

INF105 - 12
Client-side: JS

David Grellscheid

Kommende gruppetime

- torsdag 24.9. / mandag 28.9.
- Fortsetter med HTML / CSS / JS
- ingen oppmøtesjekk

  men kom hvis serveren ikke virker / du har spørsmål / vil jobbe sammen

Quiz-resultatene

- Kan sjekke rettingen i neste gruppetime
- Mulig å ta quiz om igjen: ***tirsdag 29.sep, 16:15 i grupperommet***
  - kunngjøring kommer på mitt.uib med tidspunkt
  - første forsøk vektes da med 20%, nytt resultat med 80%

Frontend:
HTML + CSS + *JS*

https://developer.mozilla.org/en-US/docs/Learn_web_development/Core

JS: JavaScript

- Ingen relasjon til Java-programmeringsspråket!
- JS er et fullstendig, eget programmeringsspråk
- programmerbar interaksjon på nettsidene
- Browseren er interpreter / compiler og utfører programmene
- JS kan også kjøres i kommandolinjen, med f.eks *node.js*

JS: JavaScript

- JS har variabler, lister, løkker, betingelser, funksjoner, ...
- konseptene er likt de som finnes f.eks i Python,
  men *syntax* er annerledes

API
Application Programming Interface

- definert grensesnitt med biblioteker av funksjonalitet
- noen API er innebygd i browser
- kan også bruke eksterne API fra andre nettsider

Browser-API

- viktigst: DOM-API
  gir tilgang til HTML-elementene og CSS
- Canvas: tegninger
- WebGL: 3D-grafikk, støttet av grafikkortet
- Audio, Video, Geolocation, ...
- https://developer.mozilla.org/en-US/docs/Web/API

