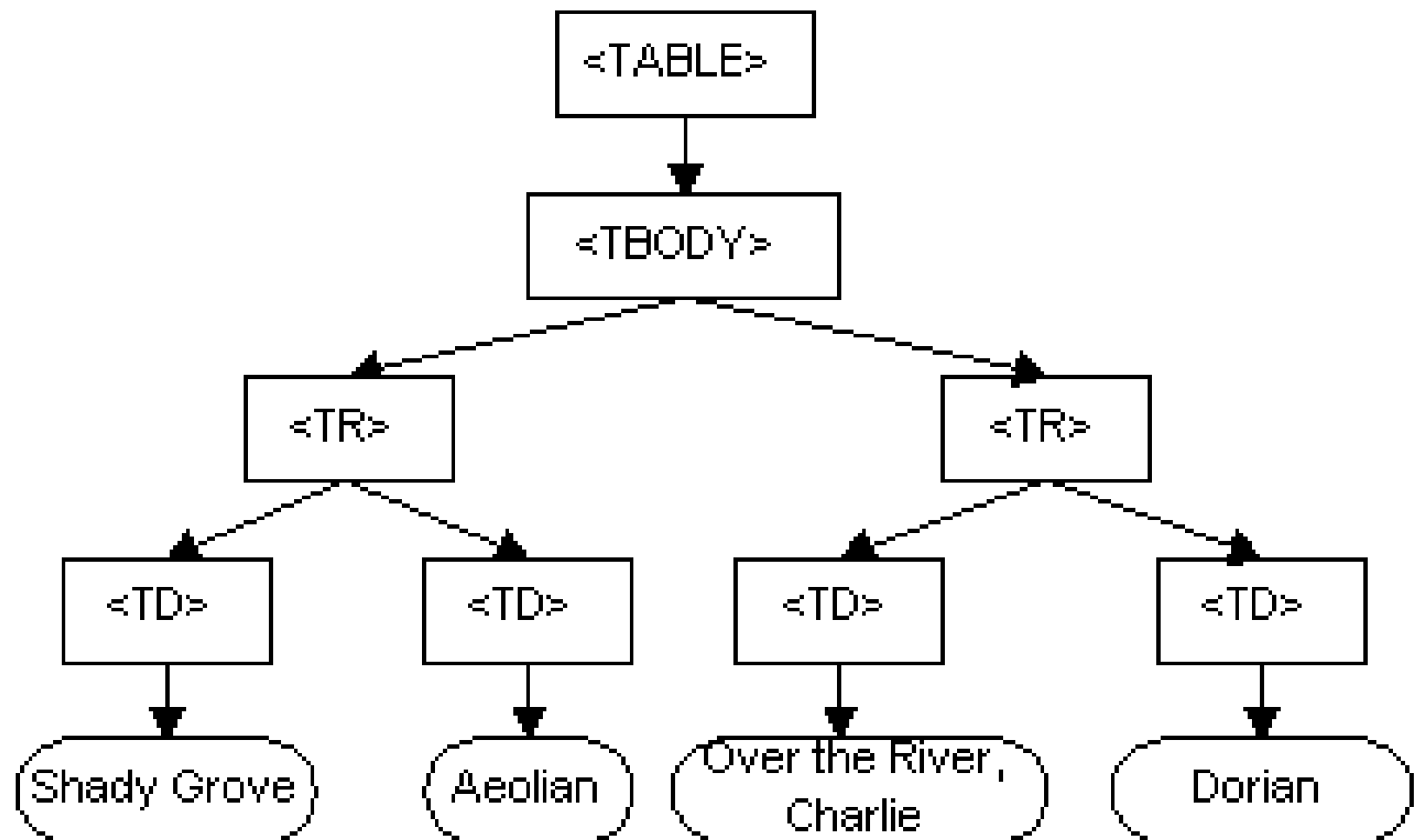
<td>Over the River, Charlie</td>

<td>Dorian</td>

</tr>

</tbody>

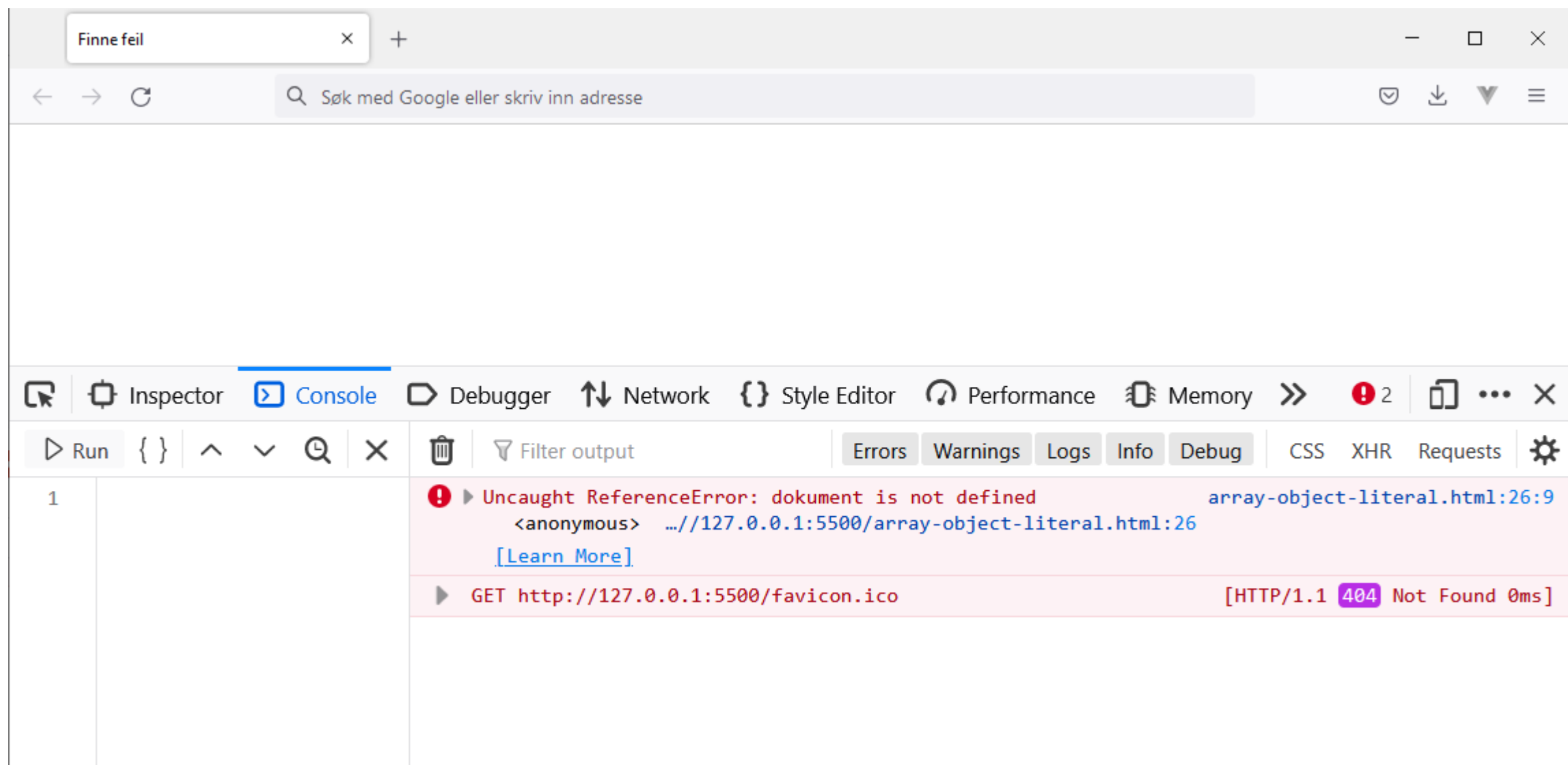
</table>



(W3C.com, 2000)

Sjekke JavaScriptkoden for feil

- Alle større nettlesere har det som heter Error Console/Web Console hvor man kan sjekke feil i JavaScriptet.



alert/console.log og feilsjekking

- `window.alert()` og `console.log()` er ofte praktisk for å gjøre feilsjekk på logiske feil (versus syntaksfeil). For eksempel for å teste om en funksjon blir utført eller ikke.

JS-kode kan plasseres i head og/eller body, men...

```
<html>
<head>
<script>
//kode her
</script>
</head>
<body>
...
<script>
//kode her
</script>
</body>
</html>
```

Ekstern JavaScript-kode

- ...men, det optimale er å plassere JS i egen fil.
- Man kan legge JavaScript i ekstern fil og hente den inn i HTML ved å bruke src-attributtet som angir navnet på JavaScript-filen:
`<script src="name-of-file.js"></script>`
- Eksterne koder kan kalles på både fra head og body og man kan hente inn flere scripts.

noscript-tagen

- noscript-tagen kan benyttes for å instruere brukeren om at JavaScript-koden på siden ikke er kjørt/ikke forstås av/er blokkert av nettleseren.
