



Digitaalisen median tekniikat

JavaScript_osa2

JavaScript oliot

- JavaScriptissä voidaan määritellä myös luokkia ja olioita.
- Luokka määritellään konstruktorifunktion avulla

```
function Ympyra(x,y,r) {  
  this.xkoord=x;  
  this.ykoord=y;  
  this.sade=r;  
  this.keha= 2*Math.Pi*this.sade;  
  this.pinta_ala= Math.Pi*(this.sade * this.sade);  
}
```

eivät muutu, jos sädettä myöhemmin muutetaan

- ja tälle instanssi

```
var y1 = new Ympyra(10,10,3);
```

JavaScript oliot

- instanssin muuttujiin viitataan piste-notaatiolla tai indeksillä:
 - `y1.sade`
 - `y1['sade']`
 - Olio voidaan ajatella assosiatiivisena taulukkona

y1	
'xkoord'	10
'ykoord'	10
'sade'	3
'keha'	
'pinta_ala'	

JavaScript oliot

- Kuten taulukkoonkin olioon voidaan lisätä uusia ominaisuuksia, jos indeksi ei ole käytössä luodaan uusi alkio:
`y1.vari='punainen';`
- aiheuttaa uuden ominaisuuden lisäyksen

JavaScript oliot

- Metodin lisääminen:
- Muutetaan ympyrän määrittelyä:

```
function Ympyra(x,y,r) {  
    function laske_keha { return (2*Math.Pi*this.sade); }  
    function laske_pinta_ala {  
        return (Math.Pi*(this.sade * this.sade));  
    }  
    this.xkoord=x;  
    this.ykoord=y;  
    this.sade=r;  
    this.keha= laske_keha;  
    this.pinta_ala= laske_pinta_ala;  
}
```

FUNKTIOARVOISIA
MUUTTUJIA




JavaScript oliot

- Edellä oleva voidaan antaa myös:

```
function Ympyra(x,y,r) {  
  this.xkoord=x;  
  this.ykoord=y;  
  this.sade=r;  
  this.keha= laske_keha;  
  this.pinta_ala= laske_pinta_ala;  
}  
function laske_keha { return (2*Math.Pi*this.sade);  
}  
function laske_pinta_ala {  
  return (Math.Pi*(this.sade * this.sade));  
}
```

FUNKTIOARVOISIA
MUUTTUJIA



JavaScript oliot

- Metodin kutsu: `y1.keha()`
- Metodilla voi olla myös parametreja
- Instanssimetodit määritellään konstruktorissa sijoittamalla arvo instanssimuuttujalle.
- Tällainen metodi on jokaisella instanssilla
- Metodeja voidaan lisätä myöhemmin, mutta silloin ne ovat instanssikohtaisia
- `y1.isonna= kasvata(z);`

```
function kasvata(z) {  
  this.sade +=z;  
}
```

JavaScript oliot

- Luokkametodit ovat myös mahdollisia:

```
function laske_halkaisija (sade) {  
    return (sade*2);  
}
```

```
Ympyra.halkaisija = laske_halkaisija;
```


JavaScript valmiit luokat

- **Date** - päiväys
 - mittaa unix-aikaa eli millisekunteja 1.1.1970 alkaen
 - `var nykyhetki = new Date();`
 - osien asetus- ja kyselymetodeja (katso esimerkki)

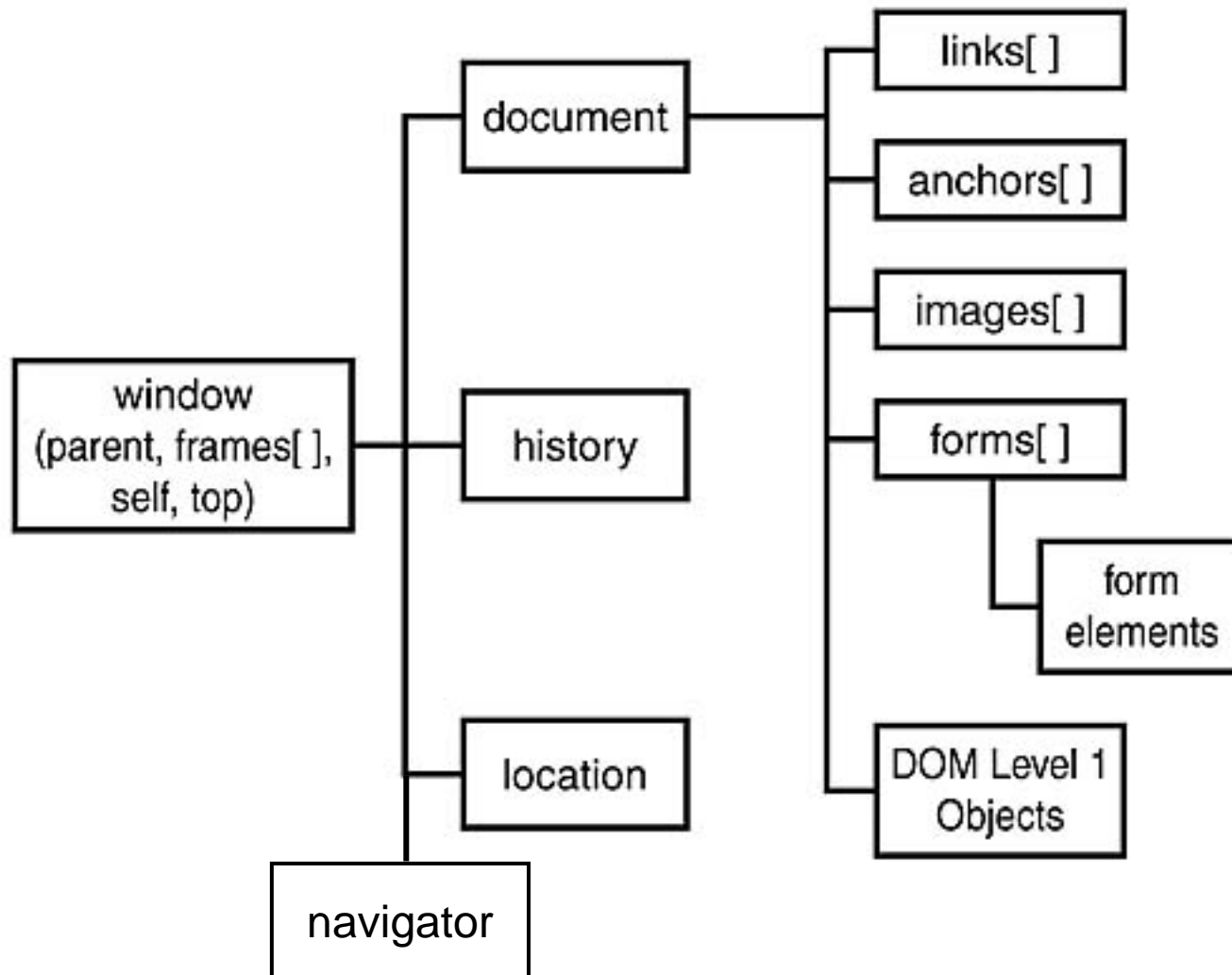
getDate()	the day on the month - numeric value
getDay()	the day of the week 1 =Monday .. 7
getMonth()	the month as an integer 0 = January .. 11
getFullYear()	the year as a four digit number
getMinutes()	minutes (0-59)
getHours()	integer in 24 hour system. (0-23)

