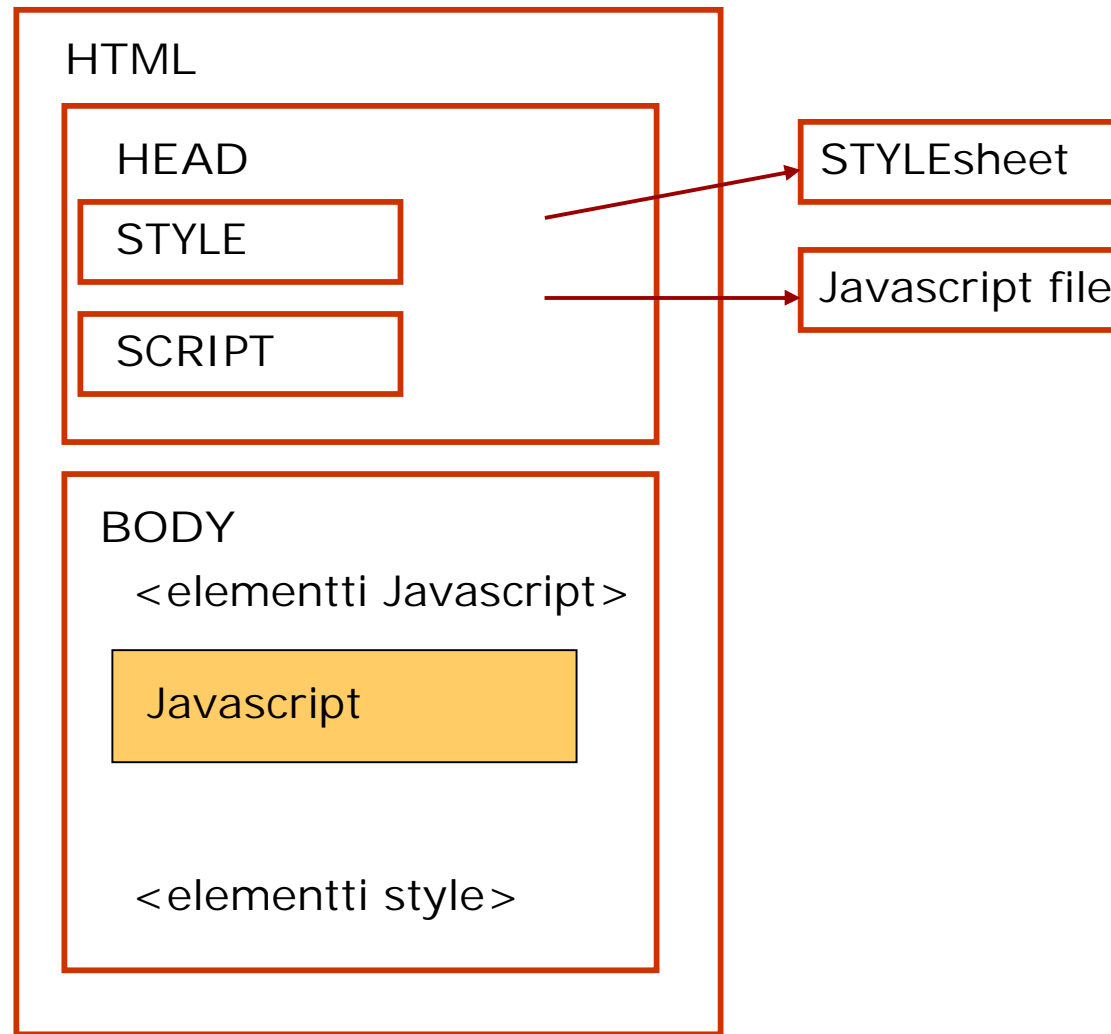


# Javascript 2: Ohjelmointikielen ominaisuudet

Jaana Holvikivi  
Metropolia

# HTML - sivun rakenne ja osiot



# Ohjelmointikielen perusosia

- § Tietotyypit
- § Vakiot
- § Muuttujat
- § Lausekkeet
- § Operaattorit
- § Käskyt ja lauseet: sijoitus, ehtolause, silmukat
- § Tapahtumat
- § Funktiot
- § Metodit

