



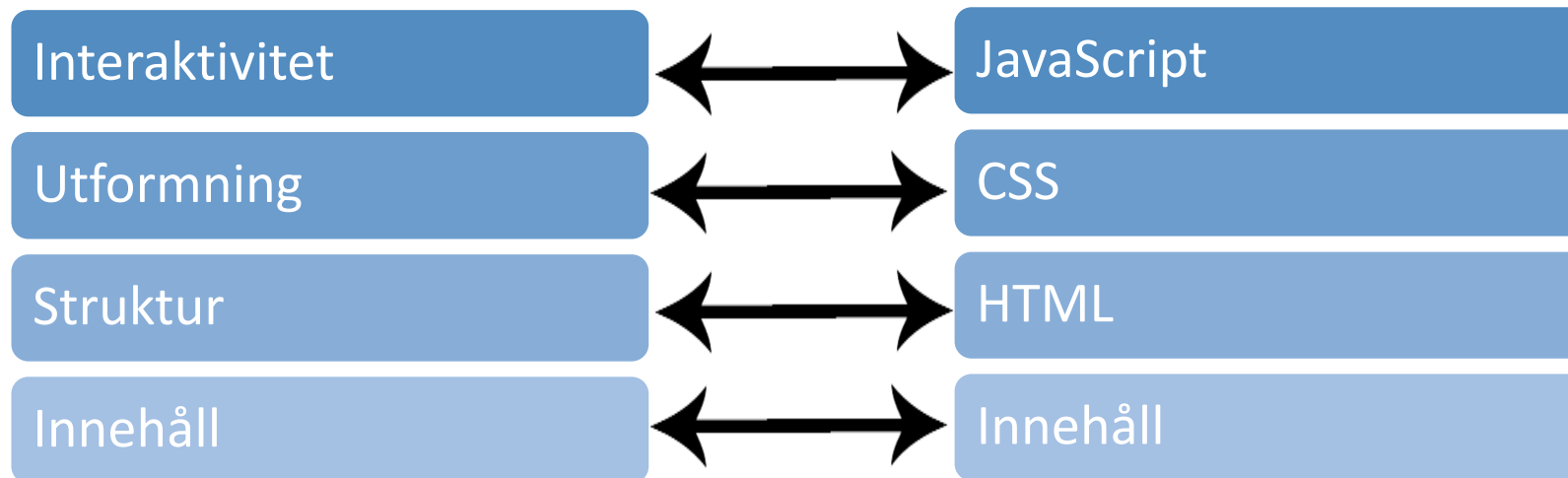
JavaScript

# Introduktion till JavaScript

# Dagens föreläsning

- JavaScript?
- JavaScript – syntax?
- JavaScript – att leta upp element i HTML-dokumentet
- JavaScript – att manipulera element i HTML-dokumentet
- JavaScript – exempel

# Var passar JavaScript in?





JavaScript

# JavaScript lever i din webbläsare

Allt som behövs för att köra JavaScript är en webbläsare – smidigt värre!

