

# 17. Intro - JavaScript i webbläsaren

# Om JavaScript

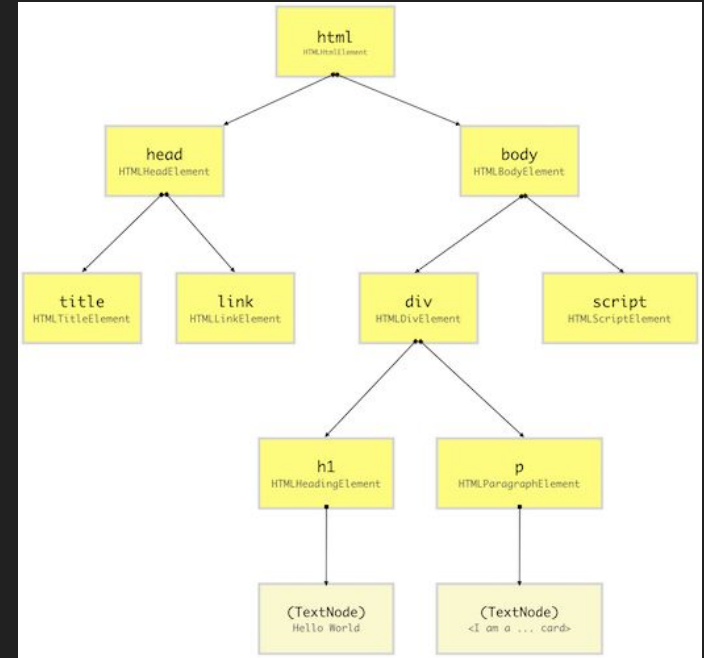
- JavaScript är det programmeringsspråk som används i en webbläsare.
- JavaScript kan också användas på andra ställen.
- JavaScript kan användas för att förändra en sida i webbläsaren.
  - Saker kan döljas/visas.
  - Information kan hämtas och “skrivas ut” i sidan.

# Om webbläsaren

- En webbläsare frågar en server om en HTML-sida.
- Därefter frågar webbläsaren om alla script, css:er, bilder mm.
- HTML läses uppifrån och ned.
- Så fort ett script laddats så körs det. Om det laddas in i <head> körs det innan hela sidan lästs in, det KAN skapa problem.
- Vartefter taggar läses skapas noder en objektstruktur i webbläsaren som heter Document Object Model (DOM).
- Webbläsaren ritar ut DOM när den har skapats.

# Document Object Model (DOM)

- DOMen kan ses som en trädstruktur.
- Den består av **noder**.
- Varje tag i HTML-koden skapar en nod.
- Varje nod (utom rotnoden) har en **parent**.
- En nod kan ha flera **children**.
- En nod kan ha syskon (**siblings**). Det är noder med samma parent.

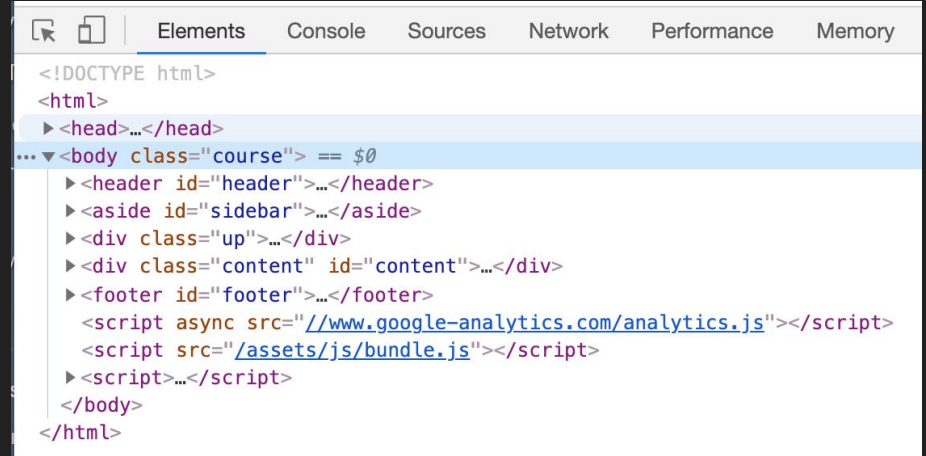
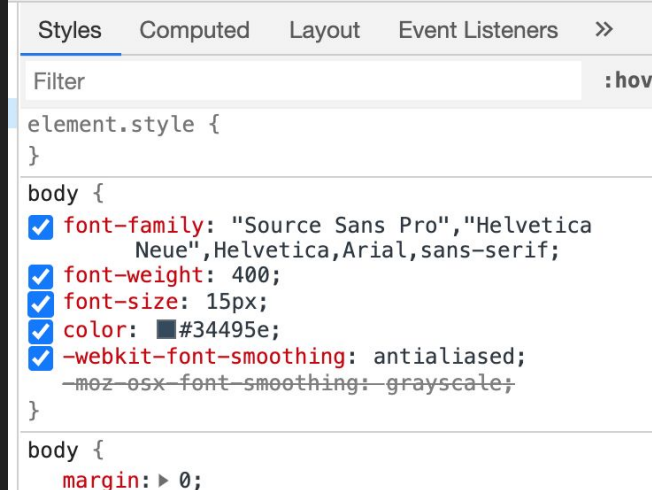


# Utvecklarverktygen i Webbbläsaren

- Öppnas med Ctrl+Shift+i,  
eller högerklicka och inspektera.

