

JavaScript 1
alert("Lektion 12");

Utbildare: Mahmud Al Hakim

NACKADEMIN

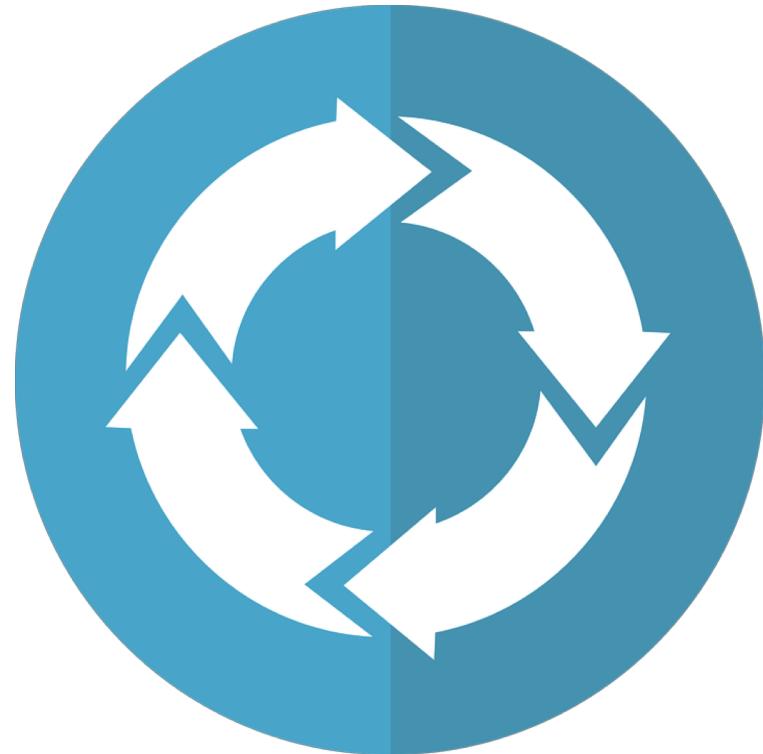
Lektionstillfällets mål

- **Mål med lektionen**
 - Arbeta med DOM
 - Att förstå DOM Tree
 - Att välja enskilda element
 - Arbeta med noder
- **Arbetsmetod**
 - Teori och praktik varvas under lektionen

Kort summering av föregående lektion

Föregående lektion:

