

HTML DOM

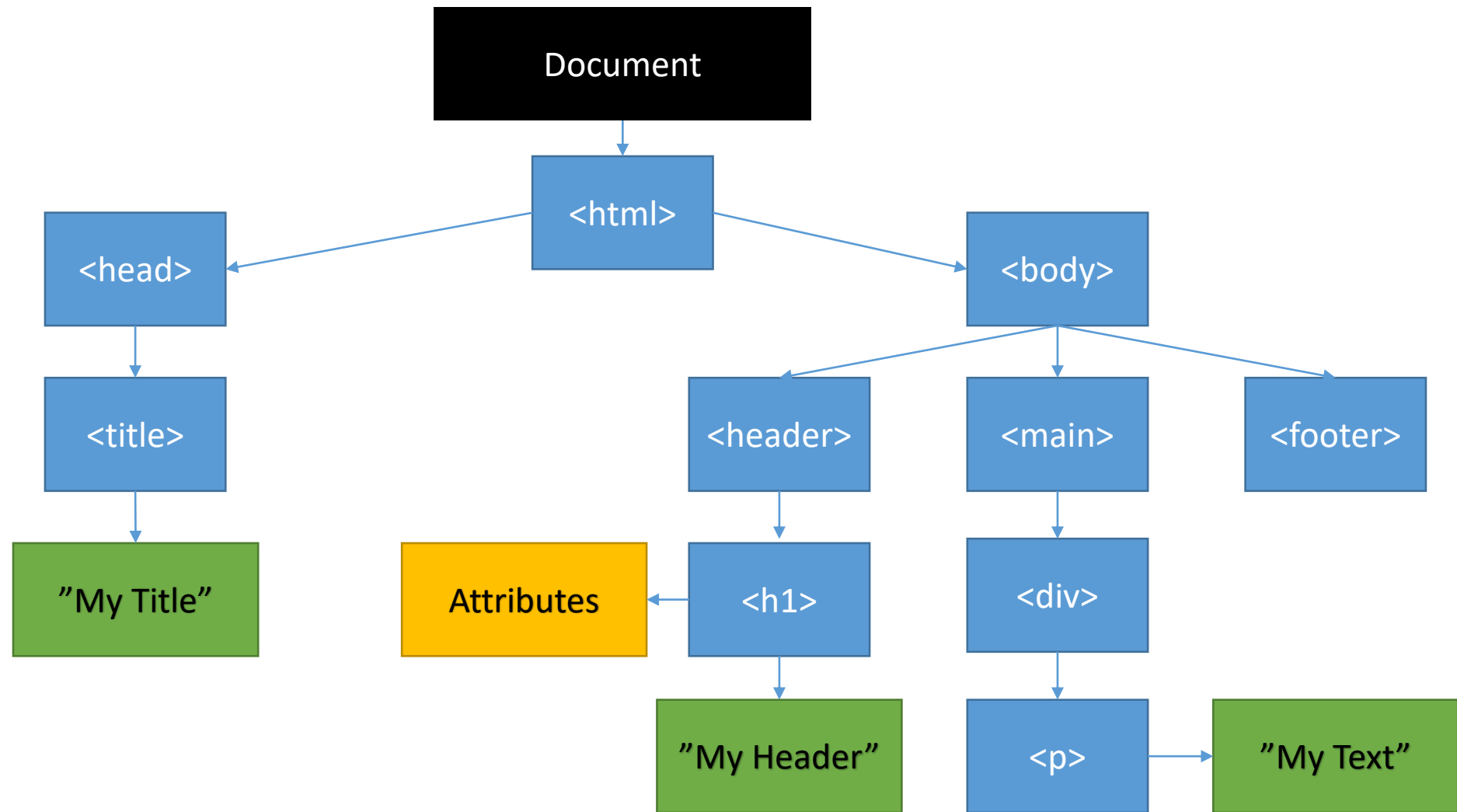
Document Object Model

Av: Namn



Vad är DOM

- DOM står för Document Object Model
- Detta är den modell som en webbläsare använder för att bygga upp grafiska objekt utifrån HTML
- DOM:en är alltså en beskrivning av alla objekt på en webbsida.
- DOM:en kan beskrivas med ett diagram som ofta kallas DOM-tree.
- Ett DOM-Tree liknar ett klassdiagram



Relationer mellan element

- Ett html-element kan ha tre olika relationer till andra element
 - Parent
 - Child
 - Sibling
- Parent är det element som har det aktuella elementet i sig.
- Child är ett element som ligger i det aktuella elementet
- Sibling är ett element som ligger i samma Parent.

DOM-manipulation

- Med hjälp av JavaScript kan vi manipulera (ändra) DOM:en.
- JavaScript kan...
 - Ändra alla html-element
 - Ändra alla html-attribut
 - Ändra all css-styling
 - Lägga till nya och bort befintliga html-element
 - Reagera på händelser(event) på sidan
 - Skapa nya händelser(event)

Att hämta element från DOM

- Det finns flera sätt att hämta element ifrån dokumentet.
- För att göra någon form av manipulation använder vi objektet **document**
- Det första sättet att hämta ett element vi ska kika på är: **getElementById()**
- Med hjälp av getElementById kan vi hämta ett specifikt element baserat på dess Id

InnerText och InnerHTML

- InnerText låter oss hämta och ändra text i ett element

```
const paragraph=document.getElementById("paragraph");  
console.log(paragraph.innerText);  
paragraph.innerText = "Ändrad text";
```

- InnerHTML låter oss hämta och ändra i html-dokumentet.

```
const div=document.getElementById("div");  
console.log(div.innerHTML);  
div.innerHTML = "<h1>Ny h1</h1>";
```

Övning 1

- Skapa ett HTML-dokument med följande struktur i body:
 - `<header>`
 - `<h1 id: header1`
 - My Header
 - `<main>`
 - `<div>`
 - `<p id: paragraph`
 - My Text
 - `<footer>`
 - `<a href: google.com`
- Lägg till en `script.js`
- Hämta och ändra texten i `h1`:an med hjälp av JavaScript.
- Hämta och ändra texten i `p`-taggen till Lorem...