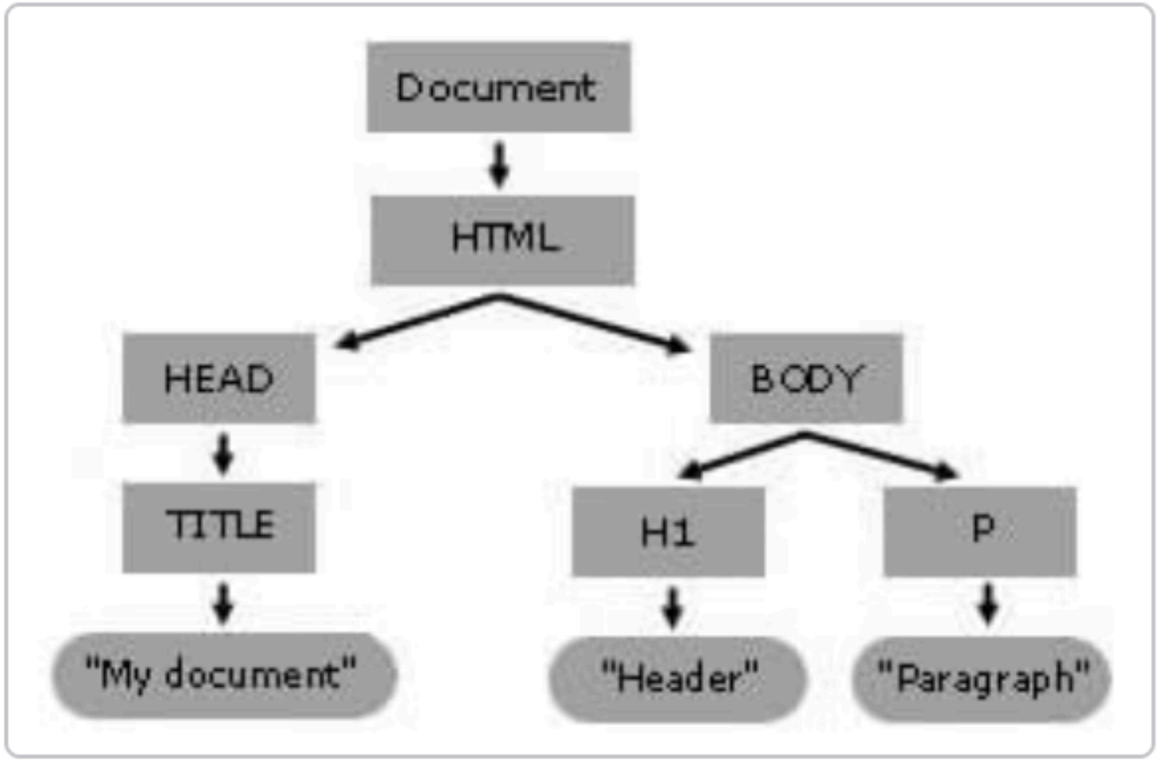
DOM
Document Object Model

Header

Paragraph

- Representerer strukturen til dokumentet som en graf

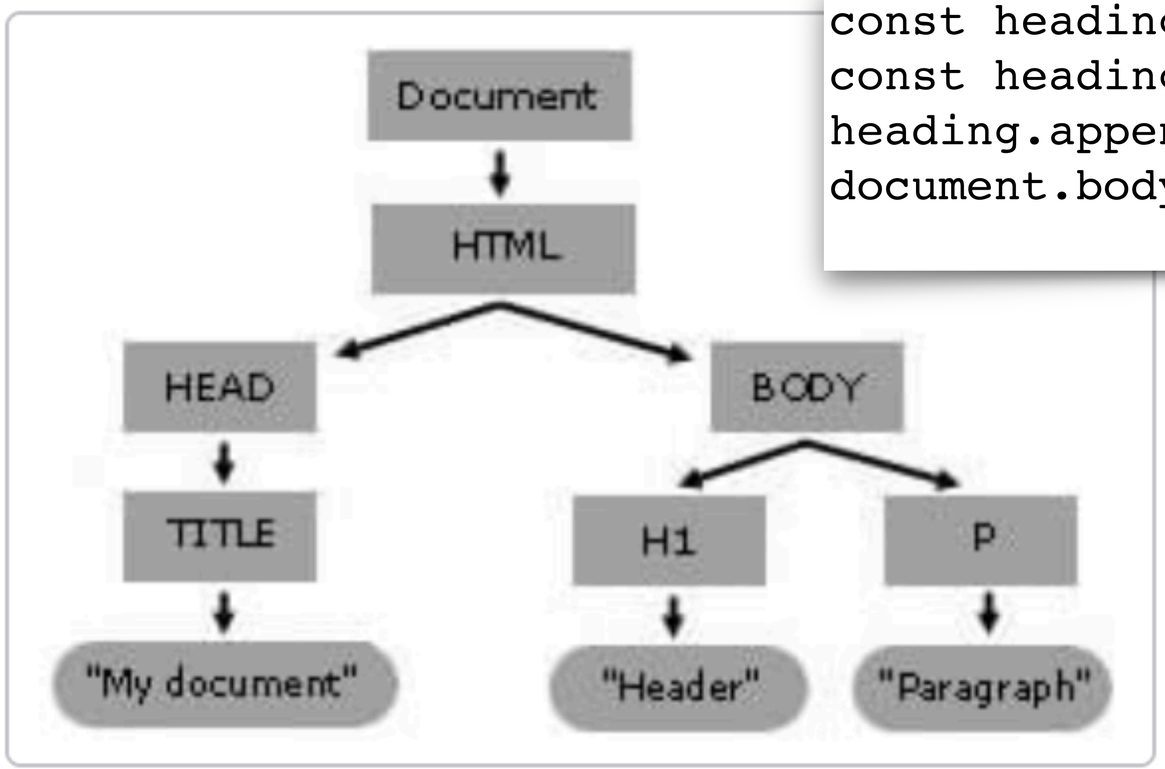
```
<html lang="en">
  <head>
    <title>My Document</title>
  </head>
  <body>
    <h1>Header</h1>
    <p>Paragraph</p>
  </body>
</html>
```



https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model#what_is_a_dom_tree

