

Ensimmäinen tunti - Hello World -ohjelma

Tällä tunnilla tehdään **K:\Kotisivu\JavaScript** -kansio, jonne tallennetaan kaikki tällä tunnilla tulevat tehtävät. Lisäksi kopioidaan J:\9ATK\JavaScript -kansioista kaksi html-tiedostoa (index ja vvaliko) **K:\Kotisivu\JavaScript** -kansioon. Avaa index.htm -tiedosto ja tutkaile sen sisältöä.

Ensimmäisenä varsinaisena JavaScript-harjoituksena tehdään ns. **Hello World** -ohjelma, joka tulostaa näytölle (*tai valitulle tulostuslaitteelle*) **Hello World** -tekstin.

Tehtävä 1. Tee seuraava html-sivu ja tallenna se **Hello World.htm** -tiedostonimellä **K:\Kotisivu\JavaScript** -kansioosi.

Muista tallentaa tiedosto ensimmäisen kerran heti kirjoitettuaasi tiedoston *ensimmäisen rivin*.

```
<html>
<head>

<script type="text/javascript" charset="iso-8859-1">
</script>

<title>Juha Korhonen - Ohjelmointikurssi 2018 - Hello World!</title>

</head>
<body>

<script>
<!-- Piilotetaan vanhoilta selaimilta

    document.write("<h1>Terve maailma!</h1>");

// Piilotus päättyy -->
</script>

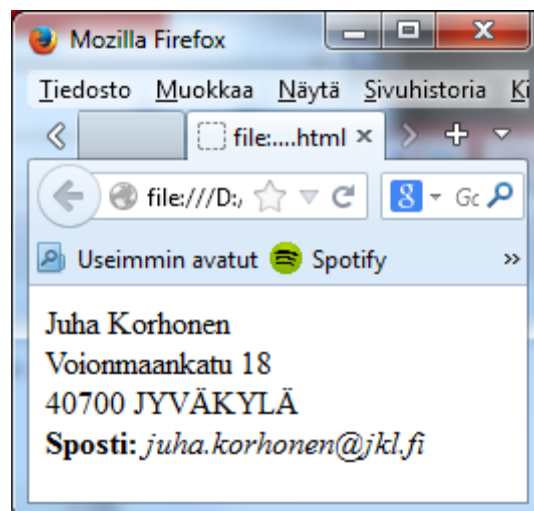
</body>
</html>
```

Avaa tiedosto Notepad++ -ohjelmaan (*hiiren 2-näppäin*). Muuten, voit avata näppäinyhdistelmällä Ctrl+Alt+Shift+x koodisi suoraan Firefox-selaimeen. *Helppo yhdistelmä yhdellä kädellä ;-)*

Tehtävä 2. Muuta edellistä koodia siten, että se tulostaa seuraavanlaisen sivun.

Huomaa, että Sposti-teksti on lihavoitu ja sähköpostiosoite on kursivoitu.

Tallenna tämä sivu **ht1b.htm** -tiedostonimellä **K:\Kotisivu\JavaScript** -kansioosi.



Toinen tunti - Muuttujan tyyppi

Harjoitellaan muuttujien (*ja vakioiden*) käyttöä tulostuslauseessa sekä kokeillaan mitä vaaroja muuttujan tyyppin määräytyminen voi merkitä ohjelmalle.

Tehtävä 3. Kirjoita ensin seuraava koodi ja testaa sen toimivuus. Tallenna tiedosto *K:\Kotisivu\JavaScript* -kansioosi ja anna sille **ht2.htm** -tiedostonimi.

```
<!DOCTYPE html>
<html>
<script type="text/javascript" charset="iso-8859-1">
</script>

<body>

<h2>Muuttujien testausta - pii, säde ja halkaisija</h2>

<script>

var kanta;           // Luodaan muuttuja kanta
var korkeus = 4;    // -----"----- ja sijoitetaan sen arvoksi 4
var sade;
var PII = 3.141;     // Käytä vakio muuttujan nimessä ISOJA kirjaimia
var mjonon1 = "Suorakulmion"; // mjonon1 on merkkijonomuuttuja
var mjonon2 = "Ympyrän";     // mjonon2 on merkkijonomuuttuja
kanta = 5;           // alustetaan muuttuja kanta arvolla 5
sade = 3;

document.write(mjonon1 + " pinta-ala noin " + kanta*korkeus);
document.write(mjonon2 + " pinta-ala noin " + PII*sade*sade);

</script>

</body>
</html>
```

Tehtävä 4. Lisää pari uutta riviä ohjelmaasi. Lisää lähdekoodiisi **sade**-niminen (*kokonaisluku*)muuttuja ja alusta se arvolla 2 eli tee sijoitus `sade=3`.

Lisää ohjelmaasi myös kaksi tulostusriviä (*katso seuraava kuva*), jotka tulostavat *ympyrän säteen* ja *halkaisijan* (**vihje**: halkaisija = 2·säde).



```
<!DOCTYPE html>
<html>
<script type="text/javascript" charset="iso-8859-1">
</script>

<body>

<h1>Muuttujien testausta</h1>

<script>

  /* Ohjelma laskee ympyrän pinta-alan */
  var PII = 3.141; // Käytä vakionmuuttujan nimessä ISOJA kirjaimia
  var sade = 4;
  var yPiiri = 0; // Sijoita tähän muuttujaan ympyrän piiri
  var yAla = 0;   // Sijoita tähän muuttujaan ympyrän pinta-ala
  var mjono = "Ympyrän";
  var rv="<br>"

  /* Tehtävä 1: Muuttujien (vakion) tulostus */
  document.write(mjono + " kehän ja halkaisijan suhde on n. " + PII);
  document.write(rv);
  document.write(mjono + " säde on " + sade);
  document.write(rv);
  document.write(mjono + " halkaisija on " + 2*sade);

</script>

</body>
</html>
```

Tehtävä 5.

Lisää edellisessä esimerkissä olevat rivit, jossa alustetaan yPiiri, yAla ja rv-muuttujat. Muista lisätä myös esimerkissä olevat kommentit, jotta ymmärrät myöhemmin mitä tekemässäsi ohjelmassa tapahtuu.

Testaa mm. rv-muuttujan toimittia, miksi se on kätevä?

Lisää itse tehtävässä 2b rivi, joka laskee ja sijoittaa yPiiri-muuttujaan ympyrän piirin pituuden.

```
/* Tehtävä 2a: Arvon sijoittaminen muuttujaan */
yAla = PII * sade * sade;
document.write("</br>" + mJono + " pinta-ala on " + yAla);
/* Tehtävä 2b: Laske ympyrän piiri */
document.write("</br>" + mJono + " piiri on " + yPiiri);
```

Nyt sinulla pitäisi olla seuraavan kuvan kaltainen tilanne selaimessasi.



Tehtävä 6. Lisää seuraavat rivit edellisen tehtävän jälleen, jossa lasketaan ja sijoitetaan yAla-muuttujaan ympyrän pinta-ala.

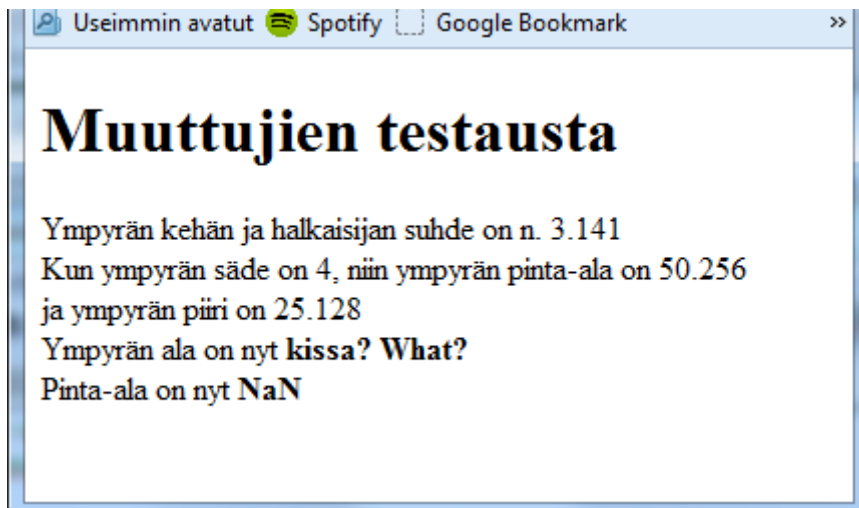
```
/* Tehtävä 2c: Tässä on tehty kielletty sijoitus, miksi? */
yAla = "kissa"; // Älä tee tätä enää ikinä!
document.write(rv + mJono + " ala on nyt <b>" + yAla + "?</b>");
```

Tehtävä 7. JavaScript herjaa edellisessä tehtävssä olevasta virheestä tai ainakin hyvin vaarallisesta ja vääränlaisesta koodaamisesta, niin seuraus voi olla yllättävä.

Kokeile seuraavaa sijoitusta ja kerro mitä tapahtui kun ajoit ohjelman. Mikä mielestäsi tekee edellisen tehtävän muuttujasijoituksen (pAla="kissa") vaaralliseksi?

```
/* Tehtävä 2d: Kerrotaan edellä tehty sijoitus luvulla, mitä tapahtuu? */
yAla = 5 * yAla;
document.write("</br>Pinta-ala on nyt <b>" + yAla + "</b>");
```

Nyt sinulla pitäisi olla seuraavan kuvan kaltainen tilanne selaimessasi.



HUOM! ÄLÄ ENÄÄ KOSKAAN SIOITAN KOKONAISSLUKUMUUTTUJAAN MERKKIJONOA!

