

# JavaScript 2

---

Forelesning 11, Webprosjekt H2013

Alfred Bratterud



# Plan for dagen

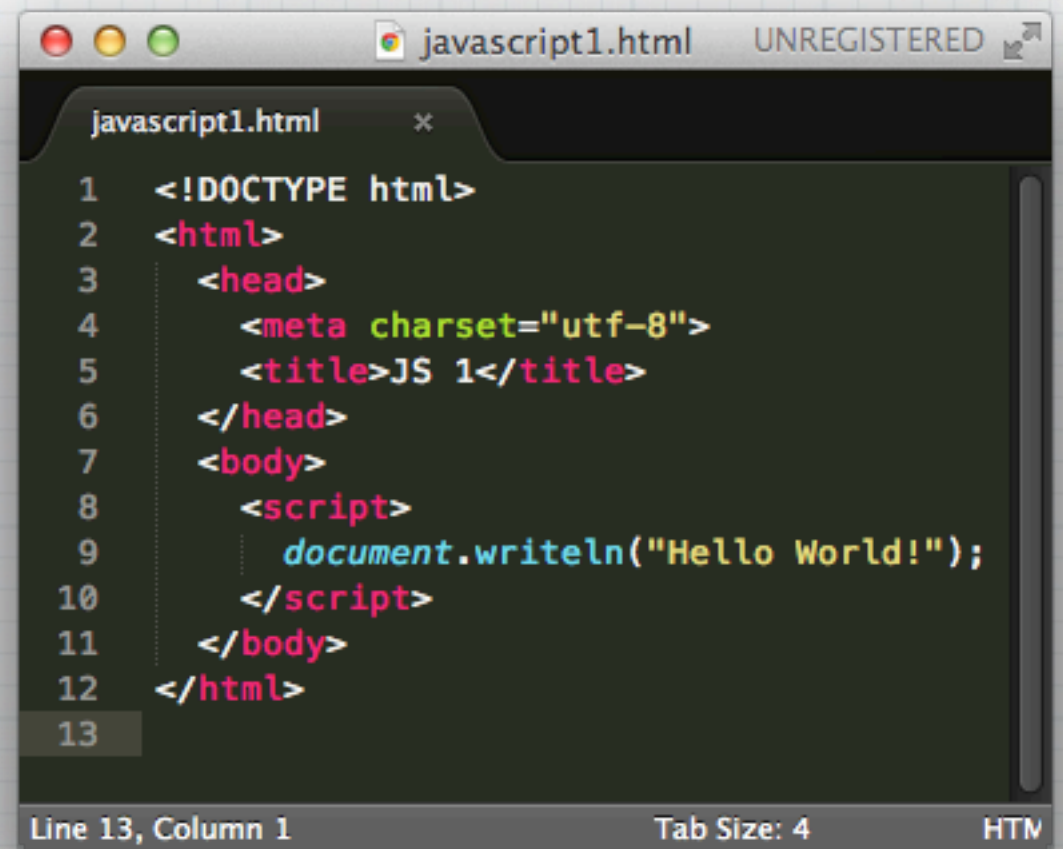
- \* JS Basics: Repetisjon
- \* Mer Document Object Model
- \* JQuery
- \* Mer sjakk!



# JavaScript

## "Hallo Verden"

- \* Javascript er et "interpretet" "scriptspråk"
- \* Kjører inni browseren
- \* Har ikke lese/skrivetilgang utenfor browser
- \* Kun nettverkstilgang tilbake der det kom fra



```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>JS 1</title>
6    </head>
7    <body>
8      <script>
9        document.writeln("Hello World!");
10     </script>
11   </body>
12 </html>
13
```