DOM
Document Object Model

- Kan manipulere grafen i JS

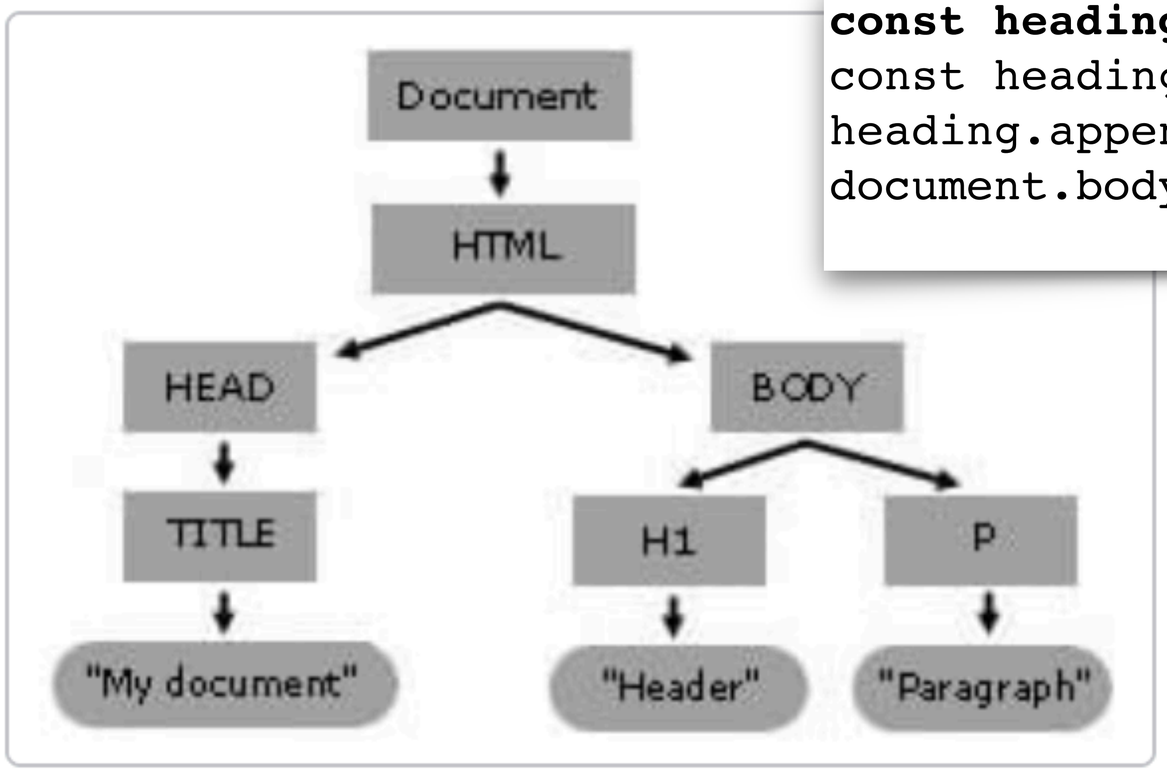



```
const heading = document.createElement("h1");
const headingText = document.createTextNode("Big Head!");
heading.appendChild(headingText);
document.body.appendChild(heading);
```