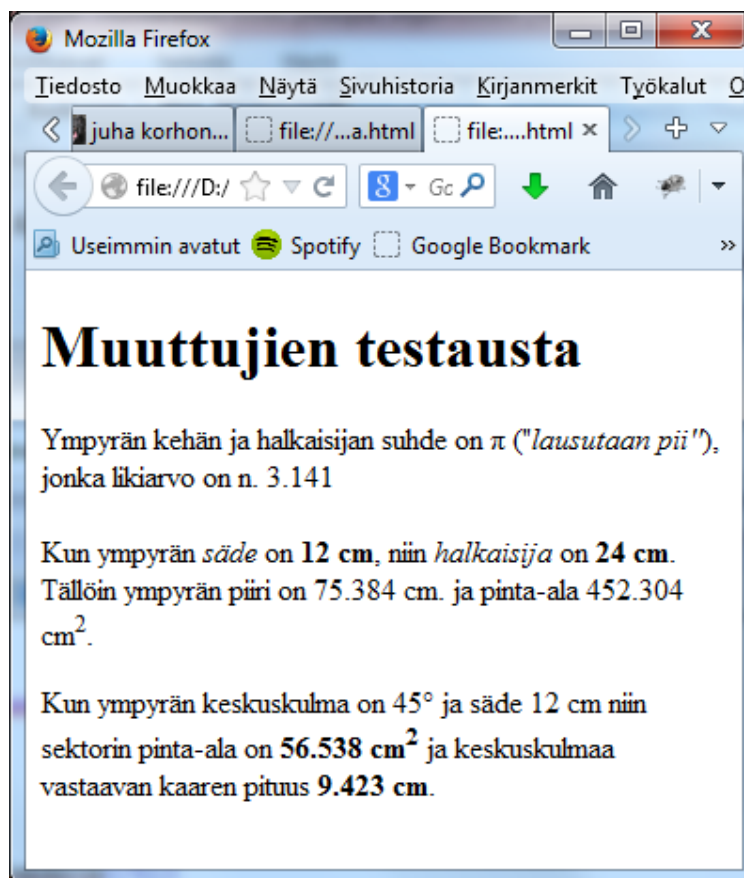
Samaa ei tietenkään tule tehdä toisinkaan päin.

Tehtävä 8. Täydennä tekemäsi ohjelma siten, että se näyttää viereisen kuvan kaltaiselta.

Lihavoi kaikki arvot (tai muuta niiden väriä, esimerkiksi punaiseksi)

- *Asteenmerkin* (45°) saa html:llä aikaan **45°** -tagilla
- **cm²** -merkinnällä tehdään yläindeksi cm^2
- π -merkki saadaan html-sivulle **π**-entiteetillä

(Katso <http://www.cs.tut.fi/~jkorpela/merkit/entiteetit.html> -sivulta oppiaksesi lisää entiteeteistä).



Kolmas tunti Muita tulostustekniikoita

Viime kerralla tulostimme selaimen ikkunaan *document.write* -lauseella. On myös muita tapoja tulostaa tietoa. Tulostaminen voidaan tehdä myös dialogien avulla.

Tehtävä 9. Kokeile seuraavat dialogit ja kirjoita kommentteihin niiden erot, ominaisuudet tai sopivat käyttötarkoitukset.

Tallenna heti ensimmäisen rivin jälkeen tiedosto **ht3.htm** -tiedostonimellä **K:\Kotisivu\JavaScript** -kansioosi.

Kokeile ohjelman toimivuutta heti *document.write* -lauseen kohdalla ja sen jälkeen jokaisen lisäämäsi rivin jälkeen.

```
<!DOCTYPE html>
<html>
<script type="text/javascript" charset="iso-8859-1">
</script>

<body>

<h1>Erilaisia tulostustapaoja</h1>

<script>
  document.write("<h2>Tulostusdialogit</h2>");
  window.alert('Hei maailma!');
  confirm("Varmastiko?");
  prompt("Ollako vai eikö olla", "Olla");
</script>

</body>
</html>
```

Tehtävä 10. Painike, joka käynnistää ohjelman tai komennon.

Huom! Painike ei ole JavaScriptiä, vaan html:lää eli sitä ei sijoiteta sivun `<script>`-tagin sisään.

```
<button type="button" onclick="alert('Terve maailma!')">Klikkaa!</button>
```

Neljäs tunti Käyttäjän kanssa keskusteleva ohjelma

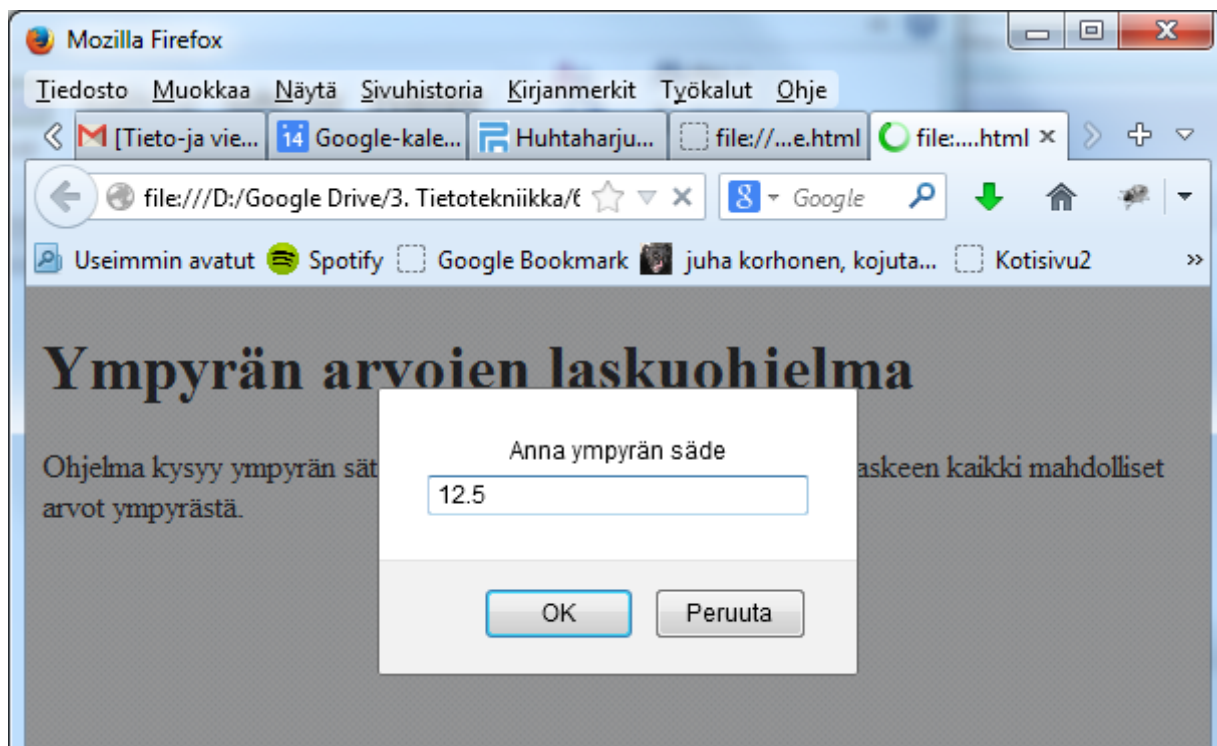
Jo nyt voimme käyttää edellä tekemäämme ympyrän laskentaohjelmaan koulutehtäviin, mutta se edellyttää tietämystä ohjelmoinnista (*joka meillä jo on ☺*) ja pääsyä lahdekoodiin, jotta säteen tai keskuskulman suuruutta päästää muuttamaan.

Tehtävä 11. Muokkaa edelliskerralla tekemäämme ohjelmaa siten, että se kysyy ympyrän **säteen** ja tulostaa sitten ympyrän **pinta-alan**. Käytä seuraavana esiteltyä prompt-dialogia.

```
var sade = prompt("Anna ympyrän säde");
```

Tehtävä 12. Kysy prompt-dialogilla myös ympyrän keskuskulman suuruus. Testaa koodin toimivuutta.

Kirjoita sivulle myös pienet ohjeet siitä, kuin ohjelmaa tulee käyttää. Esimerkiksi desimaaliluvut syötetään desimaalipisteelle eikä suomalaisen tapaan desimaalipilkulla.



Tehtävä 13. Lisää sivun alkuun painike, josta sivu ladetaan uudelleen.

```
<button type="button" onclick="javascript:location.reload(true)">Laske uudelleen</button>
```

Tehtävä 14. Kokeile syöttää luku suomalaisittain (1,2) eli desimaalipilkulla, Mitä tapahtui ja miksi?

Minkälaisia korjauksia ohjelmaan pitäisi tehdä? Helpoin korjata tämä virhe olisi käyttää if-lausetta, joka tarkistaa onko syöte luku vai merkkijono.

Viides tunti Laskeminen JavaScriptillä

Operaattoreita on kolmenlaisia:

- **Binaarioperaattori, esim. $6 * 10$**
- **Unaarioperaattori, esim. $i++$**
- **Trenäärioperaattori, joka vastaa if/else -rakennetta**

Seuraavassa harjoitellaan binaarioperaattoreilla laskemista, jotka toimivat samoin kuin matematiikassa on totuttu.