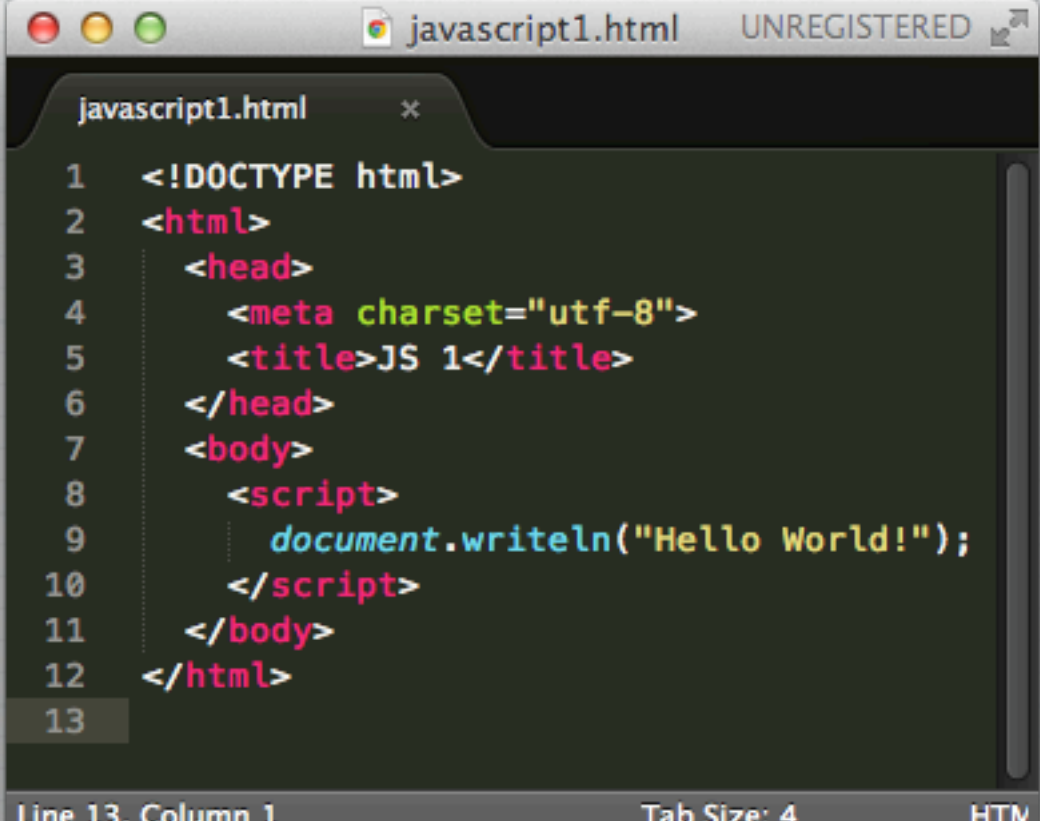
Line 13, Column 1      Tab Size: 4      HTM



# JavaScript

## "Hallo Verden"

- \* Primært formål; dynamisk HTML
- \* Gjør Ajax / "Web 2.0" mulig:
- \* HTML-gui som ikke sendes på nytt ved hvert klikk
- \* Få fellestrekk med java



