

Digitaalisen median tekniikat JavaScript

19.9.2007 Harri Laine 1

JavaScript

- ent. **LiveScript** (Netscape), muunnelma **JScript** (Microsoft)
 - yhteensopivat yksinkertaisissa asioissa, aiemmin yhteensopimattomat hiemankin edistyneemmissä
- nyk. **ECMAScript** (standardi)
- selaimessa toimiva tapahtumaperustainen skriptikieli
- ei mitään tekemistä Javan kanssa paitsi, että molempien lähtökohtana on C-kielen syntaksi

19.9.2007 Harri Laine 2

JavaScript

- Tulkettava ohjelmointikieli
- Toimii selainympäristössä (client side) – pääsy vain dokumentin dataan
- Ohjelmat ovat enimmäkseen erilaisiin tapahtumiin (käyttäjätoimintoihin tai dokumentin käsittelyvaiheisiin) liittyviä käsittelijöitä, joilla toteutetaan
 - tarkistuksia
 - esittämisen ohjausta
 - sivun ulkoasun ja näkyvän sisällön muokkausta
 - dokumentin näkyvään muotoon voidaan tehdä muutoksia sen jälkeen kun dokumentti on ladattu selaimen – dokumentti itsessään ei kuitenkaan muutu (js_esim1.html)
 - dokumentin täydennystä noutamalla lisämateriaalia
 - kommunikointia palvelimen kanssa

19.9.2007 Harri Laine 3

JavaScript

- JavaScriptillä ei voi
 - käsitellä paikallisia tiedostoja
 - käsitellä suoraan palvelimen tiedostoja tai tietokantoja
 - suorittaa selaimen ulkopuolisia ohjelmia
 - käyttää oheislaitteita suoraan (esim. kirjoitinta)

19.9.2007 Harri Laine 4

JavaScript syntaksi

- perusrakenteeltaan C:n ja Javan kaltainen
- case sensitive
- sijoitusoperaatio (=) esim. **a=b;**
- lauseet erotetaan toisistaan puolipisteellä, mutta **rivin loppu toimii myös erottimena**

```
function sample() {  
  var a = 1;  
  var b = 2;  
  var c = 3;  
  return  
  a + b + c;  
}
```

19.9.2007 Harri Laine 5

JavaScript syntaksi

- perusrakenteeltaan C:n ja Javan kaltainen
- case sensitive
- sijoitusoperaatio (=) esim. **a=b;**
- lauseet erotetaan toisistaan puolipisteellä, mutta **rivin loppu toimii myös erottimena**

```
function sample() {  
  var a = 1;  
  var b = 2;  
  var c = 3;  
  return  
  a + b + c;  
}
```

=

```
function sample() {  
  var a = 1;  
  var b = 2;  
  var c = 3;  
  return  
  a + b + c;  
}
```

Palauttaa undefined riippumatta muuttujien arvoista

19.9.2007 Harri Laine 6

JavaScript syntaksi

- Rivin loppumisen toimiminen erottimena vaikuttaa myös siihen miten lausekkeet on kirjoitettava:

```
teksti= 'ensimmäinen rivi '  
+ 'toinen rivi'  
+ 'kolmas rivi';
```

 ei toimi

```
teksti= 'ensimmäinen rivi ' +  
'toinen rivi' +  
'kolmas rivi';
```

 toimii

19.9.2007

Harri Laine

7

JavaScript syntaksi

- Kommentit:
`/*monirivinen*/`
`// yksirivinen`
- `<!--` tunnustetaan yksirivisen kommentin alkumerkiksi, mutta `-->` ei tunnistu loppumerkiksi, siksi koodin piilotus vanhoilta selaimilta:

```
<script type="text/javascript">  
<!--  
    JavaScriptiä tähän väliin  
// -->  
</script>
```

19.9.2007

Harri Laine

8

JavaScript syntaksi

- Merkkijono ei voi jakautua monelle riville

```
teksti= 'ensimmäinen rivi  
toinen rivi  
kolmas rivi';
```

 väärin

19.9.2007

Harri Laine

9

JavaScript syntaksi

- Vakiot:
 - Numeeriset vakiot: `123`
 - Merkkijonovakiot jonkinlaisten lainausmerkkien sisässä kunhan kummallakin puolella samanlainen
 - `"merkkijono"`, `'merkkijono'`, ``merkkijono``