# Tietotyypit Javascriptissä

## Vakiot

§ Luvut / Numbers **0.44** (ei erikseen kokonaislukuja)

§ Merkkijonot / Strings

```
document.write ("Tervehdys"+mj);
```

lainausmerkeissä (joko ' tai ")

```
<input value="Paina" type="button"  
onClick="alert('HEIPPA')">
```

§ Null "tyhjä"

§ String literals

```
alert("I am an alert box!! \n\t Man!");
```

käytetään silloin kun HTML ei käytössä, tuottaa rivinvaihdon

§ Totuusarvot / Boolean values

**true, false**

# Muuttujat

```
var i=0, result = 0;           // = sijoitus, ei yhtäläisyys
  for (i = 0; i <= 10; i++) {
    result += i;
    document.write(i + ": " + result + "<br/>");
  }
```

var i = 0      määritellään muuttuja ja asetetaan sen arvoksi 0

Var a, A;      JavaScript erottelee isot ja pienet kirjaimet "case sensitive"

§ Määritelty muuttuja on lokaali eli paikallinen

§ Ei saa käyttää varattuja sanoja muuttujan nimenä

# Merkkijonot (strings)

§ Esimerkkejä metodeista, joilla voidaan käsitellä merkkijonoja;

```
mj = "kissa"; toinen = "la"
```

|                        |        |
|------------------------|--------|
| <code>mj.length</code> | arvo 5 |
|------------------------|--------|

|                               |       |
|-------------------------------|-------|
| <code>mj.toUpperCase()</code> | KISSA |
|-------------------------------|-------|

|                           |   |
|---------------------------|---|
| <code>mj.charAt(0)</code> | k |
|---------------------------|---|

|                                |      |
|--------------------------------|------|
| <code>mj.substring(0,3)</code> | kiss |
|--------------------------------|------|

yhdistäminen:

|                          |         |
|--------------------------|---------|
| <code>mj + toinen</code> | kissala |
|--------------------------|---------|

## Listat (arrays)

`munlista = [8, 4, 7, 6, 15]`

`munlista.length` saa arvon 5

`munlista[0]` sisältää luvun 8

`uusi = munlista.sort()`

Sisältää listan `[4,6,7,8,15]`

Listat voivat sisältää erilaisia tietotyyppisiä

## <h2> Kuvan vaihtelua </h2> **Esimerkki kuvia sisältävän listan käytöstä**

<p>



<script>

```
var i, frame = 0; var timerID;
```

```
var pics= new Array (10);
```

```
// ensin pitää muodostaa Image-olioiden lista
```

```
for (i=0; i < pics.length ; i++){
```

```
  pics[i] = new Image();
```

```
}
```

```
//sijoitetaan kuvat tähän listaan eli haetaan ne palvelimelta
```

```
for (i=0; i < pics.length ; i++){
```

```
  pics[i].src = "kuvat/kuva" + i + ".gif";
```

```
}
```

```
function vaihtele () {
```

```
  document.images[0].src = pics [frame].src;
```

```
  //images[0] viittaa sivun ensimmäiseen kuvapaikkaan
```

```
  frame = (1 + frame);
```

```
  //aloitetaan timerID uudelleen kutsumalla käsittelijää vaihtele()
```

```
  timerID = setTimeout('vaihtele()', 800);
```

```
}
```

</script>



## Lausekkeet (expressions) ja käskyt

`i <= 10`      vertailulauseke saa arvon tosi tai epätosi

`"tulos on" + summa`  
merkkijonolauseke

Käsky eli statement:

```
timerID = setTimeout('vaihtele()', 800);  
;
```

komennon loppumerkki, parasta käyttää aina

# Operaattorit

## Matemaattiset operaattorit

- + yhteenlasku
  - $x+=y$  vastaa yhteenlaskua  $x=x+y$
  - $x++$  sama kuin  $x=x+1$
- Vähennyslasku
- \* Kertolasku
- / Jakolasku
- % Jakojäännös