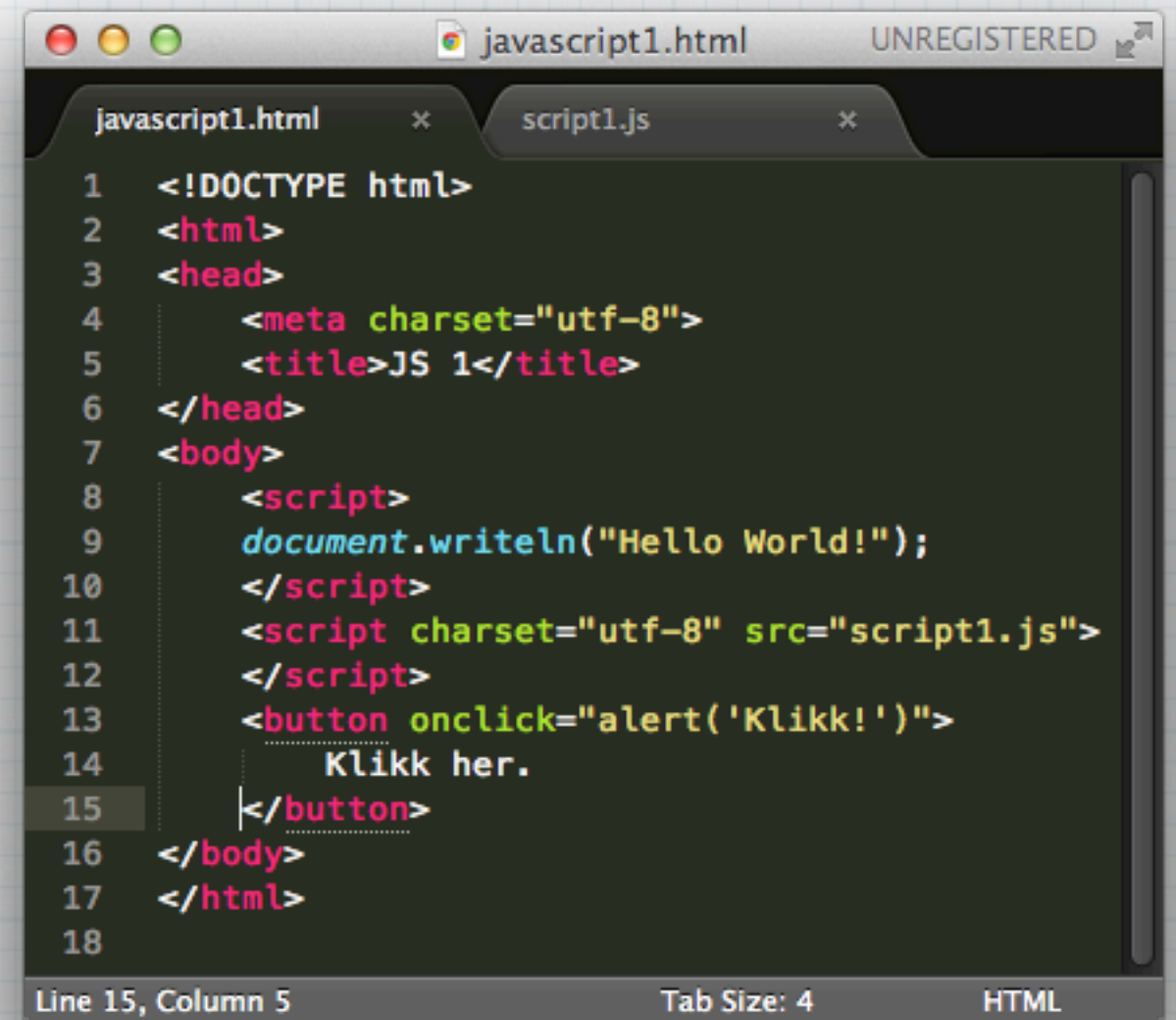
```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>JS 1</title>
6    </head>
7    <body>
8      <script>
9        document.writeln("Hello World!");
10     </script>
11   </body>
12 </html>
13
```

Line 13, Column 1      Tab Size: 4      HTM



# “Trigge” JavaScript

- \* I Script-tag; kjører ved innlasting
- \* Kan referere til egen fil via “src”
- \* Tegnsett bør da oppgis. Hvorfor?
- \* Kan startes via “HTML-events”, feks. “onclick”



