



JavaScript

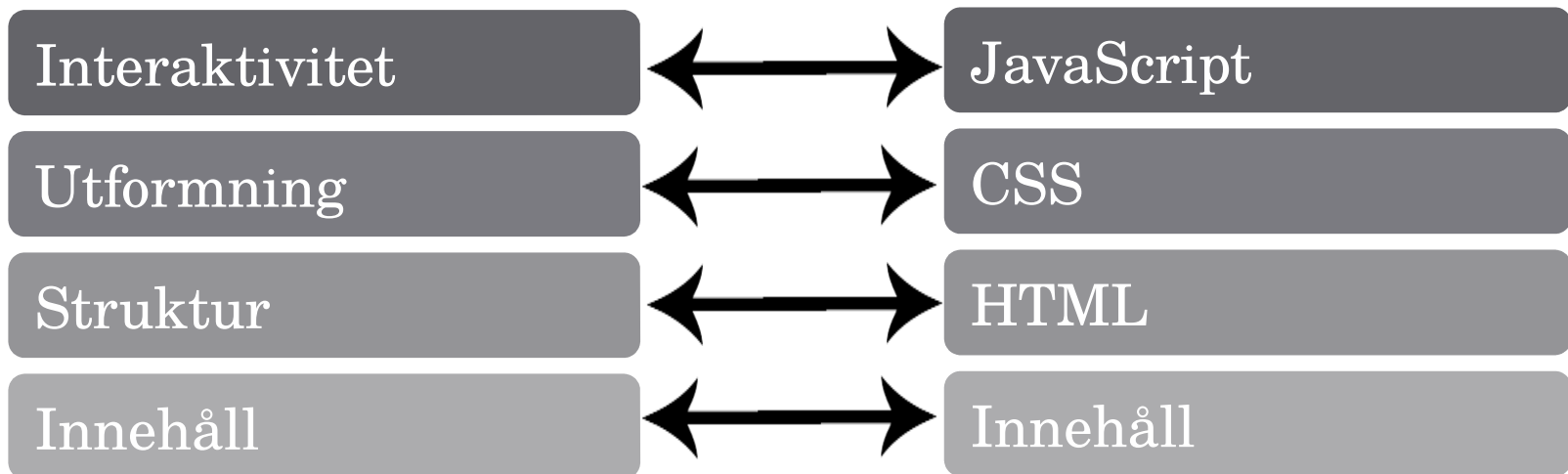
JavaScript

... och JavaScript-ramverket jQuery

Dagens föreläsning

- JavaScript?
- JavaScript – syntax?
- JavaScript – att leta upp element i HTML-dokumentet
- JavaScript – att manipulera element i HTML-dokumentet
- JavaScript – exempel
- jQuery – Ett av många JavaScript-ramverk för front-end
- jQuery – syntax?
- jQuery – Exempel

Var passar JavaScript in?



JavaScript lever i din webbläsare

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

Har ni sett det innan?

```
▼ <html>
  ▼ <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Filmtajm</title>
    <link href="css/style.css" type="text/css" rel="stylesheet">
    <script type="text/javascript">
      $(document).ready(function(e) {
        $(".entry img").parent().hide();
        $(".entry h6").hide();
      });
    </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>
  ▼ <body>
```

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

- <http://valuta.se>
- <http://yensdesign.com/tutorials/validateform/>
- <http://www.phoboslab.org/ztype/>

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 (används inte längre)
- Är en av drivkrafterna bakom AJAX
- 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 användarens 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++;  
}
```

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.

Exempel på att hitta element