19.9.2007

Harri Laine

10

JavaScript tietotyypit

- luvut
 - kokonaisluvut,
 - liukuluvut `3.14`, `1.2e3 = 1.2*103 = 1200`,
 - erikoisarvo: `NaN = not a number`
- merkkijonot
 - erikoismerkit kuten Javassa `\b (backspace)`, `\t (tab)`, `\n (rivinvaihto)`, `\'`, `\``, `\\`,
 - `\x99 (ascii merkki hexana)`, `\u9999 (unicode)`
- totuusarvot:
 - `true/false (laskennassa 1/0)`

19.9.2007

Harri Laine

11

JavaScript tietotyypit

- oliot (object)
 - JavaScript ei ole olioperustainen kieli
 - kielessä on olion käsite, mutta ei esimerkiksi periytymistä
 - Olio on kokoelma nimettyjä ominaisuuksia
 - ominaisuudella on nimi ja arvo
 - ominaisuuden arvoon viitataan joko pistenotaatiolla
 - `olio.ominaisuus` (esim. `image.src`)
 - tai assosiativisen taulukon tapaan
 - `olio['ominaisuus']` (esim. `image['src']`)
 - mahdollistaa ominaisuuden nimen antamisen muuttujana
 - Olion ominaisuuden arvona voi olla olio. Sen ominaisuuden arvoon viitataan
 - `olio.ominaisuus.ominaisuuden_ominaisuus`
 - esim `document.form1.action`

19.9.2007

Harri Laine

12

JavaScript tietotyypit

- Olioilla voi olla metodeja esim `document.write('text')`
- Käytettäessä JavaScriptiä HTML dokumenttien käsittelyyn näkyy dokumentti ohjelmalle olioista muodostuva dokumenttipuuna (DOM, Document Object Model) – tästä myöhemmin enemmän
- metoditkin olioiden ominaisuuksia
- olioiden sijoitus toimii kuten Javassa, eli sijoituksen kohde saa viitteen lähteen viittaamaan olioon

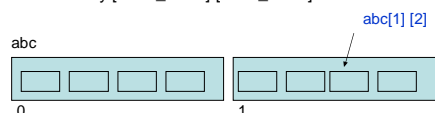
19.9.2007

Harri Laine

13

JavaScript tietotyypit

- Taulukot (array)
 - taulukko on kokoelma alkioita, joihin voi viitata järjestysnumeron avulla
 - `array [index]`, indeksointi alkaa nolasta
 - taulukon alkiona voi olla taulukko
 - `array [outer_index] [inner_index]`



19.9.2007

Harri Laine

14

JavaScript muuttujat

- Muuttujanimet:
 - alkavat kirjaimella tai `_` tai `$` -merkillä
 - muut merkit ascii-merkkejä, ei operaatiosymboleja
 - varatut sanat eivät käy muuttujina
- Muuttujat ovat **tyypittömiä**, toisin kuin esim. Javassa
 - muuttujan tyyppiä ei määritellä esittelyssä
 - tyyppi määräytyy käytön mukaan, jos muuttujan sijoitetaan luku muuttuja tulee tyybiltään lukuarvoiseksi
 - tyyppi voi muuttua

19.9.2007

Harri Laine

15

JavaScript muuttujat

- automaattiset tyyppimuunnokset

```
// arvoksi merkkijono '10'
var item1 = '10';
// arvoksi merkkijono '20'
var item2 = '20';
// kertolasku on numeerinen operaatio ja muuttaa tyyppiin 10
item1 = item1 * 1;
// item2 arvoksi tulee merkkijono '2010'
// sillä item1 muunnetaan merkkijonoksi katenointia varten
// plus-operaattori on monimerkityksinen mutta katenointi voittaa summan
item2 = item2 + item1;
// item2 arvoksi tulee luku 201
// sillä jakolasku on puhtaasti numeerinen
item2 = item2 / item1;
```

19.9.2007

Harri Laine

16

JavaScript muuttujat

- Eksplisiittiset tyyppimuunnokset
- `parseInt(string)`, `parseFloat(string)`
 - ottavat merkkijonon alusta maksimipituisen kyseiseksi lukutyypiksi tulkittavissa olevan luvun

```
var string1 = '3.1417 is the value of Pi';
var string2 = 'The value of Pi is 3.1417';
Int_1 = parseInt(string1); // value of Int_1 is 3
Int_2 = parseInt(string2); // value of Int_2 is NaN
Int_3 = parseFloat(string1); // value of Int_3 is 3.1417
Int_4 = parseFloat(string2); // value of Int_4 is NaN
```

Not a number

19.9.2007

Harri Laine

17

JavaScript muuttujat

- Muuttuja esitellään `var` avainsanan jälkeen
- Samassa `var`-lauseessa voi esitellä monta muuttujaa
- Muuttujalle voidaan antaa esittelyn yhteydessä alkuarvo, ellei alkuarvoa ole annettu on muuttujan arvona **undefined**
- On sallittua esitellä muuttuja toistuvasti (**potentiaalinen virhelähde**)
 - jos myöhemmässä esittelyssä asetetaan alkuarvo se korvaa aiemman arvon
 - jos myöhemmässä esittelyssä ei anneta alkuarvoa säilyy aiempi arvo (huh!)