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model#what_is_a_dom_tree

DOM
Document Object Model

- Kan manipulere grafen i JS



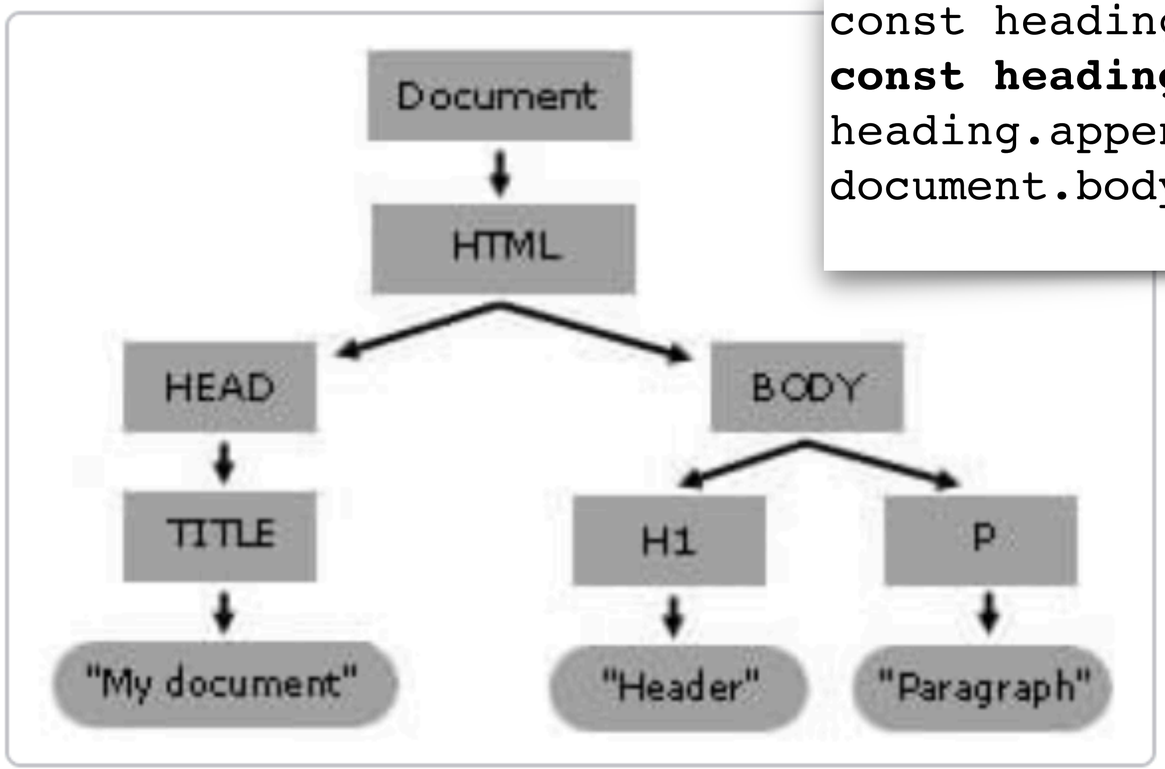

```
const heading = document.createElement("h1");
const headingText = document.createTextNode("Big Head!");
heading.appendChild(headingText);
document.body.appendChild(heading);
```

H1

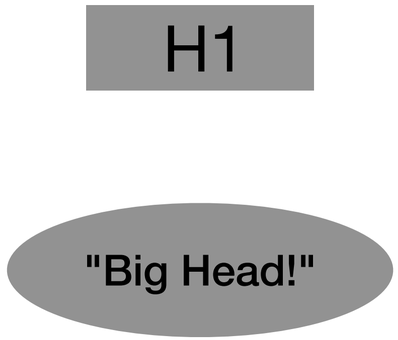
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model#what_is_a_dom_tree