# Har ni sett det innan?

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Filmtajm</title>
<link href="css/style.css">
<link href="plugins/selec">
<link href="bootstrap/css">
<script type="text/javas">
<style type="text/css">
<script type="text/javas">
<script type="text/javascript" src="plugins/selec">
<script type="text/javascript" src="bootstrap/js/bootstrap.min.js">
<script type="text/javascript" src="bootstrap/js/Sortable/src/jquery.tinysort.min.js">
<script type="text/javascript" src="plugins/tinysort/src/jquery.tinysort.charorder.min.js">
<script type="text/javascript" src="plugins/tinysort/libs/jquery.opensource.min.js">
<script type="text/javascript" src="plugins/tinysort/libs/jquery.opensource.min.js">
<script type="text/javascript" src="plugins/quicksand/quicksand.js">
<script type="text/javascript" src="plugins/highCharts/js/highcharts.js">
<script type="text/javascript" src="js/script.js">
</script>
</head>

```

# JavaScript – varför?

- Vi vill ge liv till annars statiska webbsidor!
- T.ex.
  - Automatiskt modifiera dynamiskt innehåll
    - Facebook
    - Twitter
    - Gmail
  - Diverse beräkningar
    - T.ex. valutakonverteringar
  - Förbättrade GUI
    - Animeringar
    - Lägga till/ta bort/modifiera befintliga HTML-element

# JavaScript - exempel

- Bildspel: <http://wowslider.com/javascript-slideshow-quiet-rotate-demo.html>
- Dynamisk innehåll: <http://gmail.com>, <http://facebook.com>, <https://twitter.com/>
- Beräkningar: <http://valuta.se>
- Validera formulär: <http://yensdesign.com/tutorials/validateform/>
- Dialogrutor: <http://tibbelit.se/film>
- Interaktiva element: [http://mrdoob.com/projects/chromeexperiments/ball\\_pool/](http://mrdoob.com/projects/chromeexperiments/ball_pool/)
- Spel: <http://www.phoboslab.org/ztype/>
- Animationer: <http://codepen.io/juliangarnier/pen/idhuG>
- och mycket mera!

# JavaScript - Historik

- Utvecklades av Netscape under mitten av 90-talet
- Har inget med Java att göra – det är ett eget programmeringsspråk
- Microsoft skapade sin egen version av JavaScript – Jscript
- På senare dagar (2005) har ajax blivit en populärt
- Läger till extra funktionalitet för webben och webbläsare



# Vad kan man göra med JavaScript?

- I princip vad du vill!
- Men det används ofta till att skapa interaktiva webbsidor
- Med interaktiv menas: Reagera på användaras handlingar och modifiera webbsidans innehåll

# JavaScript

- JavaScript skrivs som ren text
- Det kan skrivas antingen inbäddat direkt i HTML-koden
- Eller i en separat fil, som sedan inkluderas i HTML-dokumentet

# JavaScript - Var skrivs det?

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Filmtajm</title>
    <link href="css/style.css" rel="stylesheet" type="text/css">
    <link href="plugins/select2/select2.css" rel="stylesheet" type="text/css">
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet" type="text/css">
    <script type="text/javascript" src="jqueryUI/js/jquery-1.8.0.min.js"></script>
    <script type="text/javascript" src="jqueryUI/js/jquery-ui-1.8.23.custom.min.js"></script>
    <script type="text/javascript" src="plugins/select2/select2.min.js"></script>
    <script type="text/javascript" src="bootstrap/js/bootstrap.min.js"></script>
    <script type="text/javascript" src="plugins/tinySort/src/jquery.tinysort.min.js"></script>
    <script type="text/javascript" src="plugins/tinySort/src/jquery.tinysort.charorder.min.js"></script>
    <script type="text/javascript" src="plugins/tinySort/libs/jquery.opensource.min.js"></script>
    <script type="text/javascript" src="plugins/quicksand/quicksand.js"></script>
    <script type="text/javascript" src="plugins/highCharts/js/highcharts.js"></script>
    <script type="text/javascript" src="js/script.js"></script>
  </head>
```

---

```
<script>
  /*
   | This function is called when the document has finished loading.
   */
  function start(){
    /*
     | On newNoteForm submit.
     */
    $("form").first().submit(function(e){
      e.preventDefault();
```

# JavaScript som programmeringsspråk

- Ett högnivåspråk
  - Ett otypat språk
  - Ett skriptspråk
- 
- Ni kan programmera, så efter lite träning med syntaxen så borde det inte vara några problem att komma igång =)

# Exempel på syntax

```
var name = "Anton";  
var year = 2016;  
  
if(a == b){  
    // If true  
}else{  
    // If else  
}  
  
function myFunction(arg1, arg2){  
    // Do something fun with arg1 and arg2!  
}  
  
// Count 10 times  
for(var i = 0; i < 10; i++){  
    // Do something fun  
}  
  
var index = 0;  
while(index < 10){  
    // Do something fun  
    i++;  
}
```

Demo för JavaScript

# Tillgänglighet

Vad händer om man har inaktiverat JavaScript i webbläsaren?

# Inaktiverat JavaScript?

- Man kan inaktivera det manuellt
- På vissa företag inaktiveras det
- Gamla webbläsare stödjer ev. inte JavaScript fullt ut



# Hur löser vi det?

*Providing an alternative version of your functionality or making the user aware of shortcomings of a product as a safety measure to ensure that the product is usable.*

- Chrisitan Heilmann

# Med andra ord...

1. Bygg webbplatser för moderna webbläsare
2. Identifiera kritisk funktionalitet och
  1. Se till att detta fungerar i alternativa webbläsare
  2. Berätta för användaren att något är fel och föreslå en metod för att fixa detta

# Exempel

```
<a href="javascript:window.print()">Print page</a>
```

- Fungerar inte utan JavaScript

# Hur löser vi det?

```
<a href="javascript:window.print()">Print page</a>
```

```
<noscript>
```

```
  <p class="scriptwarning">
```

```
    Printing the page requires JavaScript.
```

```
    Please turn it on in your browser.
```

```
  </p>
```

```
</noscript>
```

# Är det den smidigaste lösningen?

- Vad är JavaScript?
- Hur aktiverar jag det?
- Får inte använda JavaScript (företagsmiljö),
- kan inte använda JavaScript
- Aktivera JavaScript bara för att skriva ut?

# Tänk på användaren

```
<a href="javascript:window.print()">Print page</a>
```

```
<noscript>
```

```
  <p class="scriptwarning">
```

Print a copy of your confirmation.