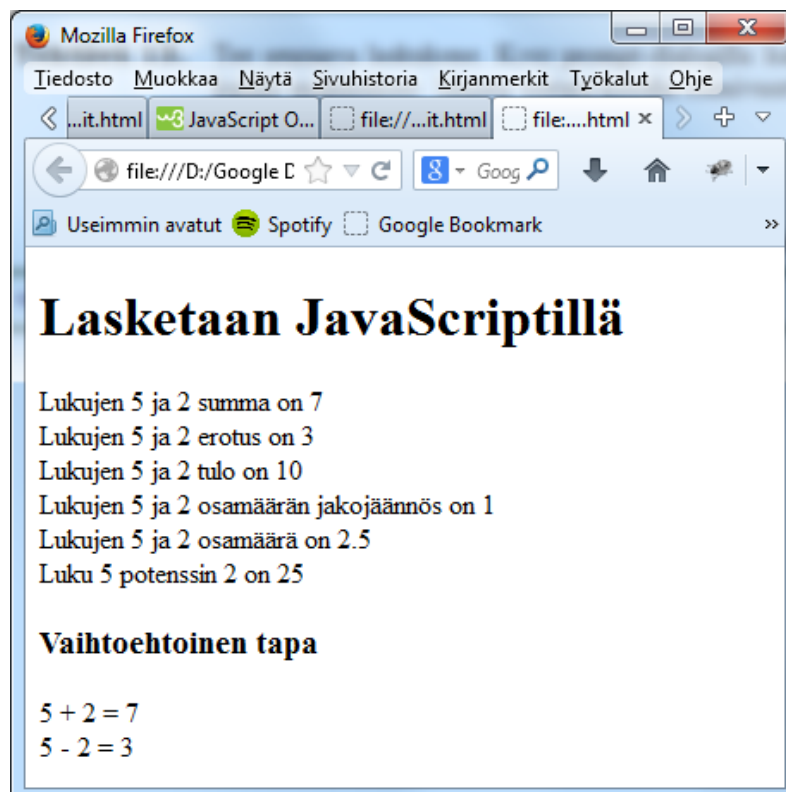
Jopa lisäksi *sidontajärjestys* eli **presedenssi** on se sama mihin peruskoulun matematiikassa on totuttu eli seuraavasta laskusta saadaan tulos: $2 - 2 \cdot 3 : 6 = 2 - (2 \cdot 3) : 6 = 2 - 6 : 6 = 2 - 1 = 1$ eikä esimerkiksi nolla. Laskentajärjestystä voi muuttaa tietenkin suluilla eli edellisessä esimerkissä $(2 - 2) \cdot 3 : 6 = 0$

Tehtävä 15. Tee seuraava laskukone (*katso mallia seuraavasta kuvasta*). Muista testata koodin toimivuutta jokaisen toimivan rivin jälkeen.

```
var luku1 = 5;
var luku2 = 2;
var summa = luku1 + luku2;

document.write("Lukujen " + luku1 + " ja " + luku2 + " summa on " + summa);
document.write("Lukujen " + luku1 + " ja " + luku2 + " tulo on " + tulo);
```

Kun olet saanut ohjelman toimimaan, niin lisää siihen ominaisuus, jolla laskettavat luvut voidaan kysyä.



JavaScriptissä potenssiin korotus tehdään Math-ohjelmakirjaston funktiolla. Huomaa isoalkukirjan Math.pow()-funtiossa.

```
var kantaluku = 4;
var potenssi = 2;
var potenssinArvo = Math.pow(kantaluku, potenssi) // vastaus saa arvon 16
```

Jakojäännös saadaan % -operaattorilla.

```
var jakojaannos = 5 % 2; // muuttuja jakojaannos saa arvon 1
```

Tehtävä 16. Muuta ohjelmaa siten, että se kysyy (esimerkiksi *prompt-dialogilla*) tarvittavat kaksi lukua.

Huomaatko virheen ohjelmassasi?

Tehtävä 17. Muutetaan luku1 ja luku2 muuttujan merkkijonosta (*liuku*) luvuiksi eli desimaaliluvuksi seuraavalla `parseFloat()`-funktiolla. On olemassa myös `parseInt()`-funktio, joka muuttaa merkkijonon kokonaisluvuksi.

```
kysyttyLuku1 = prompt("Anna 1. luku", 5.4);
luku1 = parseFloat(kysyttyLuku1); // luku1 on nyt liukuluku
```

Nyt sinulla pitäisi olla seuraavan näköinen sivu.

Lasketaan JavaScriptillä

Lukujen 5.745 ja 2 summa on 7.745
Lukujen 5.745 ja 2 erotus on 3.745
Lukujen 5.745 ja 2 tulo on 11.49
Lukujen 5.745 ja 2 osamäärän jakojäännös on 1.745
Lukujen 5.745 ja 2 osamäärä on 2.8725
Luku 5.745 potenssin 2 on 33.005025

Pyöristäminen

Luku 5.745 on noin 6
Luku 5.745 on noin yhdellä desimaalilla 5.7
Luku 5.745 on noin kahdella desimaalilla 5.75

Vaihtoehtoinen tapa

$5.745 + 2 = 7.745$
 $5.745 - 2 = 3.745$

Tehtävä 18. Lisää tehtävää. Kokeile lukujen pyöristämistä.

Jokaisessa ohjelmointikielessä on valmiina pyöristämisen tekevä funktio, se toimii seuraavassa esimerkissä esitetyllä tavalla. Valitettavasti funktio pyöristää vain lähimpään kokonaislukuun.

```
var pyorista = Math.round(5.499); // muuttuja pyorista saa arvon 5
pyorista = Math.round(5.511); // muuttuja pyorista saa arvon 6
```

JavaScriptin Math-kirjasto sisältää myös paljon muita matemaattisia laskentaoperaatioita (niin kuin edellä esitelty *Math.pow()*-funktio), joten siihen kannattaa tutustua jossain vaiheessa.

Kuudes tunti Funktiot eli aliohjelmat eli metodit

Funktioiden avulla ”*paketoidaan*” lähdekoodia eli loogisesti yhteenkuuluvia toimintoja kirjoitetaan saman ”*otsikon*” alle ns. funktioiksi, joita kutsutaan myös aliohjelmiksi tai metodeiksi ohjelmointikielestä riippuen.

Tehtävä 19. Seuraava aliohjelma eli *funktio* tulostaa *Hello World* -tekstin. Kokeile sen toimivuutta. Huomaa, että aliohjelman koodi on kirjoitettu {}-merkeillä rahatun *lohkon* sisään.

```
<script>

function helloWorld() {
    document.write("<h1>Hello World!</h1>");
}

/* Seuraavalla rivillä kutsutaan HelloWorld-aliohjelmaa. */
helloWorld();

</script>
```

Tehtävä 20. Kirjoitetaan (*vihdoinkin*) kunnollinen kommenttilohko, jota voi käyttää myöhemmin joikaisessa vakavemmin tehdyssä ohelmassa.

Hyvä tapa on laittaa jokaisen laajemman JavaScript-lohkon ja aliohjelman alkuun kommenttilohko, jossa lyhyesti kerrotaan ao. kohdan tehtävä, tekijä päivämäärä ja parametrit. Tällainen JavaScript-kommenttilohko voi olla seuraavanlainen (ja sen tulee siis sijaita <script>-elementin sisällä):

```
/* *****
 * Funktio:      helloWorld()
 * Parametrit:  -
 * Paluarvo:    -
 * Tehtävä:     Tulostaa Hello World -tekstin
 * Tekijä:      Juha Korhonen - 19.2.2012
 * Ryhmä:       9ATV 4.1
 *              Huhtaharjun koulu: 2014
 * Muokattu:    Lisäsin kommentin 19.3.2014 - Juha K.
 *
 * Testiohjelma: helloWorld();
 *
 * ***** */
```

Jokaisen rivin alussa olevat tähdet ovat vain ihmistä varten, koska ne helpottavat kommenttilohkon hahmottamista, mutta muuten ne ovat siis turhat.

Seitsemäs tunti Aliohjelmien parametrit

Parametrien välityksellä aliohjelmalle voidaan välittää tietoa.

Tehtävä 21. Tehdään hyvin yksinkertainen aliohjelma, joka on kuitenkin käyttökelpoinen. Miksi?

```
<script>
/*
 * Funktio lisää rivinvaihdon ja tulostaa
 * parametrina annetun merkkijonon.
 */

function tulosta(lause) {
  document.write("<br>");
  document.write(lause);
}

/* Testiohjelma */

var luku = 4.22;
var merkkijono1 = "Terve";
var merkkijono2 = "maailma!";

tulosta("Hello World!");
tulosta(luku);
tulosta("Lukun on " + luku);
tulosta(merkkijono1 + merkkijono2); // merkkijonojen yhdistäminen

</script>
```

Kahdeksas tunti Aliohjelmien kirjoittaminen erilliseen tiedostoon

Tehdään *kokoelma.js* -tiedosto, joka sisältää kaikki useasti tarvitsemasi funktiot. Tallenna *kokoelma.js* -tiedosto **K:\Kotisivu\JavaScript** -kansioon.

Tehdään myös *testi.html* -tiedosto, jossa kutsutaan *kokoelma.js* -tiedostoa.

Tehtävä 22. Kirjoita seuraava JavaScript-lähdekoodi, joka tulostaa parametrin ryyditettynä rivinvaihdolla. Tallenna funktio *kokoelma.js* -tiedostonimellä **K:\Kotisivu\JavaScript** -kansioon.

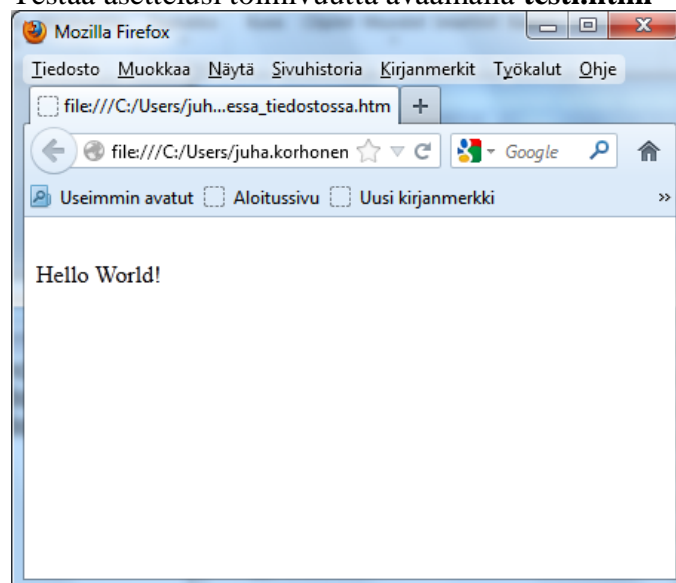
```
/* kokoelma.js -tiedosto */  
function tulosta(Teksti) {  
    document.write(Teksti);  
    document.write("</br>");  
}
```

Tehtävä 23. Kutsu tekemääsi **tulosta()**-funktiota erillisessä html-tiedostossa.