19.9.2007

Harri Laine

18

JavaScript muuttujat

```
// simple declaration
var item1;
// declaration of multiple variables
var item1, item2, item3;
// declaration with assignment
var item1 = 7;
// multiple variables with assignment
var item1 = 7, item2 = 'cat', item3 = 3.17;
```

19.9.2007

Harri Laine

19

JavaScript muuttujat

- paikalliset muuttujat
 - funktioiden sisällä määritellyt muuttujat ovat paikallisia eli voimassa vain funktion sisällä
 - funktioiden ulkopuolella määritellyt muuttujat ovat globaaleja
 - globaaleja muuttujia voi käyttää funktioissa ellei paikallisesti ole määritetty samannimistä muuttujaa, jolloin käytetään sitä

19.9.2007

Harri Laine

20

JavaScript muuttujat

```
var item1 = 'global';

function testTheScope() {
  item1 = 'local';
  document.write(item1);
}
testTheScope();
document.write(item1);
```

- tulostaa:

local local

19.9.2007

Harri Laine

21

JavaScript muuttujat

```
var item1 = 'global';

function testTheScope() {
  var item1 = 'local';
  document.write(item1);
}
testTheScope();
document.write(item1);
```

- tulostaa:

local global

19.9.2007

Harri Laine

22

JavaScript operaattorit

- JavaScriptin matemaattiset, vertailu- ja loogiset operaattorit ovat pääasiassa samat kuin Javassa
- merkittävimpiä eroja:
 - merkijonojen arvoja verrataan **samoilla operaattoreilla** kuin lukuja (==, !=, >, ...)
 - tavallisten vertailujen yhteydessä tehdään tyyppimuunnos, jos verrattavat ovat eri tyyppiä
 - 1=="1", 1===true
 - '11'<'3' mutta '11'>3
 - tiukkojen operaatioiden === ja !== yhteydessä ei tyyppimuunnosta ts 1!="1"
 - null===undefined, null===null

19.9.2007

Harri Laine

23

JavaScript operaattorit

- binäärinen + on kuormitettu
 - yhteenlasku, katenaatio
 - jos molemmat osapuolet lukuja niin yhteenlasku muuten katenaatio
- typeof(muuttuja) antaa muuttujan tyyppin

19.9.2007

Harri Laine

24

JavaScript lauseet

- ehtolauseet

```
if (ehto) { lauselohko }

if (ehto) { lauselohko }
else { lauselohko }

if (ehto1) { lauselohko }
else if (ehto2) { lauselohko }
...

switch (lauseke) {
  case arvo: lauselohko
  case ...
  default: lauselohko
}
```

break kuten Javassa

19.9.2007

Harri Laine

25

JavaScript lauseet

```
switch (parseInt(xyz)) {
  // useita case-vaihtoehtoja voidaan yhdistää
  case NaN:
  case 0:
  case 10:
    window.alert(xyz + ' is not a value I can work with!');
    break;
  default:
    window.alert(xyz + ' is ready for processing!');
    abc = someFunc(xyz);
    break;
}
```

19.9.2007

Harri Laine

26

JavaScript lauseet

- Toistolauseet **while**, **do** ja **for** kuten Javassa

- Lisäksi:

- for (muuttuja in objekti) {lauselohko}

- käy läpi taulukon alkioita tai olion ominaisuudet

```
for (elName in navigator) {
  document.write(elName);
  document.write(" = ");
  document.write(navigator[elName]);
  document.write("<br />");
}
```

19.9.2007

Harri Laine

27

JavaScript funktiot

- JavaScriptissä voidaan määrittellä funktioita.