Select the "Print" icon in your browser,  
or select "Print" from the "File" menu.


```
  </p>
```

```
</noscript>
```

# Att hitta element genom JS


- En stor fördel med JS är att vi kan leta upp HTML-element!
- Vi kan leta upp detta genom att:
  - Ange typen av element (t.ex. alla <p>-element)
  - Ange klassen för ett element (t.ex. alla element med klassen "blue")
  - Ange ett id för ett element
- Jämför detta med hur vi använder CSS för att hitta våra element.

# Att hitta ett HTML-element genom JavaScript

```
1   /*  
2      Hämta det första elementet som matchar selektorn  
3  */  
4  var htmlElement = document.querySelector("selektor");  
5  
6  // Ex.1, Hämta elementet med id "start"  
7  var el = document.querySelector("#start");  
8  // Ex.2, Hämta elementet med id "footer"  
9  var el = document.querySelector("#footer");  
10 // Ex.3, Hämta det första element som har klassen "big-image"  
11 var el = document.querySelector(".big-image");  
12 // Ex.4, Hämta den första paragrafen på sidan  
13 var el = document.querySelector("p");  
14
```



## Att hitta flera HTML-element genom JavaScript

```
16   /*  
17      Hämtar alla elementet som matchar selektorn  
18  */  
19  var htmlElement = document.querySelectorAll("selektor");  
20  
21  // Ex.1, Hämta alla paragrafer på sidan  
22  var el = document.querySelectorAll("p");  
23  // Ex.2, Hämta alla <h2>-rubriker på sidan  
24  var el = document.querySelectorAll("h2");  
25  // Ex.3, Hämta det alla element som har klassen "big-image"  
26  var el = document.querySelectorAll(".big-image");  
27  // Ex.4, Hämta det alla element som har klassen "center"  
28  var el = document.querySelectorAll(".center");
```

# Användardrivet GUI

- Vi "lyssnar" efter användarens handlingar, som t.ex.
  - Klick
  - Musrörelser
  - Svepningar
  - M.m.
- Vi kan sedan koppla ihop dessa handlingar med element – och sedan koppla detta till en funktion

```
<script>
// Användaren klickar på ett element med id:t "start"
document.querySelector("#start").addEventListener("click", myFunction, false);
function myFunction() {
    // Körs när användaren klickar på elementet med id:t "start"
}
</script>
```

# Att manipulera element genom JavaScript

- När vi i JavaScript letat upp ett element kan vi modifiera det
  - Ta bort det
  - Ändra ev. innehåll (text)
  - Ändra CSS-egenskaper
  - Ändra/lägga till/ta bort attribut

```
<p id="firstP" class="center">Lite text</p>
```

# Exempel

## HTML

```
<p id="firstP" class="center">Lite text</p>
```

## JavaScript

```
var p = document.querySelector("#firstP");  
// Hämta ut olika värden  
console.log(p.id); // Loggar => "firstP"  
console.log(p.className); // Loggar => "center"  
console.log(p.innerHTML); // Loggar => "Lite text"  
  
// Sätter nya värden  
p.id = "newId"; // Sätter elementets id till "newId"  
p.className = "newClass" // Sätter elementets class till "newClass"  
p.innerHTML = "Ny text"; // Sätter texten i elementet till "Ny text"  
  
// Stylar elementet  
p.style.color = "blue"; // Ändrar textfärgen till blå  
p.style.fontSize = "20px"; // Ändrar fontstorleken till "20px"
```

# Att modifiera element

- Vi kan ändra attribut för element
- Vi kan ändra CSS för element
- Vi kan ändra text för element
- Vi kan lägga till nya element
- Vi kan ta bort element

# Exempel på att modifiera element

```
<script>  
// Ändra ett attribut för ett element  
element.setAttribute("name", "value");  
// Ändra CSS-egenskap för ett element  
element.style.color = "black";  
// Lägg till ett element  
element.appendChild(node);  
// Ta bort ett element  
parent.removeChild(child);  
</script>
```