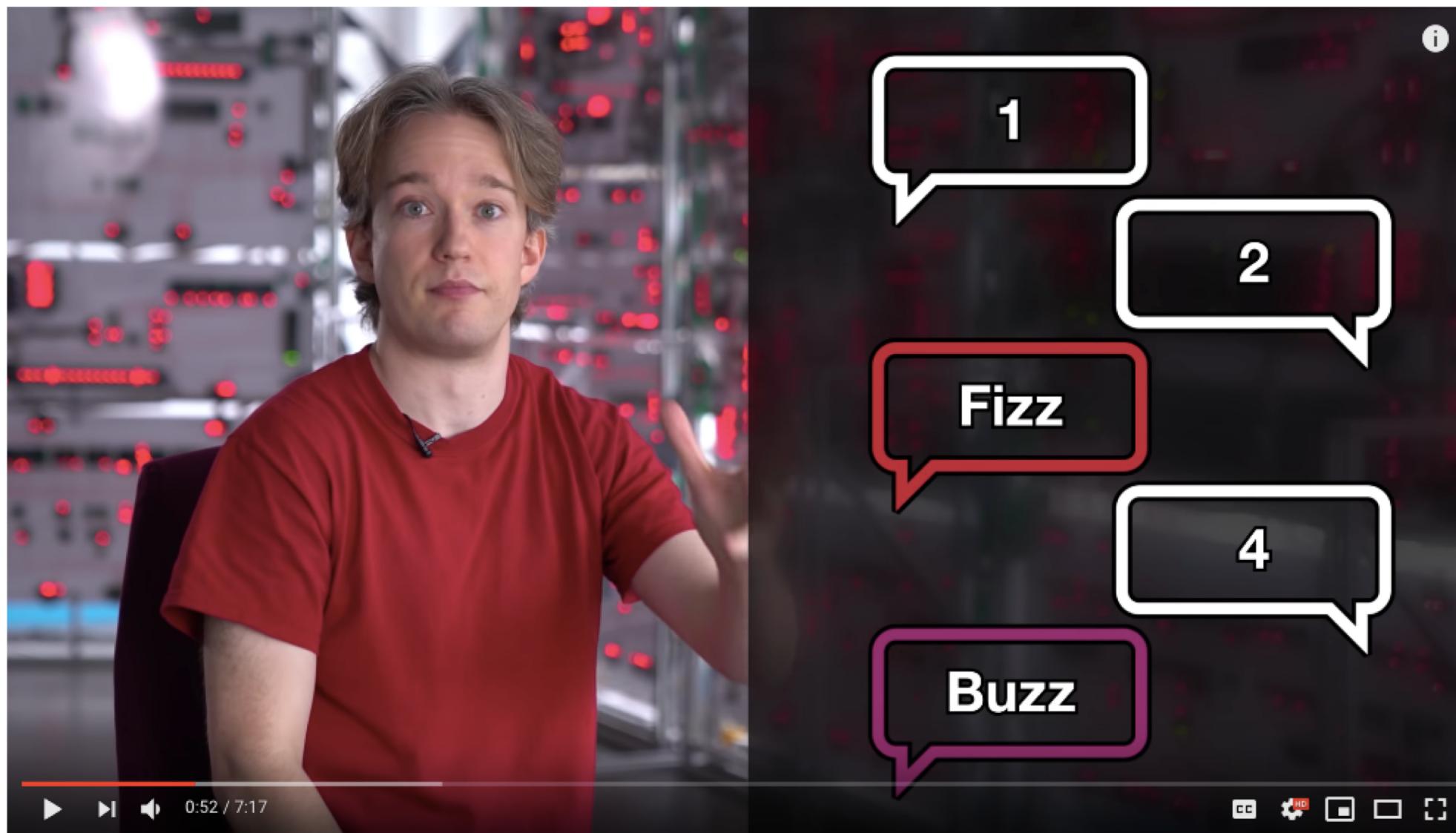
- Vi har jobbat med iterationer.



NACKADEMIN

FizzBuzz: One Simple Interview Question

https://youtu.be/QPZ0plK_wsc



Vad är DOM?