```
<script>  
// Hitta alla element av typen "p"  
document.querySelectorAll("p");  
// Hitta alla element med klassen "blue"  
document.querySelectorAll(".blue")  
// Hitta elementet med id:t "start"  
document.querySelector("#start")  
</script>
```

Användardrivet GUI

- Vi "lyssnar" efter användarens handlingar, som t.ex.
 - Klick
 - Musrörelser
 - Svepningar
 - mm.
- 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 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 hädigt att hantera – på så sätt behöver vi inte "leta upp" elementet som utlöste händelsen mer än en gång.

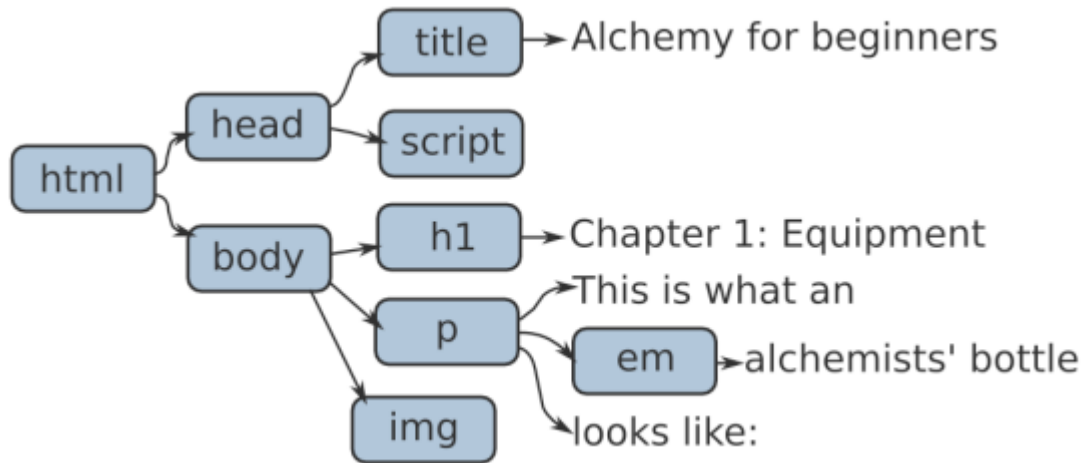
```
<script>
// Användaren klickar på ett element med id:t "start"
document.querySelector("#start").addEventListener("click", myFunction, false);
function myFunction() {
    // Byter textfärg på elementet #start
    this.style.color = "red";
}

// 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

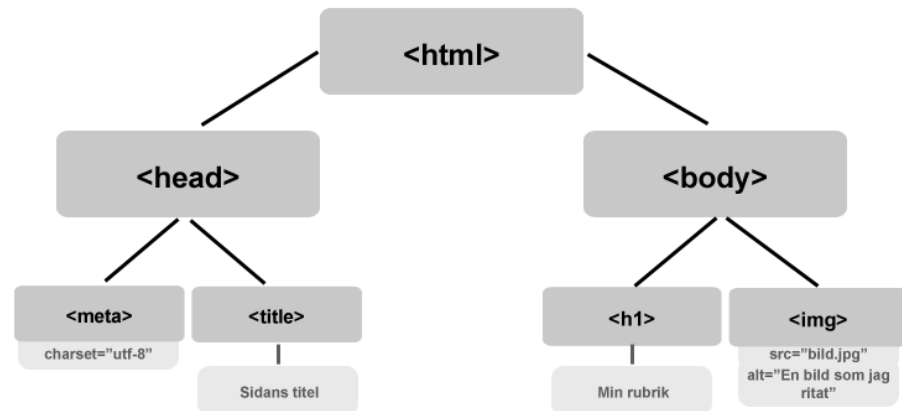


- Trädstruktur som motsvarar taggarnas ordning/nästling
- Varje element är en ”nod” i trädet
- Relationer mellan noderna beskrivs med förälder/barn och syskon

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`

Exempel - DOM

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>
```

jQuery – ett JS- ramverk

Varför ramverk?

- Snabbare utveckling
- Inbyggda funktioner
- Animeringar och andra effekter
- Cross-browser lösningar

jQuery – Referera till element

- Att referera till element ("CSS"-referenser)
Utan jQuery: `document.querySelectorAll("#id");`
Med jQuery: `$("#id");`
- Att referera till taggar
Utan jQuery: `document.querySelectorAll("a");`
Med jQuery: `$("a");`
- Att referera till klasser
Utan jQuery: `document.querySelectorAll(".blue");`
Med jQuery: `$(".blue");`

jQuery – navigera i DOM

- `Document.getElementById("id").nextsibling;`
`$("#id").next();`
- `Document.getElementById("id").childNodes;`
`$("#id").children();`
- `Document.getElementById("id").parentNode;`
`$("#id").parent();`

Att använd funktioner i jQuery

- Att dölja ett element:
`document.getElementById("id").style.display = "none";`
`$("#id").hide();`
- Att tona in ett element (animering):
`$("#id").fadeIn(1000);`

jQuery - dokumentation

- <http://api.jquery.com/>