DOM
Document Object Model

- Kan manipulere grafen i JS



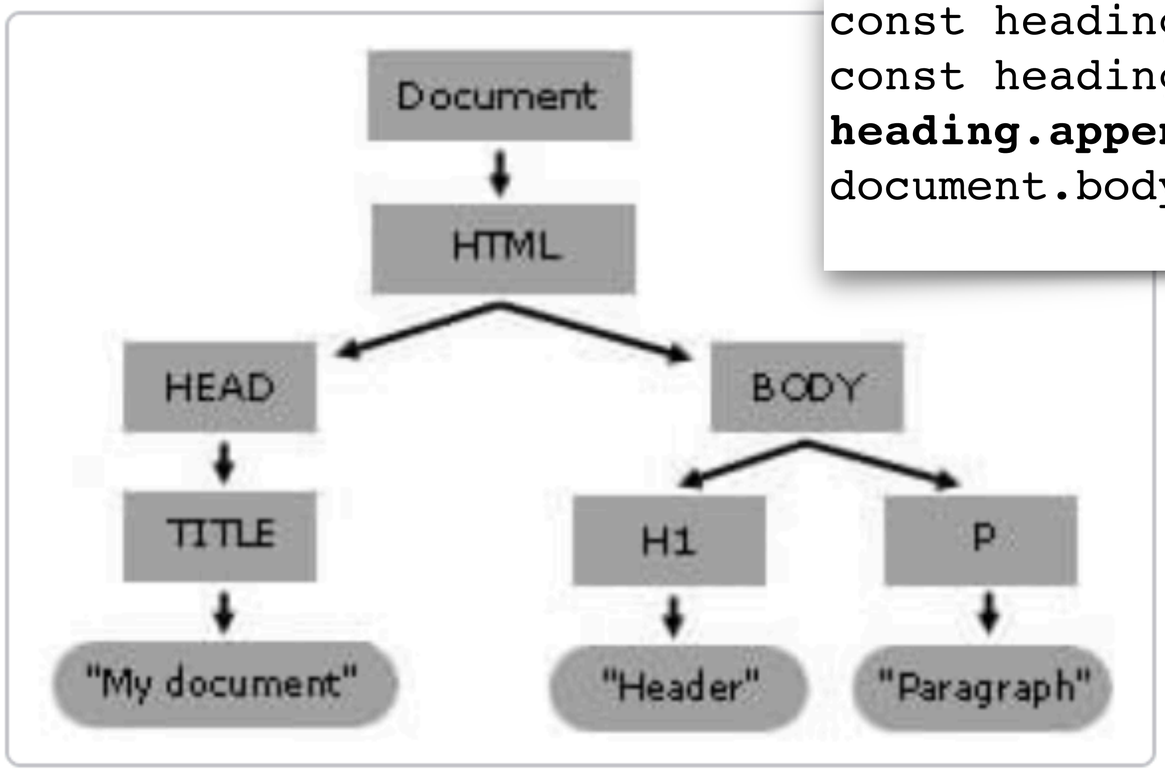

```
const heading = document.createElement("h1");
const headingText = document.createTextNode("Big Head!");
heading.appendChild(headingText);
document.body.appendChild(heading);
```



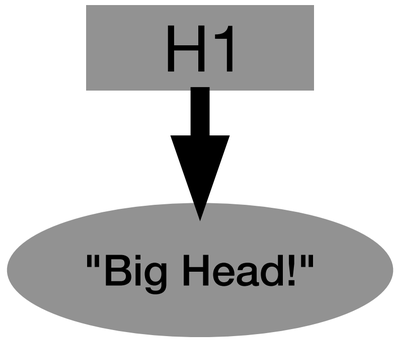
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model#what_is_a_dom_tree

DOM
Document Object Model

- Kan manipulere grafen i JS




```
const heading = document.createElement("h1");
const headingText = document.createTextNode("Big Head!");
heading.appendChild(headingText);
document.body.appendChild(heading);
```