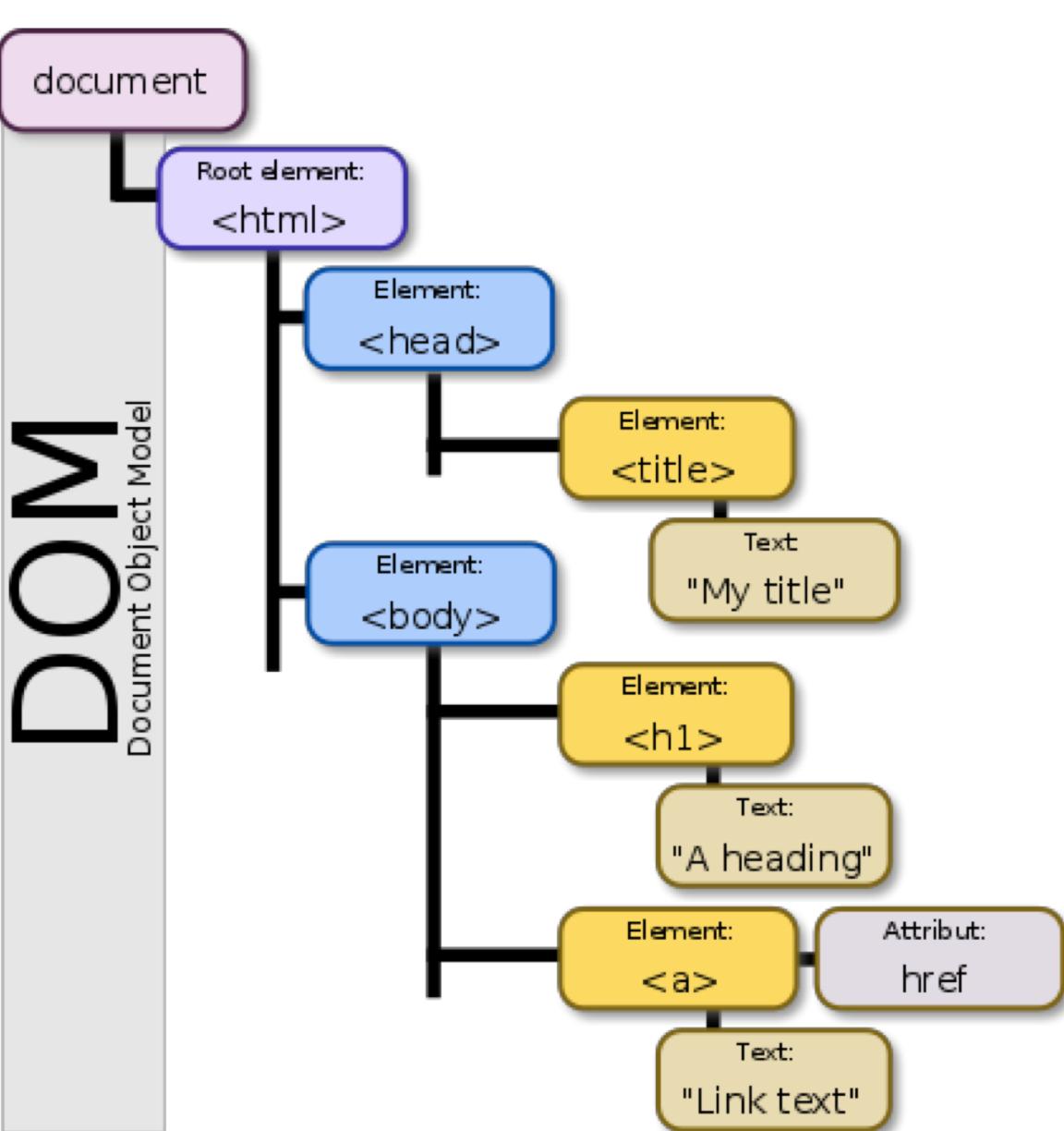
- DOM står för Document Object Model
- DOM är ett gränssnitt som ger programspråk möjligheten att dynamiskt läsa och uppdatera ett dokuments innehåll, struktur och formatering.
- DOM är ett API som utvecklas av W3C
- API = Application Programming Interface
- <https://www.w3.org/DOM>

Varför behövs DOM i JavaScript?

- DOM är en modell som finns i alla webbläsare.
- När en webbläsare laddar en HTML-sida så blir sidan ett ”dokument-objekt”.
- DOM definierar objekt av alla dokumentets element och även metoder att nå dessa element.
- Genom att använda DOM i JavaScript kan vi manipulera HTML.
- Vi kan alltså, komma åt, modifiera, lägga till och ta bort HTML-element.