Kirjota seuraava html-koodi, tallenna se *K:\Kotisivu\JavaScript* -kansioon ja anna tiedostolle nimeksi *testi.html*

```
<html>  
<head>  
<script type="text/javascript" charset="iso-8859-1" src="kokoelma.js">  
</script>  
</head>  
  
<body>  
<script type="text/javascript" language="JavaScript">  
    tulosta("Hello World!");  
</script>  
</body></html>
```

Tehtävä 24. Testaa asettelusi toimivuutta avaamalla **testi.html** -tiedosto selaimen.



Yhdeksäs tunti Funktioiden palautusarvo

Tehdään aliohjelma, joka palauttaa jonkin arvon eli **return-lauseen** käyttö.

Tehtävä 25. Tehdään *lueLuku(Teksti)* -funktio *kokoelma.js* -tiedosto, joka kysyy luvun ja palauttaa sen paluuarvossa.

```
/* kokoelma.js -tiedosto */
function tulosta(Teksti) {
  document.write("<br>");
  document.write(Teksti);
}

function lueLuku() {
  var input=0;
  input = prompt("Syötä luku");
  return input;
}
```

Testaa *lueLuku()* -funktion toimivuutta lisäämällä *testi.html* -tiedostoon seuraava lähdekoodin pätkä.

```
<html>
<head>
<script type="text/javascript" charset="iso-8859-1" src="kokoelma.js">
</script>
</head>
<body>
<script type="text/javascript" language="JavaScript">

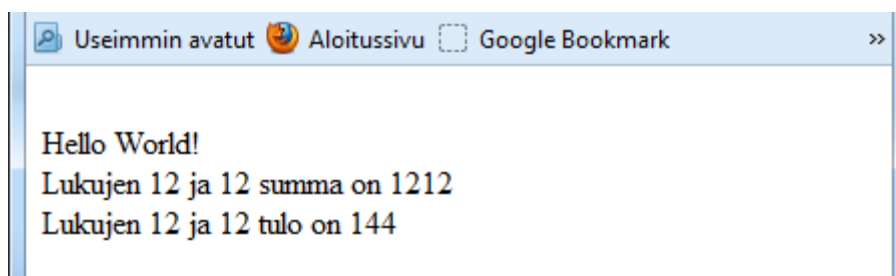
  tulosta("Hello World!");
  /* alustetaan luku1- & luku2-muuttuja numeroksi */
  var luku1=0;
  var luku2=0;

  luku1 = lueLuku();
  luku2 = lueLuku();

  /* Voit kokeilla laskentaa myös ilman sulkuja. */
  tulosta(luku1 + " + " + luku2 + " = " + (luku1 + luku2));
  tulosta(luku1 + " * " + luku2 + " = " + luku1*luku2);
  tulosta(luku1 + " &#183 " + luku2 + " = " + luku1*luku2);
</script>

</body>
</html>
```

Tulos pitäisi olla jotain seuraavan kuvan kaltaista. Kuten huomaat, summan laskeminen ei onnistu suoraan tulostelausekkeessa, vaan luku1 ja luku2 käsitellään merkkijonoina.



Muutetaan lueLuku()-funktion return-lause seuraavaan muotoon, joka pakottaa funktion palauttaman tuloksen liukuluvuksi eli

```
function lueLuku() {  
  var input=0;  
  input = prompt("Syötä luku");  
  return parseFloat(input);  
}
```

Tehtävä 26. Kirjoitetaan aliohjelma, joka pyörittää edelle tekemämme ympyräohjelman laskentatulokset haluttuun tarkkuuteen.

Apuna käytetään **Math.round()** -metodia (*huomaa iso kirjain alussa*), joka pyöristää desimaaliluvun lähimpään kokonaislukuun.

```
var luku1 = 5.4;  
var luku2 = 5.5;  
  
var tulos = Math.round(luku1); // tulos saa arvon 5  
tulos = Math.round(luku2); // nyt tulos saa arvon 6
```

Katso aliohjelman rakenne seuraavasta esimerkistä, mutta kirjoita ohjelman tarvittava sisältö itse.

```
function pyorista2(luku) {  
  
  return pyoristysKahdellaDesimaalilla;  
}
```

Vihjeenä annettakoon, että kokeilu luku-muuttujan kertomista kymmenellä.

Voit tehdä myös pyorista1(luku) -aliohjelman, joka palauttaa luvun pyöristettynä yhdellä desimaalilla.

```
function pyorista1(luku) {  
  
  return pyoristysYhdellaDesimaalilla;  
}
```

Kaikkein paras olisi tietenkin aliohjelma, johon voisi antaa parametrina desimaalien lukumäärän seuraavalla tavalla:

```
function pyorista(luku, desimaalienmaara) {  
  
  return pyoristettuLuku;  
}
```

Kymmenes tunti jos-lause

Valintalauseet ovat tärkeä osa ohjelmointia. ensimmäisenä tutustutaan jos-ehdolauseeseen. Yksinkertaisin ehdolause on, jossa lause suoritetaan, jos ehto on tosi (*true*).

```
if (ehto) {
  lause
}

/* if-lauseen jälkeen jatketaan tästä. */
```

Hieman monipuolisempi versio on if-else –lause.

```
if (ehto) {
  lause1
}
else {
  lause2
}

/* if-lauseen jälkeen jatketaan tästä. */
```

Edellä olevassa esimerkissä *lause1* suoritetaan, jos ehto on tosi (*true*). Ehdon ollessa epätosi (*false*), tehdään *lause2*.

Tehtävä 27. Tee edellistä if-lausetta hyödyntäen ohjelma, joka kysyy ikäsi. Jos olet alle 18-vuotias, tulostetaan alaikäinen.

```
<script language="javascript">
  var ika = 10;

  if(ehto tähän) {
    alert("Alaikäinen!");
  }
  else {
    alert("Täysikäinen"); // tämän voi myöhemmin ottaa pois
  }
</script>
```

Tehtävä 28. Muunna edellistä ohjelmaa siten, että se kysyy iän.

Tehtävä 29. Kirjoita `lueLuku()`-funktio. Return-lauseessa pakotetaan funktio palauttamaan luku kokonaislukuna.

```
function lueLuku() {
  var input = 0;
  input = prompt("Syötä luku");
  return parseInt(input);
}
```

Lisää edellinen koodinpätkä *kokoelma.js* -tiedostoon, jollin se on käytettävissä kaikissa kirjoittamissasi ohjelmissa.

Tehtävä 30. Muutetaan *lueLuku()*-funktiota siten, että se reagoi väärään syötteeseen, esimerkiksi seuraavalla tavalla:

```
<script language="javascript">
  var x=10;

  /* x==" " eli jos ei ole annettu mitään arvoa.
   *
   * isNaN(x) palauttaa arvon true, jos x ei ole luku.
   * Funktion luetaan seuraavasti: x is Not an Number.
   */

  if(x==" "||isNaN(x)) {
    alert("Kentässä oleva luku ei ole numero!");
  }
  else {
    alert(x + " on luku! "); // tämän voi myöhemmin ottaa pois
  }
</script>
```

Tehtävä 31. Kirjoita ohjelma, joka kysyy iän ja kertoo sen jälkeen ajoneuvoluokan, johon käyttäjällä on oikeus.

Katso ikäraajat <https://fi.wikipedia.org/wiki/Ajokortti> -sivulta. Esim.

Ikä	kortti
15	T
15	AM
16	A1
18	A2
24	A

Yhdestoista tunti For-lause

For-lauseessa on seuraava syntaksi: **for (var i = 1; i < 11; i++)** eli for(*muuttujan alustus; toistoehto; muuttujan kasvatus/vähennys*).

Tehtävä 32. Tulostetaan luvut 1-100

```
<script language="javascript">
for (var i = 1; i <100; i++) {
  tulosta(i);
}
</script>
```

For lause voidaan myös kirjoittaa seuraavalla tavalla. [] -merkintä tarkoittaa JavaScriptissä taulukkoa.

```
<script language="javascript">
// Vaihtoehtoinen tapa kirjoittaa for-lause
for (var index in [1, 2, 3, 4, 5]) {
  tulosta(index);
}
</script>
```

Tehtävä 33. Tulostetaan parilliset luvut väliltä 1-100.

Tehtävä 34. Tulostetaan parittomat luvut väliltä 1-100.

Tehtävä 35. Tee ohjelma, joka tulostaa laskee yhteen kaikki luvut 1-100.