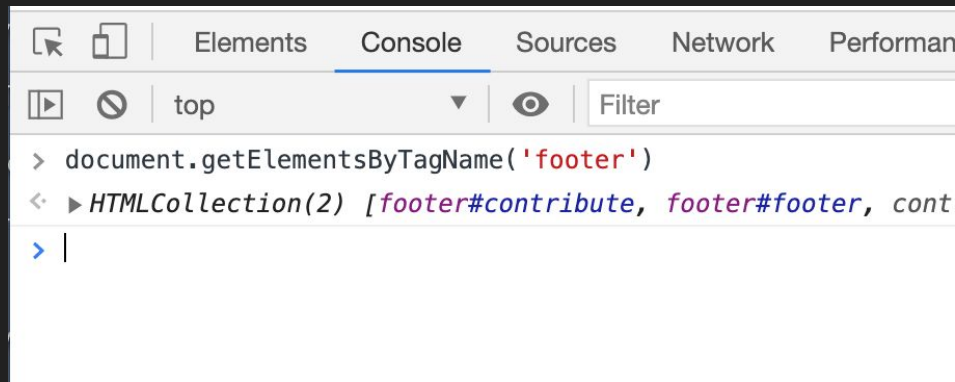
# Utvecklarverktygen i Webbläsaren

- Öppnas med Ctrl+Shift+i, eller högerklicka och inspektera.
- **Elements** innehåller DOMen, alla objekt och deras styles.



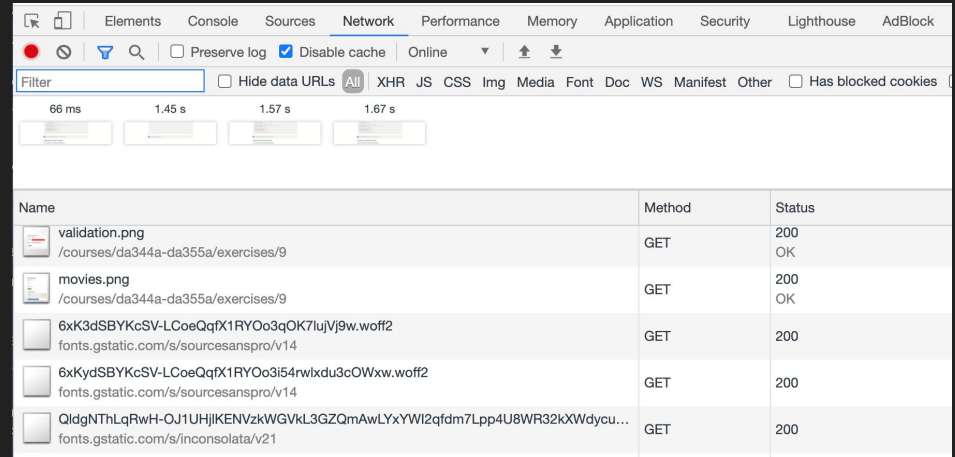
# Utvecklarverktygen i Webbläsaren

- Öppnas med Ctrl+Shift+i, eller högerklicka och inspektera.
- **Elements** innehåller DOMen, alla objekt och deras styles.
- **Console** innehåller bl.a. felmeddelanden. Här kan man skriva/köra JavaScript-kod.



# Utvecklarverktygen i Webbläsaren

- Öppnas med Ctrl+Shift+i, eller högerklicka och inspektera.
- **Elements** innehåller DOMen, alla objekt och deras styles.
- **Console** innehåller bl.a. felmeddelanden. Här kan man skriva/köra JavaScript-kod.
- **Network** innehåller alla requests webbläsaren gör.





# Webbläsare + JavaScript

- Ett JavaScript som laddats in i webbläsaren kan komma åt DOMen och göra förändringar i den.

## Webbläsare

```
1  <!DOCTYPE HTML>
2  <html>
3
4  <head>
5    <title>Rendering Test</title>
6
7    <!-- stylesheet -->
8    <link rel="stylesheet" href="./style.css" />
9  </head>
10
11 <body>
12   <div class="container">
13     <h1>Hello World!</h1>
14     <p>This is a sample paragraph.</p>
15   </div>
16
17   <!-- script -->
18   <script src="./main.js"></script>
19 </body>
20
21 </html>
```

# Webbläsare + JavaScript

- Ett JavaScript som laddats in i webbläsaren kan komma åt DOMen.