DOM Tree

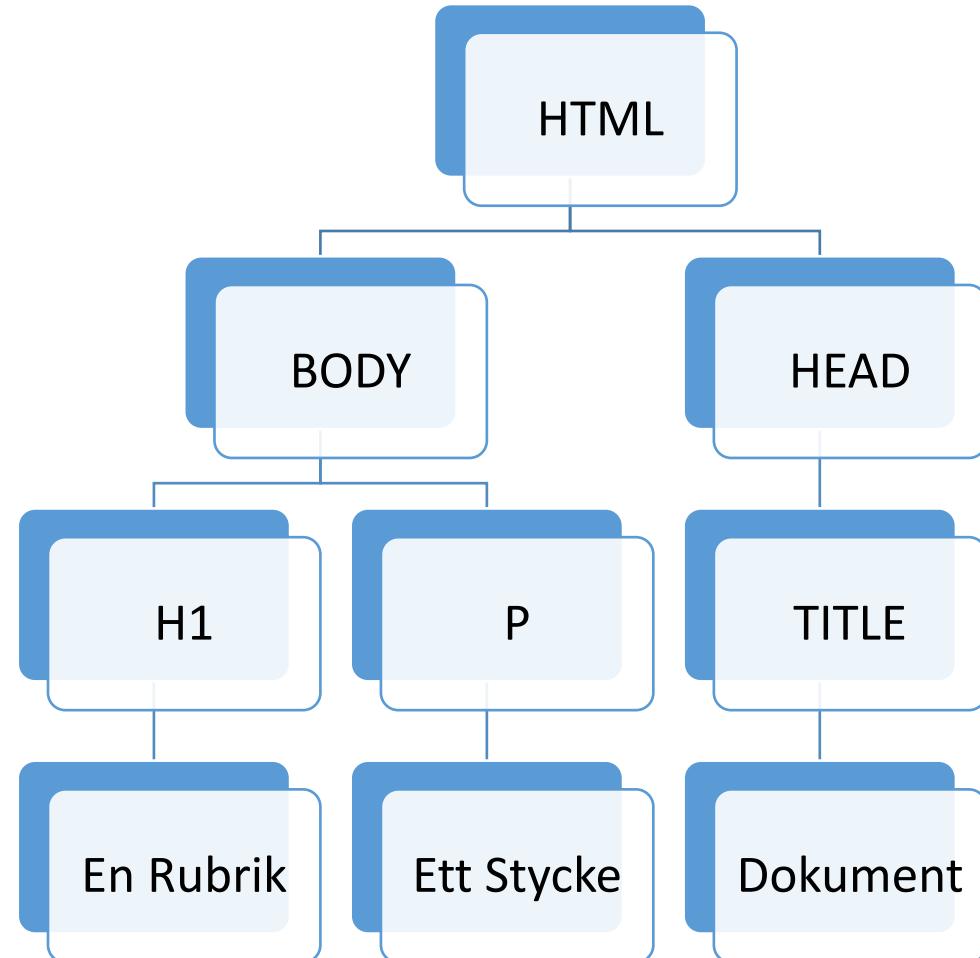
DOM Document Object Model



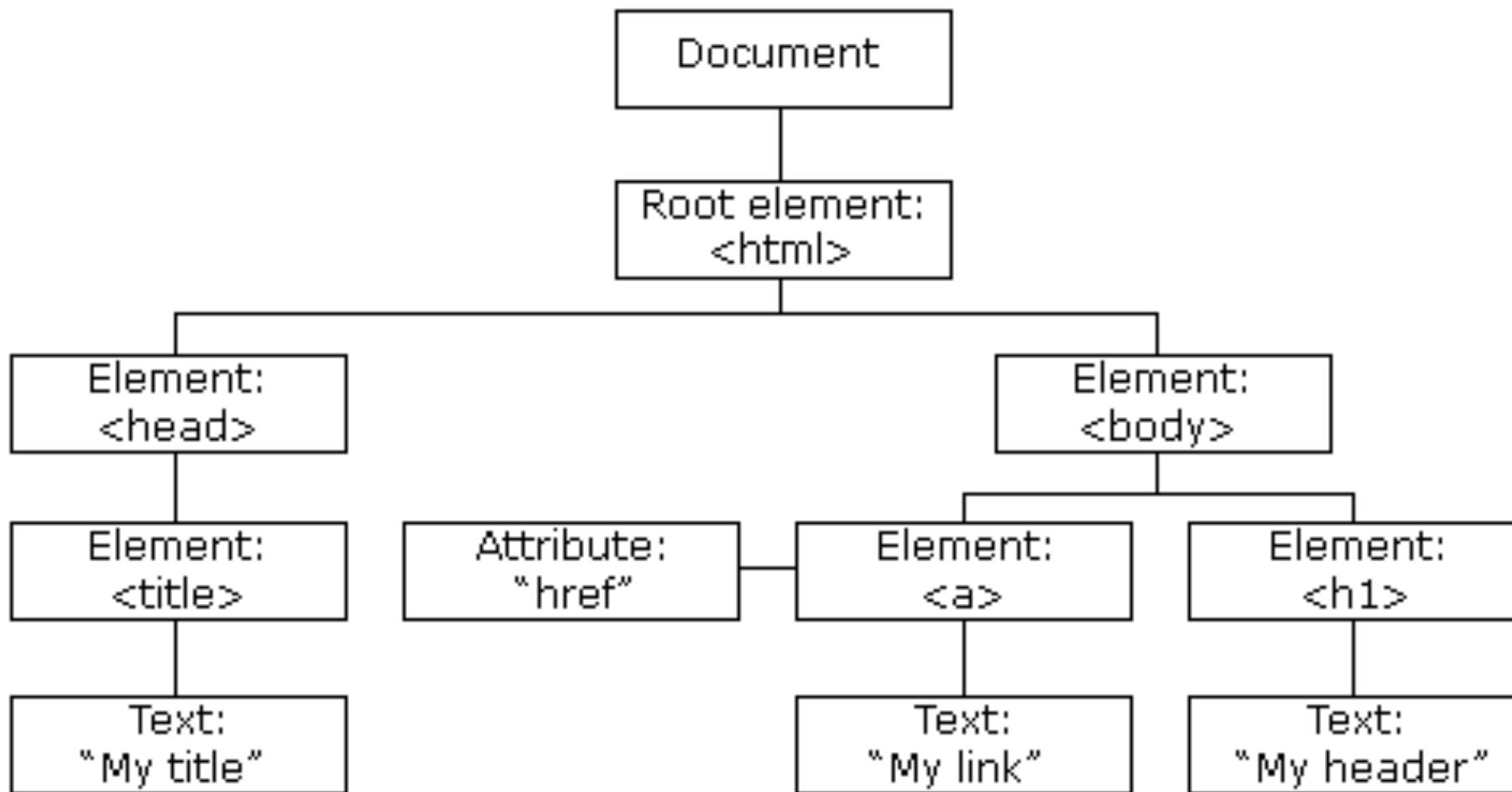
- DOM lägger in alla element i en väldefinierad trädstruktur.
- Trädet lagras i webbläsarens minne.
- **document** är ett objekt som är ”rooten” till alla andra objekt.
- Alla objekt i trädet kallas för **noder**.
- De vanligaste nodtyperna är:
Document, Element, Attribut och Text.
- Noderna är relaterade till varandra som en familj.
- En nod kan ha en förälder, syskon och barn!

DOM skapar ett träd av objekt

```
<html>  
  
  <head>  
    <title>Dokument</title>  
  </head>  
  
  <body>  
    <h1>En rubrik</h1>  
    <p>Ett stycke</p>  
  </body>  
  
</html>
```



The HTML DOM Tree of Objects



DOM Queries (frågor)

- Metoder som används för att hitta HTML-element i DOM-trädet kallas ”DOM Queries”.
- Om du behöver arbeta med element flera gånger så spara resultatet i en variabel för att slippa ställa samma frågor till trädet.
- Använd alltid variabler för att spara dina frågor.
- Dessa variabler blir då referenser till olika element som ligger i trädet.
- Detta kallas ”Caching” och ökar prestanda på dina program.

DOM Queries

Olika metoder för att hitta HTML-element

Metoder för att hitta ett enda element

getElementById()

Hittar ett element med hjälp av elementets id.

querySelector()

Hittar ett element med hjälp av en CSS-Selector.

Metoder för att hitta flera element

getElementsByName()

Hittar flera element som har samma tagg.

getElementsByClassName()

Hittar flera element som har samma CSS-klass.