## Vertailu, Comparison Operators, tulos true tai false

- == yhtäsuuri kuin
- != erisuuri kuin  $5!=8$  tuloksena "true"
- < pienempi kuin
- > suurempi kuin
- >= suurempi tai yhtä suuri kuin
- <= pienempi tai yhtä suuri kuin

# Operaattorit

Loogiset vertailut

&&    AND

||     OR

!     NOT

|     |   |   | TULOS |
|-----|---|---|-------|
| AND | 0 | 0 | 0     |
|     | 1 | 0 | 0     |
|     | 0 | 1 | 0     |
|     | 1 | 1 | 1     |
| OR  | 0 | 0 | 0     |
|     | 1 | 0 | 1     |
|     | 0 | 1 | 1     |
|     | 1 | 1 | 1     |
| NOT | 1 |   | 0     |
|     | 0 |   | 1     |

## Ehtolause/ Conditional statement

```
§ if ( !Math.random ) // funktion olemassaolon testaus
{
document.write('<p>weather called off </p>');
}
else if ( Math.floor((Math.random()*2)) == 0 )
{
document.write ("<b>It's just awful. </b>");
}
else
{
document.write ("<i>How wonderful it is!</i>");
}
```

# Silmukat / Loops

```
for (i = 0; i <= 10; i++)  
{  
    result += i;  
    document.write(i + ": " + result + "<br/>");  
}
```

Laskurin kasvattaminen (Increment)

`i=i+1` tai `i++`

# Sisäkkäiset silmukat

```
var heads = 0, tails = 0;
var i, j;
  for (j = 0; j <= 5; j++)
  {
    for (i = 0; i < 100; i++)
    {
      if ( Math.floor ((Math.random()*2)) == 1 )
        heads = heads + 1;
      else
        tails = tails + 1;
    }
    document.write ("Heads: " +heads+ </br>
    "Tails : " +tails+ " "<p>");
    heads= 0; tails = 0;
  }
}
```

# Silmukat

```
§ var x = 1;  
   var result = 0;
```

```
   while ( x <= 10 ) // toistetaan kunnes x>10  
   {  
     result += x;  
     x++;  
   }
```

```
   alert ("The result is " + result + " and x is " + x );
```

# Funktiot

## Valmiit funktiot

alert()

prompt()

parseInt()

parseFloat()

Math.sqrt()

Math.floor()

Math.round()

tuottaa kokonaisluvun

tuottaa lukuarvon

neliöjuuri

pyöristys alaspäin

pyöristys



# Funktiot: Käyttäjän määrittelemät

```
function Capitalize(mjono)
//palauttaa merkkijonon, jonka alkukirjain on isolla
{
  var firstletter, reststring, cap;
  firstletter = mjono.charAt(0);
  reststring = mjono.substring(1, mjono.length);
  cap= firstletter.toUpperCase() + reststring.toLowerCase()

  return cap;
}
```

Tämä funktio siis palauttaa mjonon arvoksi cap-arvon.

# Funktiot: käyttäjän määrittelemät

Tulostusta HTML-sivulle: Tämä funktio kirjoittaa numeroita tekstialueeseen textarea.

```
<head>
<script>
function countdown()
{
    var count;
    count = document.getElementById("countBox").value;
    document.getElementById("tulostusta").value = "";
    while (count > 1){
        document.getElementById("tulostusta").value =
        document.getElementById("tulostusta").value + count + "\n";
        count = count -1;
    }
    document.getElementById("tulostusta").value =
        document.getElementById("tulostusta").value + "hep!";}
</script>
</head>
<body>
    <p>Mistä luvusta lähtölaskenta aloitetaan?
    <input type = "text" id = "countBox" size = "3" value = "19"/>
    </p>
<p><input type = "button" value = "Aloita lähtölaskenta" onClick = "countdown()"/>
    </p>
<p>
    <textarea id = "tulostusta" rows = "20" cols = "8">
    </textarea>
</p>
</body>
```