Eli haluamme laskea kaikki luvut 1-100 yhteen. Toki matematiikka lukeneena voimme itsekin laskea tuloksen (*aritmeettinen sarja* $S_n = \frac{n(a_1+a_n)}{2}$, missä n on lukujen lkm, a_1 1. luku ja a_n viimeinen luku), mutta tällä kertaa käytämme tietokoneen laskentatehoa ja for-lausetta ratkaisemaan ongelman.

```
<script language="javascript">
var summa = 0; // alustetaan summa-muuttuja

for (var i = 1; i <=100; i++) {
  summa = summa + i;
  tulosta("Summa on " + summa);
}

tulosta("Lukujen 1-100 summa on " + summa);

</script>
```

Lasketaan JavaScriptillä

Lasketaan luvut 1-100 yhteen.

```
01 02 03 04 05 06 07 08 09 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
```

Lukujen summa on 5050

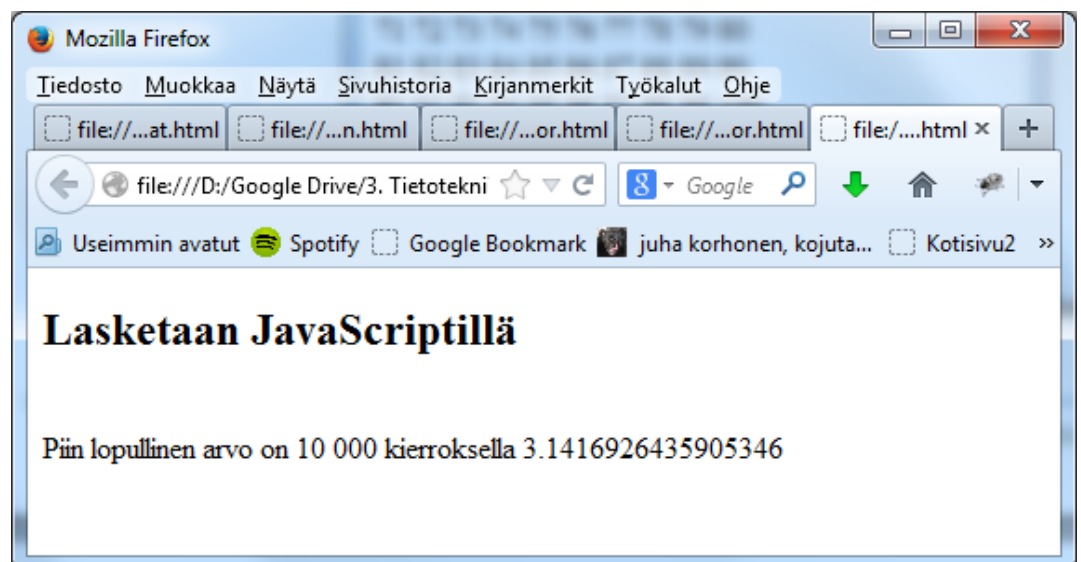
- Muuta ohjelmaa siten, että se tulostaa aina 10 numeroa yhdelle riville (*katso edellinen kuva*)
- Lisää 0 tai tyhjä merkki (alt+255) luvan alkuun, jos luku on <10

Tehtävä 36. Tee ohjelma, joka laskee summaa kaikki luvut 1-1000. Paljonko oli tulos?

Muuta edellisiä ohjelmia siten, että se kysyy mihin asti lukuja lasketaan yhteen.

- Tehtävä 37.** Tee toinen (ali)ohjelma, joka laskee luvun 10 kertoman eli $10! = 10 \cdot 9 \cdot 8 \cdot \dots \cdot 2 \cdot 1$. Muunna ohjelmaa siten, että se kysyy luvun, jonka kertoma lasketaan. Funtion (*eli aliohjelman*) kutsu voisi olla muotoa: **kertoma(luku)**. Lisää myöhemmin ohjelma **kokoelma.js** -tiedostoon.
- Tehtävä 38.** Tee ohjelma, joka laskee summaa kaikki luvut 2, 4, 6, ..., 1000 eli kaikki tuhatta pienemmät parilliset luvut.
- Tehtävä 39.** Tee ohjelma, joka laskee summaa kaikki luvut 1, 3, 5, ..., 9999 eli kaikki tuhatta pienemmät parittomat luvut.
- Tehtävä 40.** Tee ohjelma, joka tulostaa kaikki alkuluvut 1-50 eli luvut 1, 2, 3, 5, 7, 11, 13, 17,...
- Tehtävä 41.** Tee ohjelma, joka laskee piin likikarvon seuraavalla kaavalla.

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots$$



Kahdestoista tunti Kolikon heitto

Kuten edellisessä tehtävässä huomasimme voimme panna tietokoneen tekemään asioita, joita ei oikeasti ole mahdollista tehdä, kuten heittää noppaa miljoona kertaa.

Tehtävä 42. Kokeillaan JavaScriptin *Math.random()* -satunnaislukufunktiota. Se tuottaa satunnaisluvun 0-1 väliltä.

```
function sLuku() {  
  return Math.random();  
}  
  
tuosta("Satunnaisluku " + sLuku());
```

- Kokeile ensin mitä edellinen funktion tekee
- Lisää edelliseen esimerkkiin for-silmukka, jotta ohjelmasi tulostaa 10 satunnaislukua.
- Lisää edelleen ohjelmaan kysymys, jolla voidaan määrätä kuinka monta satunnaislukua arvotaan.

```
input = prompt("Anna arvottavien satunnaislukujen määrä:");
```

Tehtävä 43. Muutetaan *sLuku()* -aliohjelmaamme siten, että voimme simuloida kolikonheittoa, jolloin *sLuku()* -aliohjelmaa pitäisi muuttaa siten, että se tuottaa aina joko satunnaisluvun 1 = **kruuna** tai 2 = **klaava**.

- Tähän on kaksi eri keinoa joko pyöristäminen tai if-lause. Kokeillaan tänään if-lausetta. Eli jos satunnaisluku on alla 0,5 palutetaan **nolla** (0), muutoin palautetaan **yksi** (1).

```
if (ehto tosi) return 0;  
else return 1;
```

- Pane ohjelma heittämään 100 kertaa ja katso onko todennäköisyys todella 50% sille, että saadaan kruuna.

Vihje: tee muuttuja, jota lisätään joka kierroksella aina yhdellä, kun on arvottu kruuna.

```
var kruuna = 0;  
// tämä tulee for-lauseeseen  
kruuna = kruuna + 1;
```

Tehtävä 44. Olen myös kuullut väitteen, että jos heitetään kolikkoa miljoona kertaa, niin jossain vaiheessa heittosarjaa tulee 20 kertaa peräkkäin klaava.

- Todista väite muutamalla edellistä JavaScript ohjelmaa siten, että se tallentaa peräkkäiset heitot silloin, kun ne ovat klaava.
- Ohjelma voisi tulostaa myös kuinka pitkä oli pisin sarja, jossa oli klaavoja peräkkäin.

Tehtävä 45. Muutetaan *sLuku()* -aliohjelmaamme siten, että voimme simuloida jokeriarvontaa. *sLuku()* -ohjelmaa pitäisi muuttaa siten, että se tuottaa aina satunnasluvun 0-6.

Apuna voimme käyttää *Math.floor()* -funktiota, jolla voimme katkaista desimaaliluvusta desimaalit pois. Esimerkiksi sijoituksella

```
var sluku = Math.floor(6.258);  
tulosta(sluku); // tulostaa luvun 6
```

Ensin kannattaisi miettiä, että kuinka saisimme muutettu *sLuku()* -funktiota siten, että se tuottaisi satunnaisiadesimaalilukuja väliltä $0 < \text{sluku} < 10$? Tällöin voisimme poistaa tarpeettomat desimaalit edellä kuvatulla *Math.floor()* -metodilla.

Kolmastoista tunti - for-lause ja satunnaisluvut

Lisätään kokoelmaamme satunnaisLuku(min, maks) -funktio, joka arpoo satunnaislukuja.

```

/* Palauttaa satunnaisluvun väliltä [min, max]
 *
 * var sluku=satunnaisLuku(1, 6);
 * document.write(sluku); // tulostaa luvun 1-6
 *
 */
function satunnaisLuku(min, maks) {
  var sluku = maks * Math.random() + min;
  return Math.floor(sluku);
}

```

Tehtävä 46. Testaa *satunnaisLuku(min, maks)* -funktion toimivuutta lisäämällä *testi.html* -tiedostoon seuraava **for**-silmukan pätkä.

```

for (var i=0; i<10; i++) {
  var sluku = satunnaisLuku(1, 6);
  document.write(i + ". ");
  tulosta(sluku); // tulostaa satunnaisluvunluvun 1-6
}

```

Viereisessä kuvassa on vähän kikkailtu tulostuksen kanssa, **Nro**-sarakeeseen on tulostettu i:n arvo ja **Sluku**-sarakeessa on tietenkin tulostettu joka kierroksella arvottu satunnaisluku yhdestä kuuteen (1-6).

Tehtävä 47. Kokeile tulostaa satunnaislukuja muulta kuin ykkösellä alkavalla välillä, esimerkiksi 5-12.

Yritä korjata satunnaislukufunktiossa oleva virhe. Mieti miten kerrointa pitäisi muuttaa, jos satunnaislukujen alarajaa halutaan nostaa.

Kun saat tämän korjattua, kokeile vielä miten satunnaislukufunktio toimii esimerkiksi ykkösellä (1) alkavalla välillä. Korjaa vielä tämäkin muuttamalla kerrointa.

Tehtävä 48. Muuta *satunnaisLuku()*-funktioita siten, että voit antaa parametrina sekä minimin että maksimin eli funktiota kutsutaan seuraavanlaisesti *satunnaisLuku(min, max)* eli

Nro	Sluku
0.	5
1.	2
2.	6
3.	2
4.	4
5.	5
6.	4
7.	6
8.	5
9.	5

```

var sluku = satunnaisLuku(5, 100) // sluku saa jonkin arvon väliltä 5-100

```

- 1)
- 2)
- 3)

Tehtävä 49. Lisätään kokoelmaamme (=kokoelma.js -tiedosto) vielä edellisessä tehtävässä tehty *satunnaislukufunktioaliohjelma*.

Muutetaan funktiota siten, että siihen voi parametrina antaa välin, jolta satunnaislukuja annetaan. Jolloin sitä kutsutaan:

```
/* Palauttaa satunnaisluvun väliltä [min, max]
 *
 * var sluku=satunnaisLuku(0,11);
 * document.write(sluku); // tulostaa luvun 0-11
 *
 */
function satunnaisLuku(min, maks) {
  var sluku = (maks - min + 1 ) * Math.random() + min;
  return Math.floor(sluku);
}
```

Muista testata satunnaislukufunktion toimivuutta eri vaihtoehdoilla!

Neljästoista tunti - jos-lause eli if()-funktio

Tehtävä 50. Nyt voisimme tarkistaa kuinka satunnainen viime tehtävässä käyttämämme JavaScriptin satunnaislukugeneraattori on.

Tuloste voisi näyttää eri syötteillä seuraavalta:

Arvauksia 10		Arvauksia 100 000	
Ykkösiä	14,3 %	Ykkösiä	16,3 %
Kakkosia	8,0 %	Kakkosia	16,7 %
Kolmosia	22,3 %	Kolmosia	16,6 %
...		...	
Arvauksia 100			
Ykkösiä	15,3 %		
Kakkosia	12,0 %		
Kolmosia	14,3 %		
...			