- Toisin kuin Javassa:

- voidaan määrittellä myös irrallisia funktioita, jotka eivät ole minkään olion metodeja
- funktioiden paluuarvon tyyppiä ei voi määrittellä
- funktion esittelyssä ja kutsussa voi olla eri määrä argumentteja (puuttavilla arvo *undefined*, ylimääräisiin pääsee käsiksi taulukon *arguments* kautta)
- return:n perässä voidaan antaa paluuarvo, ellei anneta palautetaan *undefined*
- taulukot ja oliot viiteparametreja, muut argumentit arvoparametreja

```
function functionName(arguments) {
  statements;
  return;
}
```

19.9.2007

Harri Laine

28

JavaScript taulukot

- Taulukon luonti

- var taulu= new Array();
 - alkioiden määrää ei ole annettu
 - taulukot ovat dynaamisia, joten alkioita voidaan sijoittaa ylärajan ulkopuolellekin
 - taulu[0]=1;
 - taulukkoa voi käyttää myös assosiativisena (vrt hashtable)
 - taulu['uusialkio']=2; (jos taulukossa oli aiemmin yksi alkio (jonka indeksi siis 0), niin uusialkio assosioidaan arvoon 1.
- var taulu1= new Array(6); // 6 alkioita, undefined
- var taulu2= new Array(1,20,'abc',200);
 - tauluko jossa 4 alkioita
- var taulu3 = [1,20,'abc',200];
- ominaisuus length ilmoittaa taulukon koon, taulu3.length==4

19.9.2007

Harri Laine

29

JavaScript taulukot

- Taulukon koko määräytyy sen todellisen koon mukaan ei esittelyn

```
var pikkutaulu= Array(5);
pikkutaulu[200]=1;
pikkutaulu.length==201
```

- Taulukkometodeja, esim:

```
taulukko.concat(taulukko1) liittää taulukon loppuun toisen taulukon alkioita
taulukko.sort() järjestää alkioita
```

19.9.2007

Harri Laine

30

Dokumentin käsittely

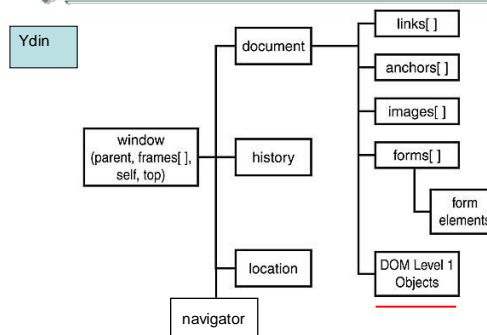
- JavaScript-ohjelma näkee dokumentin rakenteen ns. dokumenttipuuna, dokumenttipuun saattaa eri selaintoteutuksissa olla hieman erilainen, joten usein ennen puun hyödyntämistä on tutkittava mikä selain on kyseessä.

19.9.2007

Harri Laine

31

Selainluokat ja oliot



19.9.2007

Harri Laine

32

Selainluokat ja oliot

- window**
 - selainikkunaan liittyviä palveluja
 - window globaali olio, muu rakenne sen alapuolella
 - open (uusi ikkuna), close
 - pop-upit: alert, prompt, confirm (katso esimerkki)
 - kymmeniä metodeja
- navigator**
 - tietoja selaimesta (katso esimerkki)
 - selaimen tunnistus voidaan tehdä navigator:in tietojen perusteella – usein kuitenkin kysytään jotain yleistä ominaisuutta, jonka tiedetään olevan vain jollain selaimella, esimerkiksi: if (document.all) tunnistaa uudehkon IE:n

19.9.2007

Harri Laine

33

Selainluokat ja oliot

- history**
 - tietoa selailureitistä
 - voidaan esimerkiksi toteuttaa selaimen back- ja forward- näppäimet sivulle sijoitettuina kontrolleina
- location**
 - tietoa ladatusta dokumentista mm href-ominaisuus= nykyisen dokumentin URL, vaihtamalla tämän arvo vaihtuu ikkunan sisältö
 - location.replace(url) vaihtaa ikkunan sisällön ja korvaa historiassa edellisen url:n uudella. Edelliseen ei siten pääse takaisin back-näppäimellä

19.9.2007

Harri Laine

34

Selainluokat ja oliot

- document**
 - varsinaisen dokumentin pääolio
 - eri tyyppiset elementit erillisissä taulukoissa
 - anchors, applets, forms, images, links,...
 - rakenteita joilla voidaan käydä läpi eri tyyppisiä elementtejä
 - childNodes= solmun lapsielementit,
 - all (IE only)
 - dokumentin ominaisuuksia
 - referrer, cookie, title, attributes-taulukko, ...