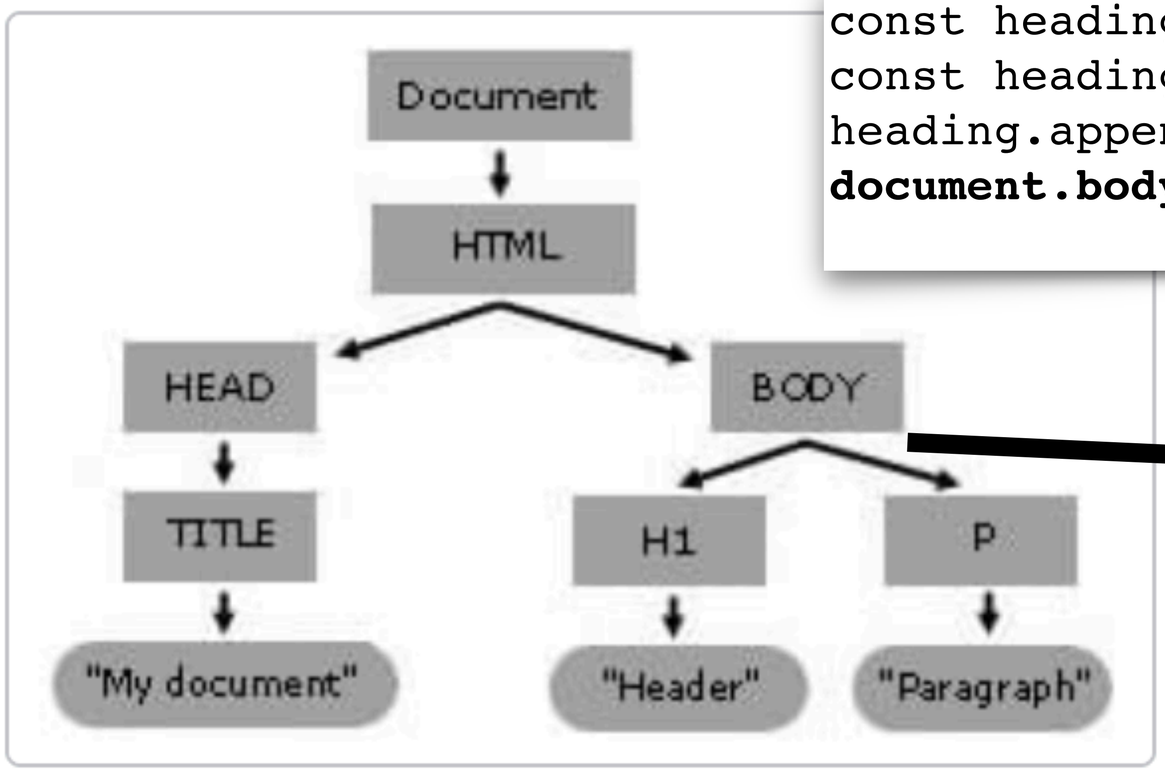


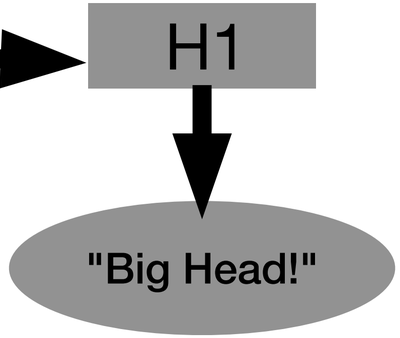
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model#what_is_a_dom_tree

DOM
Document Object Model

- Kan manipulere grafen i JS, tilgjengelig som "*document*" variable

```
const heading = document.createElement("h1");
const headingText = document.createTextNode("Big Head!");
heading.appendChild(headingText);
document.body.appendChild(heading);
```





https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model#what_is_a_dom_tree