Eli talleta ykkösten, kakkosten jne lukumäärä johonkin muuttujaan, esimerkiksi ykkostenLkm, kakkostenLkm jne.

Tätä varten tarvitset if-funktion, se toimii seuraavasti:

```
if(sluke==1) {  
  ykkostenLkm = ykkostenLkm + 1;  
}
```

Viidestoista tunti Tekstilohkon päivittäminen uudella tiedolla

Tehtävä 51. Tehdään ohjelma **jokeri.html**, joka arpoo jokeririvin. Jokeririvissä on 7 numeroa. Tulostetaan arvatut numerot suoraan html-sivulle nimettyihin kenttiin. Tällöin muuhun sivun sisältöön ei tarvitse koskea.

- 4) Käytetään kentän rajaukseen ``-tagia. Aivan yhtä hyvin voisimme käyttää `<p>`-tagia eli kappaleenvaihtoa, mutta tämä sopii paremmin käyttötarkoitukseemme, koska haluamme tuloksen yhdelle riville.

```
<font id="arvottu1">1</font>
```

- 5) Edellä nimettyyn kenttään voidaan sijoittaa uusi arvo seuraavalla tavalla.

```
document.getElementById("arvottu1").innerHTML=7;
```

- 6) Sijoitetaan *arvottu1*-kentän päivitys painikkeen taakse eli tehdään kenttäpäivitysfunktio, joka käynnistyy painiketta napauttamalla. Annetaan funktion nimeksi esimerkiksi `arvoJokeri()`.

```
<button type="button" onclick="arvoJokeri()">Arvo jokeri!</button>
```

- 7) Kirjoita seuraava lähdekoodin pätkä, jossa nämä edelliset asiat on yhdistetty ja tallenna se `K:\Kotisivu\JavaScript` -kansioon `jokeri.html` -nimellä.

```
<!DOCTYPE html>
<html>
<body>

<h1>Arvotaan 7 jokerinumeroa!</h1>

<font id="arvottu1">1</font>

<script>
function arvoJokeri() {
  document.getElementById("arvottu1").innerHTML=7;
}
</script>

<button type="button" onclick="arvoJokeri()">Käynnistä!</button>

</body>
</html>
```

- 8) Testaa koodin (=painikkeen) toimivuutta.
- 9) Liitä ennen tätä *kokoelma.js* -tiedosto osaksi **jokeri.html** -ohjelmaa, niin saat myös viime kerralla tekemämme `satunnaisLuku()`-funktion käyttöösi.

```
<head>
<script type="text/javascript" charset="iso-8859-1" src="kokoelma.js">
</script>
```

- 10) Muuta nyt ohjelmakoodia siten, että se arpoo luvun 7 paikalle jokin satunnaisluvun 0-9 eli kutsu `satunnaisLuku(0, 9)`

- 11) Lisää loput kuusi kenttää nimeä ne esimerkiksi arvottu2-arvottu7
- 12) Arvo myös näihin uusiin kenttiin satunnaisluvut 1-9
- 13) Voit muotoilla kenttiä koon ja värin mukaan sen verran kun -tagi antaa myöten.

Kuudestoista tunti Tekstikentän luominen ja lukeminen

Tehtävä 52. Tehtävänä on luoda sivu, jossa annetaan oma jokeririvi, arvotaan jokeririvi sekä tarkistetaan montako omassa rivissä on oikein verrattuna arvottuun riviin?

Omien lukujen lukemiseen voitaisiin käyttää **lueLuku()**-funktioita, joka tehtiin viime kerralla, mutta tehdään tällä kertaa tekstikentät, joiden arvot sitten luetaan ja verrataan *satunnaislukufunktiolla* arvottuihin numeroihin.

Myöhemmin syötetyille arvoille voisi tehdä myös tarkistuksen, että ne ovat varmasti lukuja. Katso ohjeita JavaScript-oppaasta *Tekstikentän lukeminen - onko syöte luku?* -luvusta)

14) Ratkaisuehdotus:

15) Seuraavassa on tehty sivulle kenttä, johon voi syöttää tietoa. Kentän koko on rajoitettu yhteen merkkiin ja sille on annettu *arvattu1*-tunnisteeksi.

```
<!-- Tehdään tekstikenttä ja annetaan sille arvattu1-nimeksi. -->
<input id="arvattu1" type="text" size="1" maxlength="1">
```

16) Tällaiselta kentältä tietoa luetaan seuraavanlaisella proseduurilla

```
var x=document.getElementById("arvattu1").value;
```

17) Testaa koodin toimivuutta esimerkiksi tulostamalla x-muuttuja.

18) Vertaa 1. arvotun luvun ja antamasi luvun yhtä suuruutta, jos ne ovat yhtä suuria, niin ilmoita osuma esimerkiksi tulostuslauseella: ”Rivissä oikeita numero x kpl” tai joitain vastaavaa.

```
// Pyydystetään ensin kentän arvo muuttujaan x
var x=document.getElementById("arvattu1").value;
// Testataan onko arvottu luku sama kuin arvauksemme?
if(x==arvottuLuku) {
    alert("osuma");
}
else {
    alert("huti");
}
```

19) Voit myös muuttaa sen kentän väri, joka on asunut oikeaan.

20) Lisää kenttiä tarvittava määrä 7 numerolle eli tee lisää *arvattu2-arvattu7* -kentät ja tee niille sama vertailu.

21) Voit muuttaa lähdekoodia siten, että se arpoo haluamasi määrän rivejä ja tulostaa ne rivit, joissa on enemmän kuin kolme oikein.



Tehtävä 53. Tehdään totuusfunktio, joka tarkistaa on muuttuja luku vai ei? Funktio on

tosi, jos syöte on luku ja epätosi, jos ei.

```
/* Palauttaa totuusarvon onko syöte luku (true) vai ei (false)?
 *
 * if(onkoNumero(100)) document.write("kyseessä on luku!");
 * else document.write("kyseessä ei ole luku!");
 * if (onkoNumero(a)) document.write("kyseessä on luku!");
 * else document.write("kyseessä ei ole luku!");
 *
 */
function onkoLuku(x) {
  if(x==" "||isNaN(x)) {
    //alert("Kyseessä ei ole luku!");
    return false;
  }
  else {
    //alert(x + " on luku! ");
    return true;
  }
}
```

Katso edellisen koodin kommentteista kuinka onkoLuku()-funktiota käytetään.

Seitsemästoista tunti

Taulukot

Tehtävä 54. Aluksi tehdään *arvoJokeriNumerot()*-funktio, joka arpoo tekemällämme *satunnaisLuku()*-funktioilla seitsemän (7) satunnaislukua **sLuvut**-nimiseen taulukkoon, joka sitten palautetaan funktion paluu arvossa.

```
/******  
* Funktio:          arvoJokeriNumerot()  
* Paluarvo:        Taulukko, jossa on 7 lukua.  
* Tehtävä:         Täyttää taulukon 7 satunnaisluvuilla 0-9  
* Tekijä:          Juha Korhonen - 5.4.2013  
*                  Huhtaharjun koulu: 2012 - 2013  
* Muokattu:  
*  
* Testiohjelma:    tulostaTaulukko(arvoJokeriNumerot());  
*  
*****/  
  
function arvoJokeriNumerot() {  
    var sLuvut = new Array();  
    for (var i=0; i<7; i++) {  
        sLuvut[i]=satunnaisLuku(0,9);  
    }  
    return sLuvut;  
}
```

Tehtävä 55. Toiseksi tehdään *tulostaTaulu(taulukko)* -funktio, joka tulostaa taulukon näytölle. Sen parametrina annetaan tulostettava taulukko.

22) Ratkaisuehdotus:

23) Luo for-lause tulostamiseen.

24) For-lauseen lopetusehto eli taulukon koko, saadaan *taulukko.lenght*-funktioilla.

25) Käytä yksittäisen taulukon alkion tulostamiseen aikaisemmin tekemäsi **tulosta()**-funktioita.

26) Testaa funktiota edellisessä tehtävässä tehdyllä satunnaislukutaulukolla, esimerkiksi näin:

```
tulostaTaulu(arvoJokeriNumerot());
```

27) Lisää tekemäsi ja testaamasi funktion kokoelma.js -tiedostoon myöhempää käyttöä varten.

Tehtävä 56. Lisäksi tehdään funktio, joka päivittää jokerinumerot nimettyihin ``-kenttiin.

```
/* Tässä päivitettävän kentän nimi on
 * generoitu puoliksi automaattisesti.
 *
 * kNimi on luettavan kentän nimi eli arvottu ja
 * nimen loppuun lisätään aina vaihtuva numero 1-7.
 *
 */

for (var j=1; j<8; j++) {
  apu = kNimi + (j);
  document.getElementById(apu).innerHTML=sLuvut[j];
}
```

28) Alusta **apu**-muuttuja merkkijonoksi (var apu=""); ja anna sopiva oletusarvo **kNimi**-muuttujalle.

29) Toinen vaihto ehto on tehdä 7 samanlaista lausetta ja antaa kentän nimi aina käsin.

Kahdeksastoista tunti

Taulukkojen vertailu

Tämän kerran idea on lukea sekä arvotut jokerinumerot että meidän oma jokeririvi kahteen taulukkoon ja sitten verrata näitä arvoja keskenään ja tulostaa mahdollinen oikea rivi.