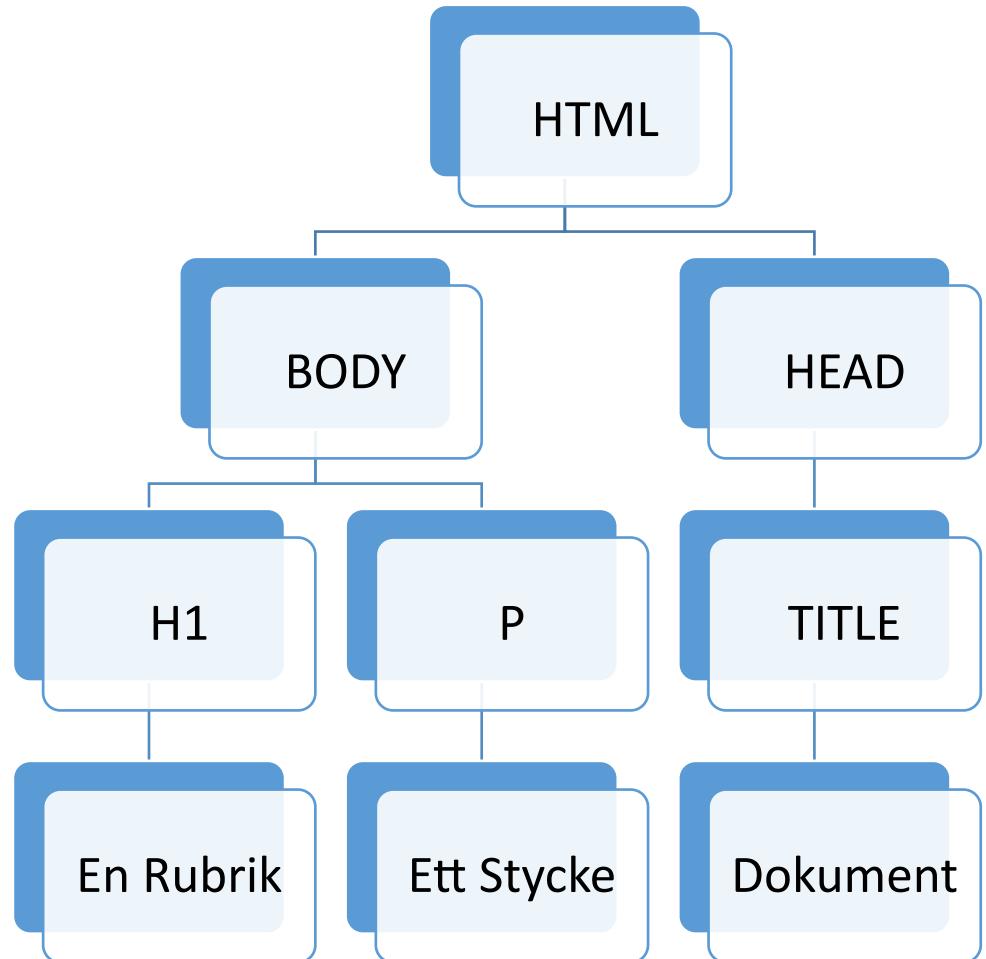
querySelectorAll()

Hittar flera element med hjälp av CSS-Selector.

Metoden getElementById

```
<h1 id="one">  
    En rubrik  
</h1>
```

```
<script>  
let h =  
document.getElementById('one');  
</script>
```



`document` refers to the `document` object. You always have to access individual elements via the `document` object.

OBJECT

The `getElementById()` method indicates that you want to find an element based upon the value of its `id` attribute.

METHOD

`document.getElementById('one')`

MEMBER OPERATOR

The dot notation indicates that the method (on the right) is being applied to the node on the left of the period.

PARAMETER

The method needs to know the value of the `id` attribute on the element you want. It is the parameter of the method.

Ändra innehåll i ett element

```
<h1 id="one">En rubrik</h1>
```

```
<script>
```