DOM
Document Object Model

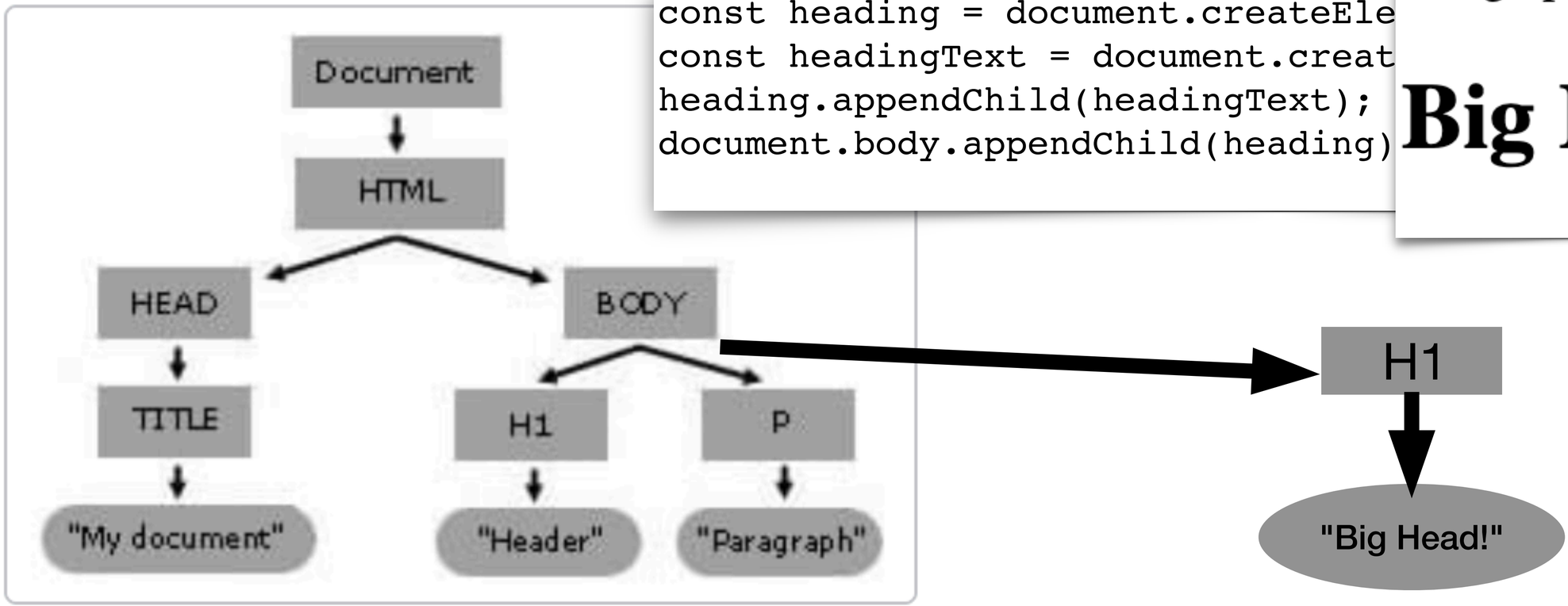
- Kan manipulere grafen i JS, tilgjengelig som "*document*"

```
const heading = document.createEle
const headingText = document.creat
heading.appendChild(headingText);
document.body.appendChild(heading)
```

Header

Paragraph

Big Head!



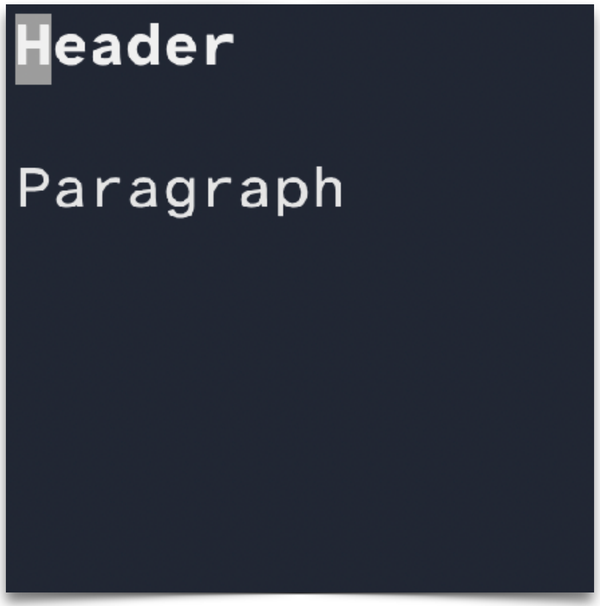
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model#what_is_a_dom_tree

JS må være på i browseren

Firefox:



w3m i terminal:



demo 00

Seleksjon av HTML-element

- const e = document.**getElementById**("foobar");

  ett element, som har <... id="foobar">

- const e = document.**querySelector**("p.info");

  første elementet som er .....

- const e = document.**querySelectorAll**("p.info");

  alle elementer som er .....

Eksempler til bruk av HTML-element

- const fbar = document.getElementById("foobar");
  fbar.style.color="red";

- const infop = document.querySelector("p.info");
  infop.textContent = "new text here";

- const allps = document.querySelectorAll("p.info");
  allps.forEach(p => p.style.color = "red");

demo 01

Eksempel: Canvas

- Programmerbar grafikk
- Se f.eks https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations

https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial

demo 02

Eksempel: WebGL

- Programmering på grafikkortet: vertex / fragment shaders, etc.

demo 03

Eksterne API

- mange!
- yr.no: https://developer.yr.no/doc/GettingStarted/
- Entur: https://developer.entur.no/docs/getting-started
- ...
- tilgang gjennom HTTP-kall, sender JSON-format tilbake
  (JavaScript Object Notation)
- **Viktig**: les dokumentasjon og betingelser om bruk!
  (identifikasjon, rate limits, sjekk om data er oppdatert, ...)