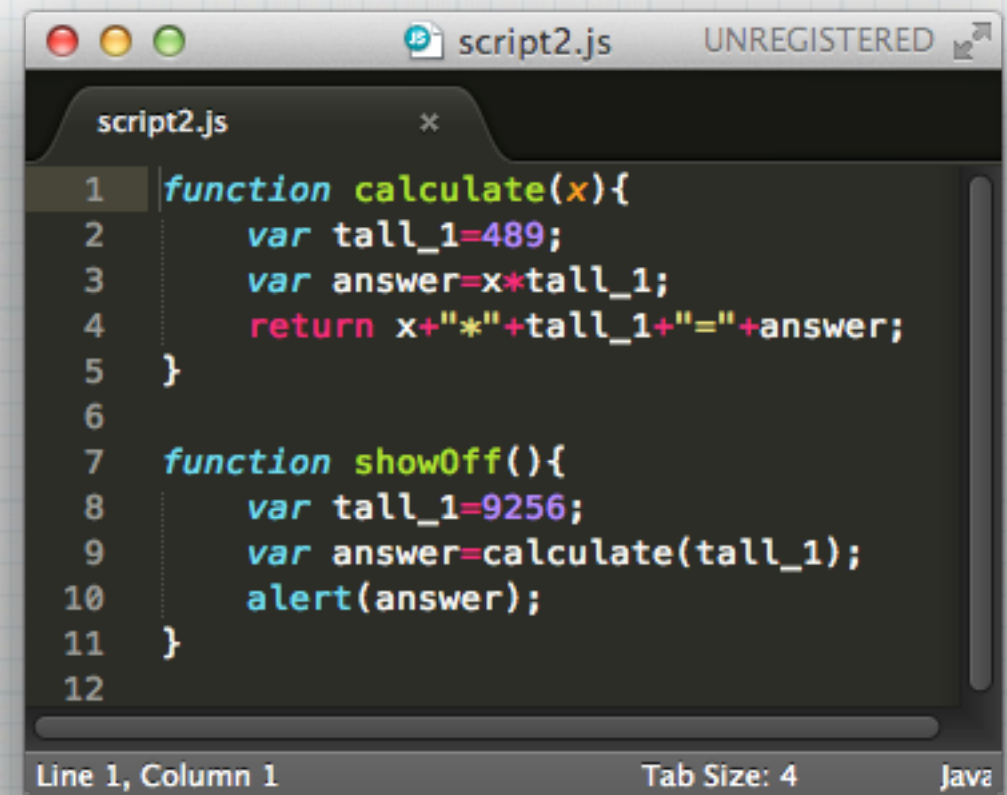
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>JS 1</title>
6  </head>
7  <body>
8      <script>
9          document.writeln("Hello World!");
10     </script>
11     <script charset="utf-8" src="script1.js">
12     </script>
13     <button onclick="alert('Klikk!')">
14         Klikk her.
15     </button>
16 </body>
17 </html>
18
```

Line 15, Column 5      Tab Size: 4      HTML



# JS-basics: Funksjoner

- \* Hva er en funksjon?
- \* Rent matematisk:  
“(Black) box”-maskineri,  
med kun input og output
- \* I programmering; ofte  
også “side-effects” -  
endrer på noe (minne,  
filer, piksler etc.)
- \* Javascript har mange  
Innebygde



```
script2.js UNREGISTERED
script2.js
1 function calculate(x){
2     var tall_1=489;
3     var answer=x*tall_1;
4     return x+"*"+tall_1+"="+answer;
5 }
6
7 function showOff(){
8     var tall_1=9256;
9     var answer=calculate(tall_1);
10    alert(answer);
11 }
12
Line 1, Column 1 Tab Size: 4 Java
```



# JS Basics: Variabler

- \* Hva er en variabel?
- \* Data lagret i minnet, med et navn
- \* Finnes i ulike typer; tall, tekststrenger ++
- \* Kan endres - men har et "skop".

A screenshot of a code editor window titled 'script3.js' with a status bar indicating 'UNREGISTERED'. The code is written in JavaScript and includes variable declarations, a function definition, and a function call. The code is as follows:

```
1  var name1="Linus";
2  var name2="Knausgård";
3
4  function getLastName(){
5      var name2="Carlsen";
6      name2=window.prompt("Linus...",name2);
7  }
8
9  function welcomeUser(){
10     getLastName();
11     alert("Welcome "+name1+" "+name2);
12 }
13
```

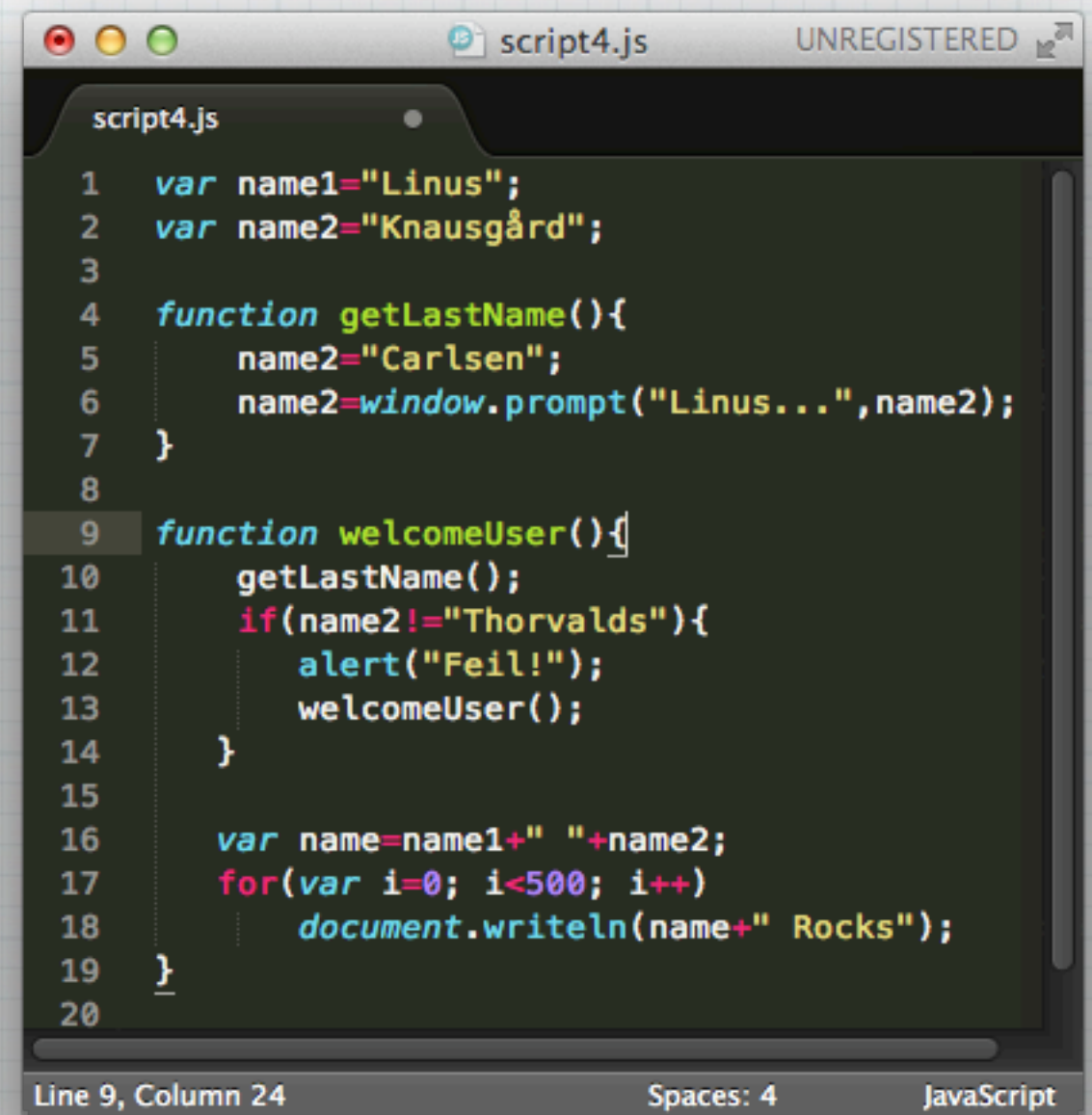
The status bar at the bottom shows 'Line 13, Column 1', 'Tab Size: 4', and 'JavaScript'.



# JS Basics:

## Kontrollstrukturer

- \* Hva er en kontrollstruktur?
- \* Styring av "programflyt"
- \* Funksjoner
- \* Løkker  
for, while, for-in
- \* Kondisjonaler  
if-else-elseif