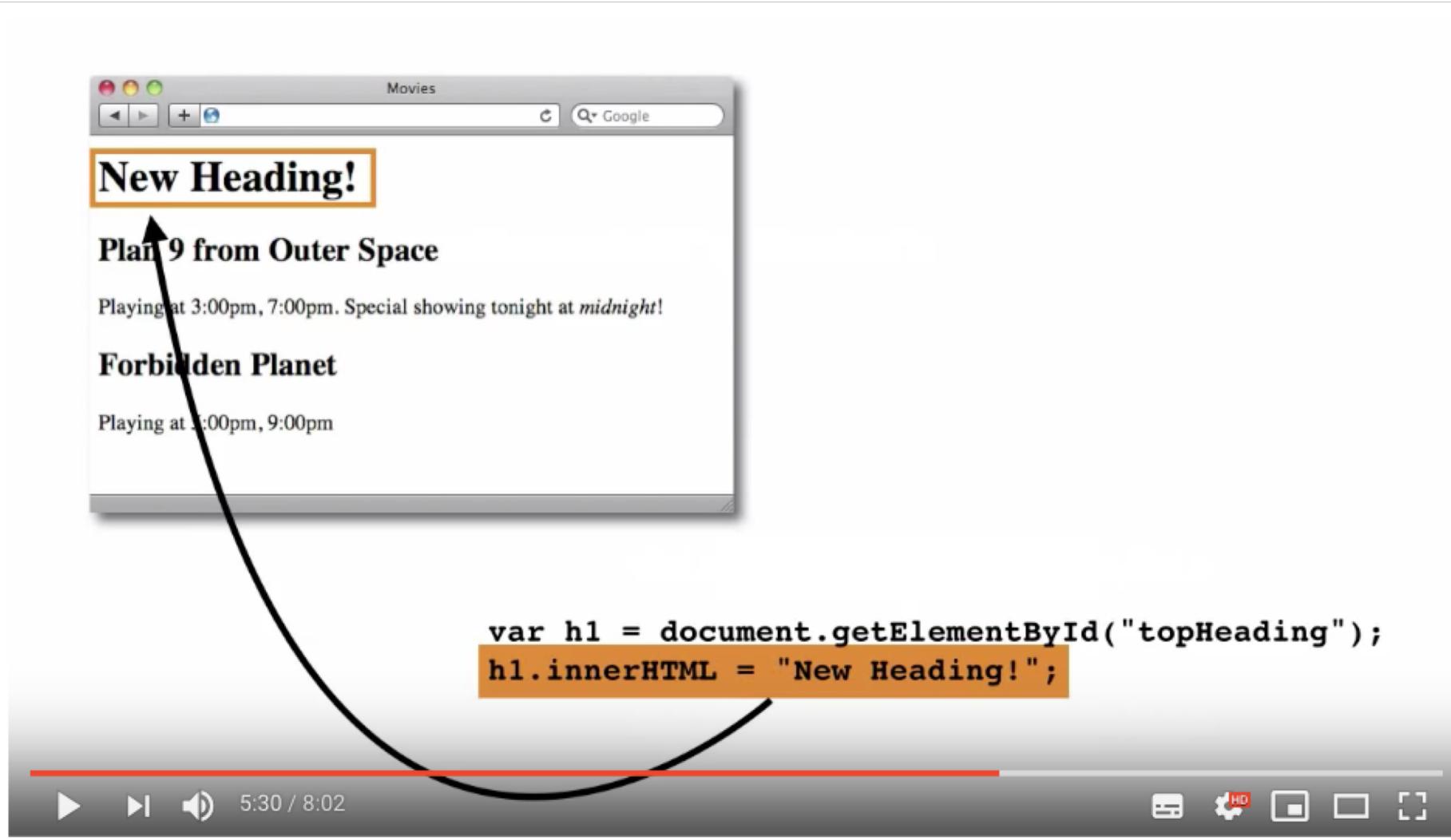
```
let h = document.getElementById('one');  
h.innerHTML = 'Välkommen till JS';
```

```
</script>
```

Video – Bra sammanfattning

Manipulating the Document Object Model in JavaScript

<https://youtu.be/4Gv81XcC8iE>



Övning

```
<h1 id="h">Huvudrubrik</h1>
```

- Skapa ett skript som letar efter huvudrubriken (h1).
- Ändra texten till versaler med JavaScript
(OBS! Använd inte CSS)

Ändra färg med CSS

```
<h1 id="one">En rubrik</h1>

<script>

let h = document.getElementById('one');

h.style.color = 'maroon';

</script>
```