Tehtävä 57. Varmista ja lisää tarvittaessa, että sinulla on tulostaTaulukko()-funktio *kokoelma.js* -tiedostossa.

```
/* Tulostaa parametrina välitetyn taulukon! */
function tulostaTaulukko(taulukko) {
  for (var i = 0; i < taulukko.length; i++){
    tulosta(taulukko[i]);
  }
}
```

Varmista myös, että sinulla on pääohjelmassasi (=jokeri.html) viimeksi tehty funktio, joka arpoo tarvittavat satunnaisluvut.

```
/******
 * Funktio:      arvoJokeriNumerot()
 * Paluuarvo:   Taulukko, jossa on 7 lukua.
 * Tehtävä:     Täyttää taulukon 7 satunnaisluvuilla 0-9
 * Tekijä:      Juha Korhonen - 5.4.2013
 *              Huhtaharjun koulu: 2012 - 2013
 * Muokattu:
 *
 * Testiohjelma: tulostaTaulukko(arvoJokeriNumerot());
 *
 * *****/

function arvoJokeriNumerot() {
  var sLuvut = new Array();
  for (var i=0; i<7; i++) {
    sLuvut[i]=satunnaisLuku(0,9);
  }
  return sLuvut;
}
```

Tämän lisäksi meillä pitäisi olla valmiina jo -kentät (7 kpl), joihin arvotut jokerinumerot tulostetaan

```
<font id="arvottu1">1</font>
.
.
.
<font id="arvottu7">1</font>
```

Ja edelleen valmiina piti olla tekstikentät (7 kpl), joihin saattoi syöttää oman jokeririvin.

```
<input id="arvattu1" type="text" size="1" maxlength="1">
.
.
.
<input id="arvattu7" type="text" size="1" maxlength="1">
```

Varmista siis, että sinulta löytyvät nämä kaikki, ennen kuin jatkat seuraaviin harjoituksiin.

Tehtävä 58. Tehdään funktio, joka lukee tekstikentästä syötetyt arvot

arvatutLuvutTaulukko[]-nimiseen taulukkoon.

Tämän voi tehdä myös näppärimmin for-silmukalla, mutta tehdään raakaversio, joka varmasti toimii näin.

```
// Luodaan arvatutLuvutTaulukko[]-taulukko. Katso malli viime tunneilta!
. . .

// Pyydystetään kenttien arvot taulukkoon (7kpl).
arvatutLuvutTaulukko[0]=document.getElementById("arvattu1").value;
arvatutLuvutTaulukko[1]=document.getElementById("arvattu2").value;
.
.
.
```

Tehtävä 59. Luodaan totuus-funktio (), joka vertaa kahta lukua eli jos luvut ovat samoja palautetaan TRUE, muutoin FALSE.

30) Ratkaisuehdotus:

31) Tee ensin yksinkertainen testifunktio, joka erottelee **jos**-lauseella edellä mainitut kaksi eri vaihtoehtoa.

```
function onkoSama(luku1, luku2) {
// Testataan onko arvottu luku sama kuin arvauksemme?
  if(luku1==luku2) {
    // alert("osuma");
    return true;
  }
  else {
    //alert("huti");
    return false;
  }
}
```

32) Testaa funktiota esimerkiksi luvuilla 1 ja 2 tai 175 ja 175 ja seuraavalla koodinpätkällä.

```
if(onkoSama(175, 175)) alert("osuma");
else alert("huti");

if(onkoSama(1, 2)) alert("osuma");
else alert("huti");
```

Tehtävä 60. Tehdään funktio, joka tarkistaa vastaavatko taulukossa olevat luvut toisiaan. Taulukossa välitetään parametrina kaksi taulukkoa ja palautusarvona välitetään yhteisten numeroiden lukumäärä.

Tämä funktio tekee kaksi asiaa kerralla eli vertaa soluja ja palauttaa osuimien lukumäärän. **TÄMÄ ON HUONOA OHJELMOINTIA!**

Koska aina, kun mahdollista, tehtävä pyritään jakamaan mahdollisimman pieniin osiin eli yksi funktio tekee VAIN yhden tehtävän.

Koeta soveltaa seuraavaan funktioon edellä tehtyä onkoSama()-funktioita.

```
/* Funktio vertaa kahta taulukkoa keskenään ja palauttaa
 * oikeiden arvauksien määrän.
 */
function vertaaTaulukot(taulu1, taulu2) {
  var osumat=0;
  for (var i=0; i<taulu1.length; i++) {
    if(taulu1[i]==taulu2[i]) {
      osumat = osumat + 1;
      //testi: tulosta(taulu1[i]);
    }
    else
      return osumat;
  }
  return osumat;
}
```

Tehtävä 61. Näiden kaikkien toimien lisäksi tarvitaan sivun html-osaa kenttä, johon tulokset tulostetaan. Lopputulos voisi olla esimerkiksi seuraavan kuvan kaltainen.

```
<p><button type="button" onclick="arvoJokeri()">Arvo jokeri!</button></p>
<p>Saatiin <font id="tulos" color=red>x</font> oikein tulos kierroksella
<font id="voittotulos">x</font>, kun arvontoja suoritettiin
<font id="kierrokset">x</font> kpl.</p>
```

Niin ja tietenkin tarvitaan arvoJokeri()-funktioon seuraavat lisäykset, joissa omarivi ja arvotturivi luetaan kahteen eri taulukkoon, jonka jälkeen niitä verrataan.

Sitten tulos tulostetaan edellä nimettyihin seuraavassa kuvassa näkyvällä tavalla ;-)

Kierrosten laskeminen eli ilmoittaminen kannattaa tehdä vasta sen jälkeen kun kaikki toimii.

```
tulos = vTaulut(tKentat(), pKentat());
document.getElementById("tulos").innerHTML=tulos;
document.getElementById("kierros").innerHTML=kierrokset;
```

33)

**Tehtävä 62.**

Lisätään ominaisuuksia **lueLuku()**-funktioon. Eli kun käyttäjä syöttää jonkin arvon, niin funktio tarkistaa onko syötetty arvo todella luku.

Mielestäni elegantin ratkaisu on rakentaa edellisestä funktiosta ns. totuusfunktio (*boolean*), joka palauttaa arvon **true**, jos syötetty arvo on luku ja **false**, jos se ei ole luku.

34) Ratkaisuehdotus:

35) Tehdään **onkoLuku()**-funktion, joka palauttaa true-arvo, jos annettu muuttuja on luku ja false-arvo jos se on joitain muuta kuin luku tai tyhjä arvo.

36) Periaatteessa tutkimuksen voidaan tehdä seuraavalla tavalla.

```
if (x=="" || isNaN(x))
```

Yhdeksästoista tunti

Pääohjelman selkiyttäminen

Tehdään pääohjelma, joka sisältää periaatteessa vain aliohjelmakutsuja (=funktioita). Tosi meidän tapauksessa tarvitaan myös taulukoita ja muuttujia, joihin funktioiden palauttamat arvot tallennetaan, mutta periaatteessa pääohjelma näyttää seuraavanlaiselta:

1. arvoJokeriNumerot()
2. paivitaJokeriKentat()
3. lueJokeriRivi()
4. tarkistaJokeri() eli vertaaTaulukot() -funktio
5. ilmoitaVoitot()

Yleensä funktiot (=aliohjelmat) kirjoitetaan pääohjelman yläpuolelle, paitsi ne, jotka ovat yleiskäyttöisiä.

Yleiskäyttöiset funktio, kuten esimerkiksi kahden arvon vertaaminen keskenään, voidaan kirjoittaa erilliseen tiedostoon, josta ne ovat käytettävissä useissa eri ohjelmointiprojekteissa.

```
/*  
* Funktio:         arvoJokeriNumerot()  
* Paluuarvo:      Taulukko, jossa on 7 lukua.  
* Tehtävä:        Täyttää taulukon 7 satunnaisluvulla 0-9  
* Tekijä:         Juha Korhonen - 5.4.2013  
*                Huhtaharjun koulu: 9ATV 2012 - 2013  
* Muokattu:  
*  
* Testiohjelma: tulostaTaulukko(arvoJokeriNumerot());  
*  
***/  
  
function arvoJokeriNumerot() {  
    var sLuvut = new Array();  
    for (var i=0; i<7; i++) {  
        sLuvut[i]=satunnaisLuku(0,9);  
    }  
    return sLuvut;  
}
```

Kirjoita myöhemmin seuraava satunnaisluvut päivittävätekstikenttiä lukeva funktio elegantimmin käyttäen for-silmukaa.

```
/******  
* Funktio:      paivitaJokeriKentat(taulukko)  
* Parametri:   Numerotaulukko  
* Paluarvo:    Ei ole  
* Tehtävä:     Kirjoittaa html-sivulle 7 nimettyyn kenttään parametrina  
*              välitetystä taulukosta 7 lukua.  
* Tekijä:      Juha Korhonen - 12.4.2013  
*              Huhtaharjun koulu: 9ATV 2012 - 2013  
* Muokattu:    30.4.2013 - muutetaan yleisempään muotoon  
*              - myöhemmin kentän nimen ja määrän voisi välittää  
*              funktion parametrina, jolloin sijoituslause  
*              joudutaan muuttamaan for-lauseeksi.  
*  
*****/  
function paivitaJokeriKentat(taulukko) {  
    document.getElementById("arvottu1").innerHTML=taulukko[0];  
    document.getElementById("arvottu2").innerHTML=taulukko[1];  
    document.getElementById("arvottu3").innerHTML=taulukko[2];  
    document.getElementById("arvottu4").innerHTML=taulukko[3];  
    document.getElementById("arvottu5").innerHTML=taulukko[4];  
    document.getElementById("arvottu6").innerHTML=taulukko[5];  
    document.getElementById("arvottu7").innerHTML=taulukko[6];  
}
```

Seuraavana on viimeksi tehty tekstikenttiä lukeva funktio, joka lukee antamamme jokeririvin taulukkoon. Kirjoita tämä funktio myöhemmin seuraava elegantimmin käyttäen for-silmukaa.

```
function lueJokeriRivi() {  
    // Luodaan arvatutLuvutTaulukko[]-taulukko.  
  
    var veikattuJokeririviTaulukko = new Array();  
  
    // Pyydystetään annettujen jokerinumeroiden arvot taulukkoon (yht. 7kpl).  
  
    veikattuJokeririviTaulukko[0]=document.getElementById("arvattu1").value;  
    veikattuJokeririviTaulukko[1]=document.getElementById("arvattu2").value;  
    veikattuJokeririviTaulukko[2]=document.getElementById("arvattu3").value;  
    veikattuJokeririviTaulukko[3]=document.getElementById("arvattu4").value;  
    veikattuJokeririviTaulukko[4]=document.getElementById("arvattu5").value;  
    veikattuJokeririviTaulukko[5]=document.getElementById("arvattu6").value;  
    veikattuJokeririviTaulukko[6]=document.getElementById("arvattu7").value;  
    return veikattuJokeririviTaulukko;  
}
```