## Webbläsare

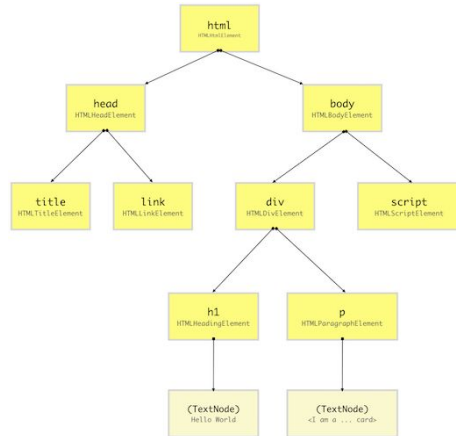
```
1  <!DOCTYPE HTML>
2  <html>
3
4  <head>
5    <title>Rendering Test</title>
6
7    <!-- stylesheet -->
8    <link rel="stylesheet" href="./style.css" />
9  </head>
10
11 <body>
12   <div class="container">
13     <h1>Hello World!</h1>
14     <p>This is a sample paragraph.</p>
15   </div>
16
17   <!-- script -->
18   <script src="./main.js"></script>
19 </body>
20
21 </html>
```

DOMen ska skapas.  
Wait for it!

# Webbläsare + JavaScript

- Ett JavaScript som laddats in i webbläsaren kan komma åt DOMen.

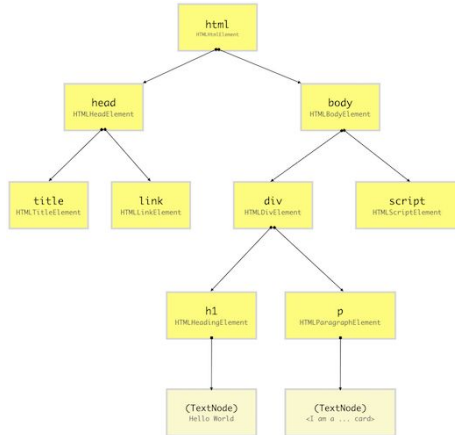
## Webbläsare



# Webbläsare + JavaScript

- Ett JavaScript som laddats in i webbläsaren kan komma åt DOMen.

## Webbläsare



JS main.js ×

json > JS main.js > ...

```
1 var headings = document.getElementsByTagName('h1')
```

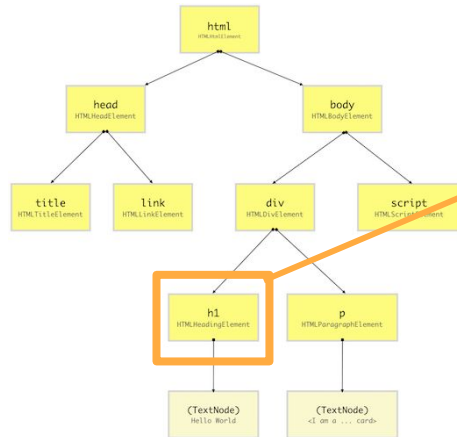
```
2
```

```
3
```

# Webbläsare + JavaScript

- Ett JavaScript som laddats in i webbläsaren kan komma åt DOMen.

## Webbläsare



JS main.js ×

json > JS main.js > ...

```
1 var headings = document.getElementsByTagName('h1')
```

# Webbläsare + JavaScript

- Ett JavaScript som laddats in i webbläsaren kan komma åt DOMen.
- Ett javascript laddas in med `<script>`. Ex:  
`<script src="scripts/app.js"></script>`
- Man kan ha flera egna JavaScript i en sida. T.ex.: main.js, events.js, validation.js, animations.js
- Man kan ha flera **externa** JavaScript i en sida. Tex: jQuery, popper, Bootstrap
- Funktioner och variabler i JavaScript-filer är globalt definierade som default. Det betyder att ett inladdat script kan anropa funktioner eller läsa/ändra variabler i ett annat inladdat script.

DEMO

# Felsökning

- Händer inget?
  - Kontrollera om det finns felmeddelande i Consolen i webbläsaren.
  - Kontrollera att du inkluderat ditt/dina JavaScript med `<script>`
  - Verifiera att du väntat på att DOMen är skapad i webbläsaren innan ditt script körs.
- Förstår du inte vad som händer?
  - Använd spårutskrifter med `console.log()`.



# Uppgift

1. Utgå från kod i <https://github.com/martinloman/js-start-template>
2. Ändra i funktionen som hanterar click-event på JS-loggan så att den gör följande:
  - a. sätter om sidans bakgrundsfärg genom att använda `document.body.style`. Googla hur man gör.
  - b. kan du skriva en funktion som slumpar bakgrundsfärgen varje gång man klickar på loggan? Ta google (eller AI) till hjälp.
  - c. tar bort logotypen från sidan. Du kan använda metoden `.remove()`.  
[https://www.w3schools.com/jsref/met\\_element\\_remove.asp](https://www.w3schools.com/jsref/met_element_remove.asp)
3. Hinner du ändå hit så titta på koden i <https://github.com/martinloman/js-add-to-dom> och testa att ändra saker i den.

# Videos

Om [DOM](#) (ca 5 min)

[Manipulera DOM med JavaScript](#) (ca 18 min)