JavaScript valmiit luokat

- Math: matemaattisia funktioita luokkametodeina
- String:
 - merkkijono on luokkakin vaikka sijoitukset ja vertailut toimivat kuten perustietotyyppillä
 - ominaisuuksia ja metodeja esim. (katso esimerkki)
 - length
 - substring(alku,loppu) substring(0,1) =ensimmäinen merkki =charAt(0)
 - substr(alku,pituus)
 - indexOf(merkkijono, alku)
 - toLowerCase(),toUpperCase(),
 - RegExp,Object,.....

Selainluokat ja oliot

Ydin



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

Selainluokat ja oliot

- **history**
 - tietoa selailureitistä
 - voidaan esimerkiksi toteuttaa selaimen back- ja forward- näppäimet sivulle
- **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

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-tila, ...

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
 - `tabs=document.getElementsByTagName("table");`
 - alkioihin viitataan `tabs[i]` tai `tabs.item(i)`
 - aineiston luonti
 - createElement(tag)
 - createTextElement(string)
- arvomuutokset
 - setAttribute(name,value)

Selainluokat ja oliot

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

Selainluokat ja oliot

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

<http://www.mozilla.org/docs/dom/domref/>

http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/dhtml_reference_entry.asp

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](#) use in body elementin yhteydessä

JavaScript html-dokumentissa

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

JavaScript html-dokumentissa

- Tyypillisesti dokumentin head-osaan sisältyy:
 - yksi `<script>`-elementti, jossa määritellään paikalliset funktiot ja globaalit muuttujat
 - 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 määritellä suoritettava JavaScript-koodi
`OtherFile`
 - Elementin tapahtumakäsittelijäksi voidaan antaa Javascript-koodia
 - ``
(kehysta() on aiemmin määritelty funktio)

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

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>

Dynaaminen valikko

- Kurssisivun valikko
 - katso toiminta tarkemmin sivulta
<http://www.cs.helsinki.fi/group/vertti/index.html>
 - idea:
 - Sivusto muodostuu joukosta samarakenteisia sivuja
 - Kullakin sivulla on kaksisarakkeinen taulukko, jonka vasemmassa sarakkeessa on sivuvalikko.
 - Sivun ollessa valittuna on sitä vastaava valikkoalkio korostettu
 - Valikko voi olla 3-tasoinen
 - alempi taso saadaan näkyviin ja piiloon valikkoalkiota näpäyttämällä

Dynaaminen valikko

- idea:
 - valikko on toteutettu monitasoisena listana
 - kullekin valikkotasolle on kaksi tasokohtaista luokkaa (class) mnuX (X=0..2) ja mactX
 - mactX luokitus annetaan aktiiviselle valikkoalkiolle
 - vain yksi valikkoalkio voi olla samanaikaisesti aktiivinen
 - luokitus tehdään JavaScriptillä, sivun tekijän ei tarvitse tietää siitä mitään
 - sivun tekijä tekee valikon tavallisena monitasoisena linkkilistana, jokaiselle lista-alkiolle on kuitenkin määriteltävä id:
 - alemman tason alkion id ylemmän tason id:n alukseen
 - esim valinta1 → valinta11 → valinta111, valinta112,...
 - ylimmällä tasolla ei mikään id saa olla toisen osa
 - id-standardinimeämistä käytetään siihen, että yhdellä läpikäynnillä voidaan avata tarvittavat valikot