# Metodit

- § `close()` Sulkee tulostuksen, joka on aloitettu `document.open()` metodilla, näyttää datan
- § `getElementById()` Palauttaa viittauksen ens. Objektiin, jolla on annettu id
- § `getElementsByName()` Palauttaa joukon nimen mukaisia objekteja
- § `getElementsByTagName()` Palauttaa joukon elementin mukaisia objekteja
- § `open()` Avaa tulostusputken `document.write()` or `document.writeln()` metodeille
- § `document.write()` Kirjoittaa HTML tai JavaScript koodia dokumenttiin

# Tapahtumankäsittelijät

|                              |                                      |
|------------------------------|--------------------------------------|
| onabort                      | käyttäjä keskeyttää sivun lataamisen |
| onClick                      | käyttäjä painaa oliota               |
| onfocus                      | käyttäjä aktivoi olion               |
| onblur                       | osoitin poistuu oliosta              |
| onerror                      |                                      |
| onSelect                     |                                      |
| onChange                     |                                      |
| onmouseover                  |                                      |
| onkeypress                   |                                      |
| onload                       | olion lataaminen on suoritettu       |
| onsubmit                     | käyttäjä lähettää lomakkeen          |
| onunload                     |                                      |
| setTimeout(), clearTimeout() | ajotus tulevaan hetkeen              |

# Event onload()

```
<html>
  <head>
    <title>AJAX Foundations: javascript and DOM</title>
    <script type="text/javascript" src="morejsdom.js">
      <!-- tiedosto sisältää funktion process -->
    </script>
  </head>

  <body onload="process()">
    Hello! Here is a cool list of colors for you:
    <br/>
    <div id= " hook" />
  </body>
</html>
```

## onload() - selostusta

morejsdom.js

```
function process() // eräs tapa tuottaa tulostusta HTML-sivulle
{
    var string;
    string="<ul>"
        + "<li>Black</li>"
        + "<li>Red</li>"
        + "<li>Blue</li>"
        + "</ul>";
    var myDiv=document.getElementById("hook");
    myDiv.innerHTML=string;
}
```

# innerHTML - ominaisuus

ESIMERKIKSI:

```
<div id="hook">stories and <h2>headings</h2></div>
```

hook:in **innerHTML** -property  
stories and <h2>headings</h2>

hook:in **outerHTML** -property

```
<div id="hook">stories and <h2>headings</h2></div>
```

## Esimerkki sivun sisällön muuttamisesta

```
<html>
<head>
<title>Minun tulosteet</title>
<script>

function process() // eräs tapa tuottaa tulostusta HTML-sivulle
{
  var string;
  string="<ul>"
    + "<li>Black</li>"
    + "<li>Red</li>"
    + "<li>Blue</li>" + "</ul>";
  var myDiv=document.getElementById("hook");
  myDiv.innerHTML=string;
}
</script>
</head>

<body>
<h1> Sivulle tulostusta </h1 >
<h2> </h2>
<p>
<div id="hook">stories and <h2>headings</h2></div>
<input type="button" value="Tulosta lista" onClick="process(); "/>
</body>
</html>
```



# Javascript ohjelmointikielenä

## § Olio-orientoitunut:

- § Kaikesta ei tarvitse kirjoittaa omaa ohjelmaa, vaan voi käyttää hyväksi/ luoda luokkakirjastoja, joihin sisältyy olioiden käyttäytymistä
- § DOM sisältää tyyppien hierarkian
- § Joustava funktioiden käyttö
- § OO koodi, virheiden käsittely, erilaisia metodeja ja tyyppihierarkiat
- § JSON object notation, ynnä muuta valmista tukea
- § Google Webtoolkit
- § <http://jquery.com/> valmis kirjasto, jossa paljon käyttöliittymäominaisuuksia