# ”this” – att referera till elementet som utlöste händelsen

- ”this” som nyckelord är väldigt händigt att hantera – på så sätt behöver vi inte ”leta upp” elementet som utlöste händelsen mer än en gång.

```
<script>
// Användaren klickar på ett element med id:t "start"
document.querySelector("#start").addEventListener("click", myFunction, false);
function myFunction() {
    // Byter textfärg på elementet #start
    this.style.color = "red";
}

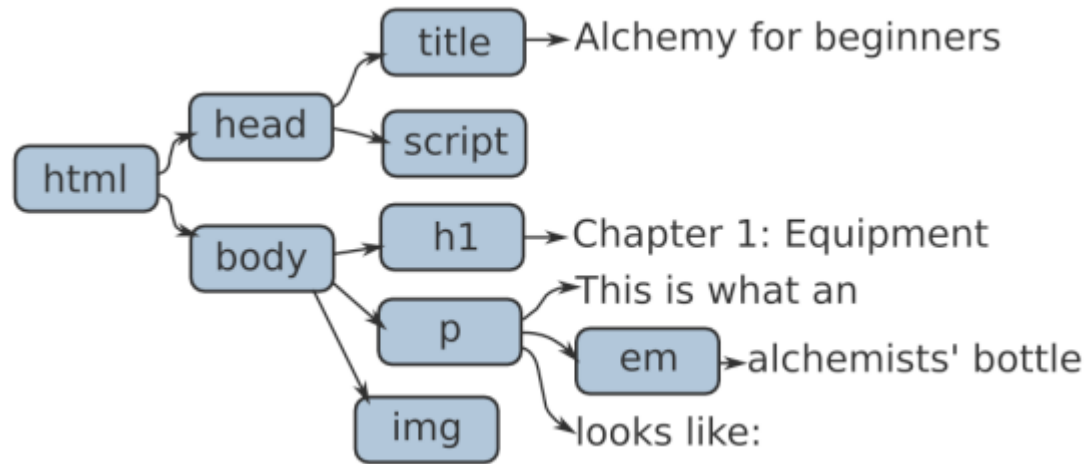
// Hitta alla element av typen "p"
var p = document.querySelectorAll("p");
for(var i = 0; i < p.length; i++){
    p[i].addEventListener("click", colorize, false);
}
function colorize() {
    // Ändrar färg på (bara) den paragrafen som vi klickade på
    this.style.color = "red";
}
</script>
```

# Exempel med JavaScript

Hitta & manipulera element



# DOM

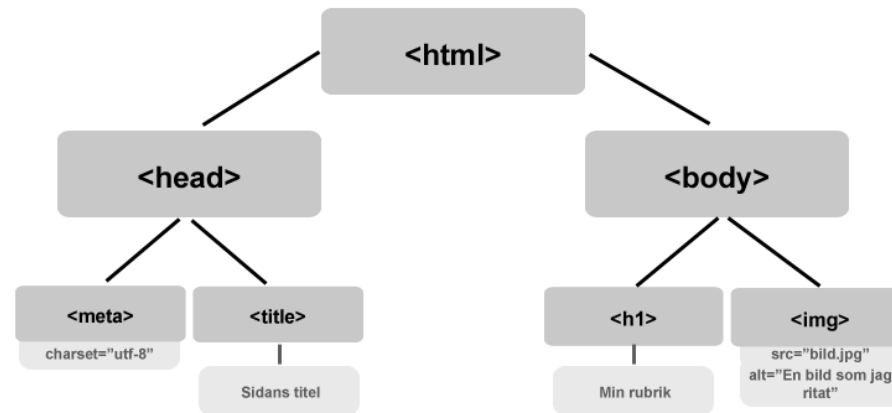


- Trädstruktur som motsvarar taggarnas ordning/nästling
- Varje element är en "nod" i trädet
- Relationer mellan noderna beskrivs med förälder/barn och syskon

# Källkod vs. noder

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sidans titel</title>
  </head>

  <body>
    <h1>Min rubrik</h1>
    
  </body>
</html>
```



# Använda DOM

- I JavaScript kan man alltid komma åt DOM via objektet "document"
- Därifrån kan vi "navigera" genom trädet

*// referens till body-elementet:*  
**document.body**

*// lista över alla element i body*  
**document.body.childNodes**

# Att navigera i DOM

- Vi antar att "node" är en nod/element i DOM-trädet

```
// Navigera mellan noder på samma nivå  
node.nextSibling  
node.previousSibling
```

```
// Nedåt, barnnoder  
node.firstChild  
node.lastChild  
node.childNodes // alla barn
```

```
// Uppåt, förälder  
node.parentNode
```

# Att loopa igenom alla element

- Hämta en mängd med element (en lista med element):
  - `document.querySelectorAll`
- "Loopa" igenom alla element som returneras, ex.

---

```
var p = document.querySelectorAll("p");  
for(var i = 0; i < p.length; i++){  
    p[i].onclick = myFunction;  
}
```

# Att hämta värde från formulär

```
<label for="name">Namn: </label>
<input type="text" id="name">
<button id="showName">Visa namn i popup-ruta</button>

<script>
// Genom anonym funktion
document.querySelector("#showName").addEventListener("click", function(){
    var name = document.querySelector("#name").value;
    alert("Hej " + name + "!");
}, false);

// Samma sak
document.querySelector("#showName").addEventListener("click", showName, false);
function showName(){
    var name = document.querySelector("#name").value;
    alert("Hej " + name + "!");
}
</script>
```

Exempel - DOM