```
script4.js
UNREGISTERED

1  var name1="Linus";
2  var name2="Knausgård";
3
4  function getLastName(){
5      name2="Carlsen";
6      name2=window.prompt("Linus...",name2);
7  }
8
9  function welcomeUser(){
10     getLastName();
11     if(name2!="Thorvalds"){
12         alert("Feil!");
13         welcomeUser();
14     }
15
16     var name=name1+" "+name2;
17     for(var i=0; i<500; i++)
18         document.writeln(name+" Rocks");
19 }
20
```

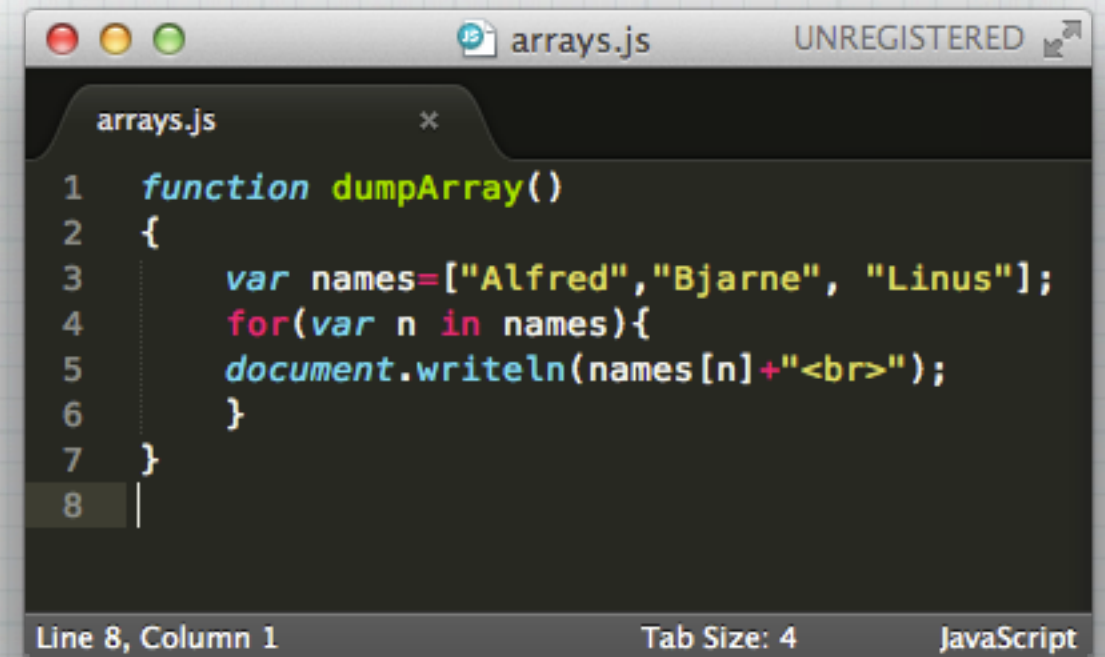
Line 9, Column 24      Spaces: 4      JavaScript



# JS Basics:

## Datastrukturer - Array

- \* Et array er en indeksert liste av objekter (eller typer)
- \* Initialiseres slik:  
`arr=["a", "b", 1, 2];`
- \* Itereres med vanlig løkke eller "for( x in y)"
- \* Aksesseres med "indeks-operator", `arr[2]`



```
1 function dumpArray()
2 {
3     var names=["Alfred","Bjarne", "Linus"];
4     for(var n in names){
5         document.writeln(names[n]+"<br>");
6     }
7 }
8
```

Line 8, Column 1      Tab Size: 4      JavaScript



# Demo

---

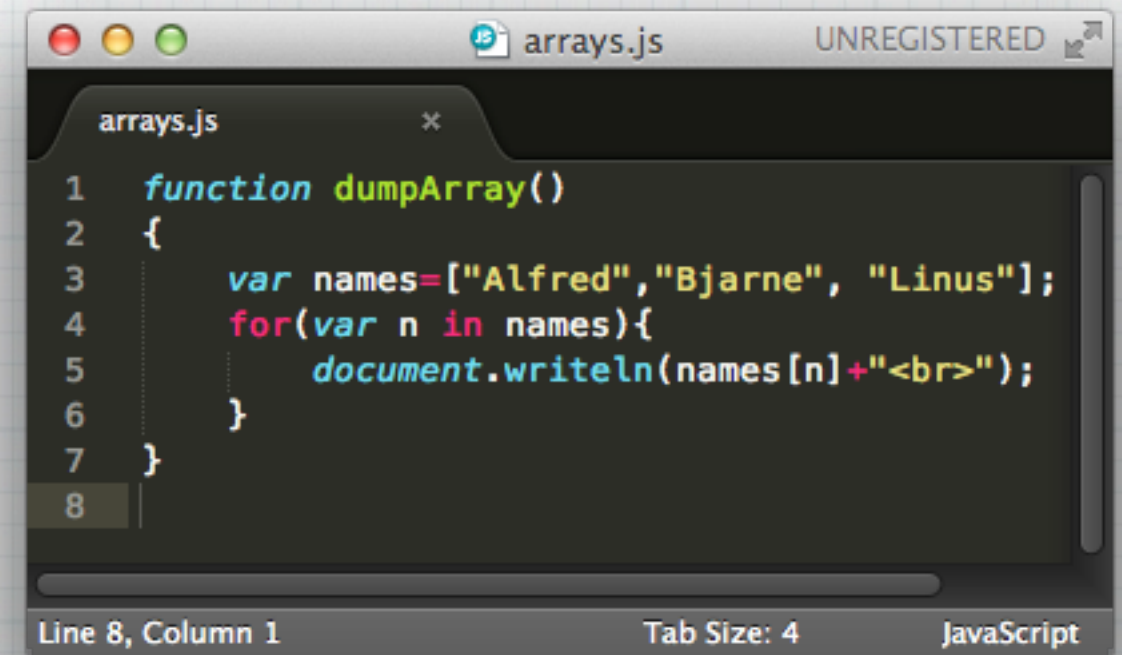
arrays.html



# JS Basics:

## Datastrukturer - objekter

- \* Et "objekt" i javascript er en uordnet liste av navngitte objekter (eller typer)
- \* Initialiseres slik:  
`obj={a: 1, b:[1,2], "c":d};`
- \* Itereres med vanlig løkke eller "for( x in obj)"
- \* Aksesseres med "indeks-operator", `obj["b"]` ELLER `obj.b`
- \* JSON!