Ändra teckenstorlek med CSS

```
<h1 id="one">En rubrik</h1>

<script>

let h = document.getElementById('one');

h.style.color = 'maroon';
h.style.fontSize = '50px';

</script>
```

Ändra till versaler med CSS

```
<h1 id="one">En rubrik</h1>

<script>

let h = document.getElementById('one');

h.style.color = 'maroon';
h.style.fontSize = '50px';
h.style.textTransform = 'uppercase';

</script>
```

Dölj ett element med CSS

```
<h1 id="one">En rubrik</h1>
```

```
<script>
```

```
let h = document.getElementById('one');
```

```
h.style.display = 'none';
```

```
</script>
```

Manipulera DOM med egenskapen body

```
<body>
<h1>Egenskapen body</h1>
<p>Body är en egenskap som finns i objektet document.</p>
<p>Egenskapen är en referens till elementet body.</p>
<script>
document.body.style.backgroundColor = "yellow";
console.log(document.body);
</script>
</body>
```

Metoden getElementsByTagName

```
<p>Stycke 1</p>
```

```
<p>Stycke 2</p>
```

```
<p>Stycke 3</p>
```

```
<script>
```

```
let p = document.getElementsByTagName('p');
```

```
console.log(p);
```

```
</script>
```

```
▼ HTMLCollection(3) [p, p, p]
```

```
▶ 0: p
```

```
▶ 1: p
```

```
▶ 2: p
```

```
length: 3
```

Iterera över element

```
let p = document.getElementsByTagName('p');

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

Att lägga till CSS-Klasser

```
<style>
  .cool{color:white; background-color: black;}
</style>
```

```
let p = document.getElementsByTagName('p');

for (let i = 0; i < p.length; i++) {
  p[i].className = 'cool';
}
```

Övning

```
<style>
  .nice{
    color:green;
  }
</style>
```

- Det finns flera rubriker (h1) i ett HTML-dokument.
- Leta efter den första rubriken och lägg till till CSS-klassen **nice**.
- OBS!
Det finns inget id på den första rubriken.

Lite om querySelector och querySelectorAll

```
<h1 id="one">Rubrik</h1>
<h1>Rubrik</h1>
<script>
let oneH1 = document.querySelector('#one');
console.log(oneH1);
```

querySelector
och
querySelectorAll
använder
CSS-syntax
(överkurs)

```
let allH1 = document.querySelectorAll('h1');
console.log(allH1);
</script>
```

Övning

```
<p>Stycke 1</p>
<p>Stycke 2.</p>
<p>Stycke 3</p>
```

- Skapa ett skript som går igenom alla stycken i ett dokument.
- Du ska undersöka om varje stycke avslutas med en punkt.
- Om inte så ska skriptet lägga till en punkt.

Övning

```
<p>Stycke 1.</p>
<p>stycke 2.</p>
<p>stycke 3.</p>
```

- Skapa ett skript som går igenom alla stycken i ett dokument.
- Du ska se till att alla stycken påbörjas med en stor bokstav.

VG-Övning

<p>första ordet i en ny mening. man sätter alltid stor bokstav på det första ordet i en ny mening. detta för att visa att en ny kommunikation startar.</p>

- Skapa ett skript som går igenom alla stycken i ett dokument.
- Du ska se till att alla meningar i alla stycken startar med en stor bokstav.

Övningar

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dom_html1

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dom_html2

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dom_html3

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dom_html6

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dom_html7

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dom_html8

Övning 4 och 5 är överkurs (HTML bilder och formulär).

Övning 9 handlar om events (nästa lektion).

Summering av dagens lektion

- **Vi har idag jobbat med DOM.**
- Reflektioner kring dagens lektion?
 - Vad tar du med dig från dagens lektion?
 - Finns det något som var extra svårt att förstå?
 - Finns det något som vi behöver repetera?
 - Hur upplevde du dagens arbetsmetoder?

Framåtblick inför nästa lektion

- Läs: Bok 2, sid. 183–207
- Under nästa lektion kommer vi att gå igenom händelsehantering i JavaScript (events).