- Det kan være greit å ha med link til alternativ side eller til en informasjonsside som forklarer hvilke innstillinger som må settes for at nettleseren skal kunne eksekvere JavaScript.
- noscript-tagen benyttes i body til HTML-dokumentet.

<noscript>

<p>JavaScript-kode for denne siden er enten blokkert eller forstås ikke av din nettleser. Alternativ side/Infoside</p>

</noscript>

defer

- Opprinnelig er det slik at nettsidens kode (HTML, CSS og JavaScript) genereres fra topp til bunn, linje for linje. Dette betyr at hvis man har `<link>` (CSS) eller `<script>` (JS) i `<head>` så vil nettsiden utføre den før den så fortsetter til å genere HTML nedover i .html-filen.
- Overnevnte er også viktig å være klar over fordi at hvis JS skal gjøre DOM-manipulasjon så forutsetter det at DOM (dvs. HTML) er ferdig skapt.
- «defer» er et attributt som kom med ES6+ som man kan sette på script for å få JS-kode i den til å først utføres etter at DOM (HTML) er generert. Dette betyr at vi nå kan plassere `<script>` hvor som helst så lenge vi legger det til: `<script defer src=...`

Referanser

- ECMAScript International (2019) ECMAScript Language Specification, <https://tc39.es/ecma262/#sec-overview>, [last accessed 09.09.19]