```
1  function dumpArray()
2  {
3      var names=["Alfred","Bjarne", "Linus"];
4      for(var n in names){
5          document.writeln(names[n]+"<br>");
6      }
7  }
8
```

Line 8, Column 1      Tab Size: 4      JavaScript



# Demo

---

objects.html



# D.O.M.

# Document Object Model

- \* Hva er "et objekt"?
  - \* En samling egenskaper, med felles navn
  - \* Muligens definert i en felles klasse
  - \* Egenskaper er variabler
- \* Objekter er "sammensatte typer"
  - \* Muligens med private funksjoner (metoder)
  - \* `document.writeln("Hey!");` betyr  
kjør "objektet document" sin "writeln(x)"-funksjon  
og send med strengen "Hey!" som argument



# Demo

---

objects\_width\_functions.html



# D.O.M.

# Document Object Model

- \* Vi vil lage dynamisk GUI av HTML.
  - \* Da må vi ha tilgang til HTML-elementene
  - \* Og egenskapene (farge, form, posisjon, innhold +++)
- \* Ønsker ikke "rendre" alt på nytt
  - \* Ferdige HTML-elementer er tilgjengelig som "objekter" i browser
  - \* DOM gjør disse tilgjengelige for javascript.



# D.O.M.

# Document Object Model

- \* DOM er nå en W3C-standard
- \* En "crossbrowser API" for å manipulere DOM-treet (Rendret HTML)
- \* Definerer objekter som "document", "window", "screen", "HTML-element" osv.
- \* <http://www.w3schools.com/jsref/>



# JQuery

- \* JQuery er et javascript-“rammeverk”, som gjør interaksjon med DOM lettere
- \* Brukes først og fremst til:
  - \* Enkelt finne frem i DOM.
  - \* Bedre styring av HTML-eventer
  - \* Ajax
- \* <http://jquery.com/>



# jQuery: bedre D.O.M-interaksjon

- \* JQuery er et javascript-“rammeverk”, som gjør interaksjon med DOM lettere
- \* `x=document.getElementById("ball")`.  
V.S.  
`x=$("#ball")`
- \* `x=document.getElementsByClassName("person")`  
V.S.  
`x= $(".person")`
- \* `x.toggle(400);`



# Demo

---

jquery1.html



# JQuery: ready();

- \* Problem: Browser laster inn HTML, CSS og javascript over HTTP.
- \* Hvis vi nå tenker at vi er "inne i et javascript". Hvordan vet vi om DOM'en er ferdig?
- \* Vet vi at div'en med id "box" finnes?
- \* Løsning: JQuery ready():
  - \* `$(...).ready( ...legg til funksjon...)`
  - \* `$( ...legg til funksjon ... )`



# Demo

---

`dom_ready.html`



# Demo

---

Sjakk!