19.9.2007

Harri Laine

35

Selainluokat ja oliot

- Dokumentin elementteihin voi viitata niiden name-attribuutin arvon avulla, esimerkiksi lomakkeen lomake1 kenttään k1 voi viitata
 - document.lomake1.k1
- ja sen arvoon
 - document.lomake1.k1.value

19.9.2007

Harri Laine

36

Selainluokat ja oliot

- document metodeja, mm:
 - open(), close(), write(string), writeln(string)
 - navigointimetodit
 - getElementById(string) –id:n perusteella
 - getElementsByTagName(string) – tagin perusteella
 - getElementsByName(string) – name attribuutin perusteella
 - aineiston luonti
 - createElement(tag)
 - createTextElement(string)
 - arvomuutokset
 - setAttribute(name, value)

19.9.2007

Harri Laine

37

Selainluokat ja oliot

- element yli luokka dokumentin elementeille
 - kaikille yhteisiä rakenteita ja metodeja, tärkeimpiä:
 - attributes - attribuuttitaulukko
 - childNodes - lapsielementit
 - id - tunniste
 - class - elementin luokitus (class-määre)
 - innerHTML - sisältö tageineen
 - style - tyylimääritys
 - JavaScriptissä tyyliattribuuteille on eri nimet kuin CSS:ssä tähän tapaan: borderTop =CSS:border-top

19.9.2007

Harri Laine

38

Selainluokat ja oliot

- element metodeja
 - getAttribute(name), setAttribute(name,value), hasAttribute(name)
 - getElementsByTagName(tag)

19.9.2007

Harri Laine

39

JavaScript tapahtumakäsittelijät

- [onblur](#) Returns the event handling code for the blur event.
- [onclick](#) Returns the event handling code for the click event.
- [ondblclick](#) Returns the event handling code for the dblclick event.
- [onfocus](#) Returns the event handling code for the focus event.
- [onkeydown](#) Returns the event handling code for the keydown event.
- [onkeypress](#) Returns the event handling code for the keypress event.
- [onkeyup](#) Returns the event handling code for the keyup event.
- [onmousedown](#) Returns the event handling code for the mousedown event.
- [onmousemove](#) Returns the event handling code for the mousemove event.
- [onmouseout](#) Returns the event handling code for the mouseout event.
- [onmouseover](#) Returns the event handling code for the mouseover event.
- [onmouseup](#) Returns the event handling code for the mouseup event.
- [onresize](#) Returns the event handling code for the resize event.
- [onload](#), [onunload](#) usein body elementin yhteydessä

19.9.2007

Harri Laine

40

JavaScript html-dokumentissa

- JavaScriptiä voidaan upottaa HTML-koodiin sekaan:
 - koodia sisältävinä <script> - elementteinä
 - ulkoisia tiedostoja kytkevinä <script> - elementteinä
 - linkin kohteina
 - tapahtumakäsittelijöinä.

19.9.2007

Harri Laine

41

JavaScript html-dokumentissa

- Tyypillisesti dokumentin head-osaan sisältyy:
 - yksi <script>-elementti, jossa määritellään paikalliset funktiot ja globaalit muuttajat
 - mahdollisesti useita <script>-elementtejä, joissa ladataan ulkoisia JavaScript-tiedostoja (nämä sisältävät tyypillisesti useilla sivuilla tarvittavia funktioita)
- Body-osaan voidaan laittaa:
 - <script>-elementtejä sellaisiin kohtiin, joihin halutaan tuottaa sisältöä JavaScriptillä
 - Linkin kohteeksi voidaan määrittellä suoritettava JavaScript-koodi
`OtherFile`
 - Elementin tapahtumakäsittelijäksi voidaan antaa Javascript-koodia
``
(kehysta() on aiemmin määritelty funktio)

19.9.2007

Harri Laine

42

JavaScript html-dokumentissa

- JavaScript koodi suoritetaan siinä vaiheessa kun se tulee vastaan dokumentin käsittelyssä:
- <script>-elementti, jossa koodia

```
<script type="text/javascript" language="JavaScript">
<!-- hide script from older browsers
var hiThere = 'I am a JavaScript statment'; window.alert(hiThere);
// stop hiding script -->
</script>
```
- XHTML:ssä <script> -elementtiin voi liittää `defer="defer"` attribuutin ilmaisemaan, että koodi suoritetaan vasta kun koko dokumentti on ladattu

19.9.2007

Harri Laine

43

JavaScript html-dokumentissa

- ulkoisen tiedoston lataus:

```
<script type="text/javascript" language="JavaScript"
src="myscript.js">
</script>
```
- Esimerkkejä:
- Tämän kurssin kurssisivujen valikko
 - <http://jdstiles.com/javamain.html>
 - <http://developer.irt.org/script/script.htm>

19.9.2007

Harri Laine

44