Eksempel: Værmeldinger

- YR API-dokumentasjon: https://developer.yr.no/doc/GettingStarted/

demo 04

Javascript hos UiB

Involving Students in Design and Implementation of ECMAScript Proposals

Mikhail Barash (University of Bergen, Norway)
SPLASH-E 2025, Singapore

Outcomes

- **Merged PRs** to two JavaScript engines (*SpiderMonkey* and *Boa→V8*)
- **Merged Test262 test cases**
- First-hand experience in interacting with **professional review process**
- **Soft skills**
  - teamwork
  - coordination in a distributed environment
  - professional communication
- **Student-driven knowledge transfer**

Javascript hos UiB

Involving Students in Design and Imple

[−] aapoalas 12 months ago

△ 3

Hiya Jase, glad to see you working in or around Boa still/again!

Just wanted to shout out that University of Bergen is pretty cool: they do programming language research and development in both ECMAScript's TC39 and the C++ committee. On a personal note, UoB students (who were in the Boa project) are now doing a project to bring temporal_rs into Nova JavaScript engine as well, so you'll soon have even more engines to count under your belt :)

[−] JaseW [OP] 12 months ago

△ 2

That's amazing news aapoalas! That's great news to have Nova engine join the fold. I'll certainly be keeping an eye out. And yes, the UoB have been awesome, we've really appreciated their help.

V8)

rocess

- coordination in a distributed environment
- professional communication

- **Student-driven knowledge transfer**

Javascript hos UiB

TC39 proposal	engine	implementation language	mentor support required	contributions and outcomes	knowledge transfer & outreach by students	remarks
Array Grouping (https://github.com/tc39/proposal-array-grouping)	SpiderMonkey	engine's "self-hosted" JavaScript	very significant	• implementation merged in Firefox Nightly[1]	• tutorial[2] • talk[3]	• first iteration of the course • two undergraduate students without any prior JavaScript experience • one primary industry mentor, two secondary industry mentors • another undergraduate student then independently reimplemented the proposal following the tutorial • industry mentor has given her own talk about the proposal[4]
Upsert (https://tc39.es/proposal-upsert/)	SpiderMonkey	engine's "self-hosted" JavaScript C++	moderate	• proposal design • ECMA-262 specification text[5] • Test262 tests[6] • implementation merged in Firefox Nightly[7]	• tutorial[8] • two talks[9,10]	• four undergraduate students and one graduate student • one industry mentor • students participated in API design and rationale formulation • an initial and then an optimized implementation in C++
	V8	C++ Torque[11]	(no mentor)	• local implementation • Test262 tests[6]	• tutorial[12] • talk[9]	• one graduate student (mentioned above), while participating in the SpiderMonkey implementation
Joint Iteration (https://github.com/tc39/proposal-joint-iteration)	SpiderMonkey	engine's "self-hosted" JavaScript	light	• local Test262 tests • local implementation	• materials for a next course iteration	• four undergraduate students • one industry mentor
Temporal (https://github.com/tc39/proposal-temporal)	Boa[13] *and* V8 (indirectly)	Rust	significant	• 15 merged PRs[14] • Test262 tests[15] • brought Test262 conformance of Boa's *Temporal* from 67% to 92%	• talk[9]	• six undergraduate students • one primary industry mentor (who also gave an in-person crash course), two secondary mentors • Boa's implementation of *Temporal* is used by Chrome's V8 engine[16]

Norske studente
JavaScript-nyhe
utrolig mulighet

A new course on
web browser engineering!
(150+ students,
to be given annually)



https://www.kode24.no/artikkel/norske-studenter-bak-javascript-nyhet-helt-utrolig-mulighet/243076

snakk med Mikhail Barash / ta emnet INF224 når det kommer i '28