Övning 2

- Använd koden från Övning 1.
- Med hjälp av innerHTML, byt ut a-taggen i footer mot en lista av tre a-taggar med följande länkar:
 - Google.com
 - W3schools.com
 - Youtube.com

Demo

Eller rast?

Andra sätt att hämta element

- `document.getElementsByClassName` : Hämtar alla element med en given class
 - `document.getElementsByName` : Hämtar alla element med ett givet namn
 - `document.getElementsByTagName` : Hämtar alla element av en viss typ
-
- Alla dessa returnerar en HTMLCollection, även om det bara finns ett eller inga alls.
 - En HTMLCollection är inte en array, alltså den har inte Array-metoder.
 - En HTMLCollection låter oss leta i den med index, id eller namn

Children och Parent

- Alla element har en property med en lista av child-element som heter children.
- Alla element har två properties för sin parent. parentNode och parentElement.
- Notera att dessa ibland kan vara samma. En är egentligen raden ovanför det aktuella elementet. Det kan vara element, men också kommentarer eller text.
- parentElement returnerar alltid html-elementet som innehåller det aktuella elementet.

Övning 3

- Använd koden från övning 1&2
- Lägg till 2 divar i main med klassen "myClass"
- I båda divarna lägg till en p-tagga med Lorem..
- Hämta båda divarna med hjälp av `getElementsByClassName`
- Loopa igenom resultatet och ändra texten i p-taggar.
 - Loopa igenom `Children` för att hitta p-taggar

Demo

Skapa element

- För att skapa element kan man använda createElement()
- createElement tar in namnet på ett element den ska skapa.

```
const nyH1 = document.createElement("h1");  
nyH1.innerText = "Apa";  
document.body.append(nyH1);
```

Lägga till element

- AppendChild kan användas för att lägga till element inuti det aktuella elementet.
- AppendChild tar emot ett objekt skapat med createElement
- Man kan också flytta element med hjälp av detta.

```
const test = document.createElement("h1");  
test.innerText = "NUSÅ!";  
div.appendChild(paragraph);
```


InsertBefore

- Om man inte vill att elementet ska läggas till sist i children för ett parent-element med appendChild kan man använda insertBefore()
- insertBefore() tar in ett element som ska läggas till och det element det ska läggas till innan.

```
<ul id="lista">
  <li id="item1">item1</li>
  <li id="item2">item2</li>
</ul>
```

```
const lista = document.getElementById("lista");

const item0 = document.createElement("li");
item0.id = "item0";
item0.innerText = "item0";

const item1 = document.getElementById("item1");

lista.insertBefore(item0, item1);
```

Demo

Eller Lunch?

Övning 4

- Utgå ifrån koden från övningarna innan
- I script.js, lägg till följande array:
 - "Hund", "Katt", "Kanin"
- Använd createElement och appendChild för att skapa en <div>
- Använd sedan createElement för att skapa en
- Med hjälp av en loop, lägg till en för varje text i arrayen du skapade tidigare som innehåller den texten.

Events

- I JavaScript kan vi lägga till eller aktivera events (händelser)
- Exempel på event:
 - När sidan har laddat
 - När en bild laddat
 - När en användare klickar på något
 - När en användare håller musen över något
 - När en användare trycker på en tangent

Hur används ett event?

- Ett event tar som argument in en funktion som ska utföras när det inträffar.
- De fungerar väldigt likt event och commands i wpf.

onclick, onmousedown, onmouseup

- Eventet **onclick** kan användas för att göra något om användaren trycker på ett specifikt element.
- Eventet **onmousedown** körs när man trycker ner vänster musknapp på ett element
- Eventet **onmouseup** körs när man släpper musknappen från ett element.

onmouseover och onmouseout

- Eventet **onmouseover** körs när muspekaren dras in över ett element
- Eventet **onmouseout** körs när muspekaren dras ut ifrån ett element

Övning 5

- Skapa ett nytt html-dokument
- Länka till script.js
- Lägg till header, main och footer
- I header lägg till en h1 med id header1 och texten "Min första knapp!"
- I main lägg till en button med id minKnapp
- I main lägg också till en h2 med id counter
- I script.js lägg till en variabel som heter counter som har värdet 0
- I script.js lägg till ett onclick-event på din knapp som ökar värdet på counter med ett och sedan sätter innerText på din h2 med id counter till värdet av variabeln counter.

Utmaning!

- Undersök om du kan ändra färg på h1 från tidigare övning när musen hålls över den.
- Försök lägga till en ul med id `todo`, ett input-fält och en till knapp och se om du kan få texten som skrivs in i input-fältet att läggas till i listan när du trycker på knappen.