Seuraavassa funktiossa, joka tarkistaa ovatko kahden taulukon vastaavat solut samoja, tarvittaisiin yksi aliohjelma lisää, joka vertaa kahden arvon erisuuruutta.

```
/******  
* Funktio:      tarkistaJokeri(taulu1, taulu2)  
* Parametrit:   Kaksi taulukkoa.  
* Paluarvo:     Yhtäsuurien lukujen määrä.  
* Tehtävä:      Vertaa taulukon vastaavien solujen  
*               yhtäsuuruutta.  
* Tekijä:       Juha Korhonen - 12.4.2013  
*               Huhtaharjun koulu: 9ATV 2012 - 2013  
* Muokattu:     30.4.2013 - muutetaan funktion nimi.  
*  
*****/  
function tarkistaJokeri(taulu1, taulu2) {  
  var osumat=0;  
  for (var i=0; i<taulu1.length; i++) {  
    if(taulu1[i]==taulu2[i]) // if(onkoSama(taulu1[i], taulu2[i]))  
      osumat = osumat + 1;  
    else  
      return osumat;  
  }  
  return osumat;  
}
```

Tämän funktion voi ja kannattaa lisätä *kokoelma.js* -tiedostoon myöhempää käyttöä varten. Sen nimi voisi kuitenkin olla yleiskäyttöisempi, esimerkiksi *vertaaTaulukot()*.

```
function onkoSama(luku1, luku2) {  
  if(luku1==luku2)  
    return true;  
  else  
    return false;  
}
```

Pääohjelmamme siis pitäisi näyttää suurin piirtein tältä:

```
<script>  
  
var jokeriRivi = new Array();  
var jokeriVeikkaus = new Array();  
var tulos=0;  
  
jokeriRivi = arvoJokeriNumerot();  
jokeriVeikkaus = lueJokeriRivi();  
tulos = tarkistaJokeri(jokeriVeikkaus, jokeriRivi);  
ilmoitaVoitot(tulos);  
  
</script>
```

Lisätään vielä nimetty kenttä tulosta varten eli html-sivun koodi olisi seuraavanlainen:

```
<p>  
Saatiin <font id="tulos">x</font> oikein tulos, kun arvontoja suoritettiin  
<font id="kierrokset">xx</font> kpl.  
</p>
```

Kentän päivitys voitiin tehdä seuraavalla koodin pätkällä. Tämän koodin voi sijoittaa esimerkiksi *ilmoitaVoitot()* -funktioon tai sitten aluksi suoraan *tarkistaJokeri()* -funktioon.

```
document.getElementById("tulos").innerHTML=tulos;  
// document.getElementById("kierrokset").innerHTML=;
```

Tällä hetkellä ei vielä mietitä miten saadaan selville kierrosten lukumäärä, vaan keskitytään pelkästään tuloksen esittämiseen.

Kun saat koodin toimimaan, voit testata sitä while-silmukalla siten, että arvontaa pyöritetään niin kauan, että saadaan haluttu tuos, esimerkiksi vähintään kolme oikein.

Seuraavassa esimerkissä on vain idea, se ei toimi noin, ainakaan tekemässämme ohjelmassa.

```
var tulos=0;  
while (tulos<3) {tulos=arvoJokeri();}
```

Lisätehtävää

- Tämän jälkeen voit keskittyä siihen, mistä saat selville sen kierroksen, jolla haluttu tulos saavutettiin.
- Lisää ohjelmaamme ominaisuus, joka kysyy kuinka monta kierrosta arvotaan.
- Lisää ohjelmaamme ominaisuus, joka kysyy halutun voittoluoka, niin voimme simuloida kuinka vaikea on saada esimerkiksi 5 oikein jokerissa.
- Jotta ohjelma olisi täydellinen, sen pitäisi ottaa talteen ja tulostaa kaikki ne rivit, joilla esiintyi voitto.

Kahdeskymmenes tunti

Tehtävä 63. Kopioi toimiva lähdekoodi eli `J:\JavaScript\07-tunti-jokeri.html` -tiedosto. Samasta kansioista löytyy myös `kokoelma.js` -tiedosto, tallenna se myös omaan `K:\Kotisivu\JavaScript\jokeri` -kansioosi.

Näin varmistetaan, että kaikilla on toimiva lähdekoodi. Kokeile siis aluksi koodin toimivuutta.

Ohjelmahan arpoo yhden jokeririvin ja vertaa sitä veikattuun riviin arvo kerrallaan.

Tehtävä 64. Lisää ohjelmaan ominaisuus, jolla voit kokeilla arpoa useamman kierroksen siten, että saavutetaan jokin tietty tulos, esim. neljä (4) oikein tulos.

Tehdään tämä kokeiluversio lisäämällä `while`-silmukka suoraan pääohjelmaan. Myöhemmin tehdään hienostuneempi versio, jolloin kysytään haluttujen arvontojen määrä.

Seuraava `while`-silmukka toimii niin kauan, kunnes suluissa oleva ehto täyttyy eli se tulostaa noin 10 riviä tekstiä.

```
while(i<10) {
    dokument.write("Laskurin arvo: " + i);
    i++;
}
```

Koeta keksi mikä ehto on tässä jokeriarvonnassa.

Tehtävä 65. Seuraavaksi lisätään jokeriohjelmaan ominaisuus, joka kertoo ja tulostaa näyttöön monennellako arvontakerralla saavutettiin haluttu tulos eli neljä oikein tässä esimerkissä.

Nimeä ensin alue, tekstikenttä, johon tulos sijoitetaan seuraavasti:

```
<p>
Saatiin <font id="tulos">x</font> oikein tulos, kun
arvontoja suoritettiin <font id="kierrokset">xx</font> kpl.
<p>
```

Tämän jälkeen kierrokset kentään saadaan tulostettua arvo seuraavalla lähdekoodilla:

```
document.getElementById("kierrokset").innerHTML=voittokierros;
```

Tehtävä 66. Lopuksi lisätään ohjelmaan kenttä, josta voi syöttää arvottavien kierrosten määrän ilman, että itse lähdekoodia tarvitsee joka kerta muuttaa.

Tehtävä 67. Hupia. Tee ns. puppugeneraattori eli kirjoita taulukkoon ”hauskoja” lauseita ja arvo satunnaisluvulla mikä lause tulostetaan sivua avattaessa. Esimerkiksi

seuraavaan tapaan.

```
tervehdys[0]="Hauskaa päivän jatkoa!";  
tervehdys[1]="Ei niin hauskaa päivän jatkoa!";  
tervehdys[2]="Tosi hauskaa päivän jatkoa!";
```

Tehtävä 68. Lottoarvonta. Satunnaisluku väliltä 1-39 ja arvotaan seitsemän numeroa. Tähän tarvitaan tietoa taulukoista ja lajittelusta tai ainakin ne tekevät tehtävän helpommaksi ☺

Mikä on ongelma lottonumeroissa verrattuna jokerinumeroihin?

Tehtävä 69. Tulosta lottorivi pienimmästä numerosta suurimpaan eli lajittele arvotut numerot.

Kahdeskymmenesyhdes tunti**Oliot**

Harjoitellaan tekemällä oma olio. Ensi kerralla käytetään JS:n valmiita Date()-luokan olioita.

Tehtävä 70. Kirjoita henkilö-luokka, jolla on seuraavat ominaisuudet: etunimi, sukunimi, syntymävuosi ja osoite.

Luo Matti-olio, jolla on kaikki henkilö-luokan ominaisuudet.

```
matti = new Henkilo(....., ....., ....., .....);
```

Tehtävä 71. Tulosta matti-luokan kaikki ominaisuudet. Käytä opettajan esittämää metodia eli funktiota siihen.

Kun olet testannut funktion toimivuuden voit siirtää sen *kokoelma.js* -tiedostoon.

```
function ominaisuudet(olio) {  
  var tulos = "";  
  for (var i in olio) {  
    // Miten tulos += voitaisiin kirjoittaa toisin  
    tulos += i + " = " + olio[i] + "<br>";  
  }  
  return tulos;  
}
```

Miten tulos += voitaisiin kirjoittaa toisin?

Tehtävä 72. Tee luokalle metodi, joka tulostaa henkilön etu- ja sukunimen siten, että etunimi on alleviivattu.

Käytä tulostamisen apuna *kokoelma.js* -tiedostossa olevaa **tulosta()**-funktiota

Tehtävä 73. Tee luokalle metodi, joka laskee henkilön iän synnyinvuodesta.

```
Ika = KULUVAVUOSI - synnyinvuosi;
```

Tehtävä 74. Lisää lähdekoodiin rivi, joka tulostaa henkilön iän.

Käytä tulostamisen apuna *kokoelma.js* -tiedostossa olevaa **tulosta()**-funktiota

Tehtävä 75. Tee osoitteesta oma luokka, jolla on ominaisuudet: lähiosoite, postinumero ja postitoimipaikka.

```
matin_osoite = new Osoite(....., ....., .....);
```

Luo matti-olio uudestaan käyttäen luomaasi luokkaa seuraavasti:

```
matti = new Henkilo(....., ..... , matin_osoite);
```

Kahdeskymmeneskahdes tunti

JavaScriptin oma Date()-olio

Tällä tunnilla harjoitellaan valmiin JavaScript-olion käyttöä. Math()-oliota käytimme hiukan satunnaislukufunktiiossamme ja nyt testaamme Date()-oliota, johon on koottu päivämäärään liittyviä metodeja.

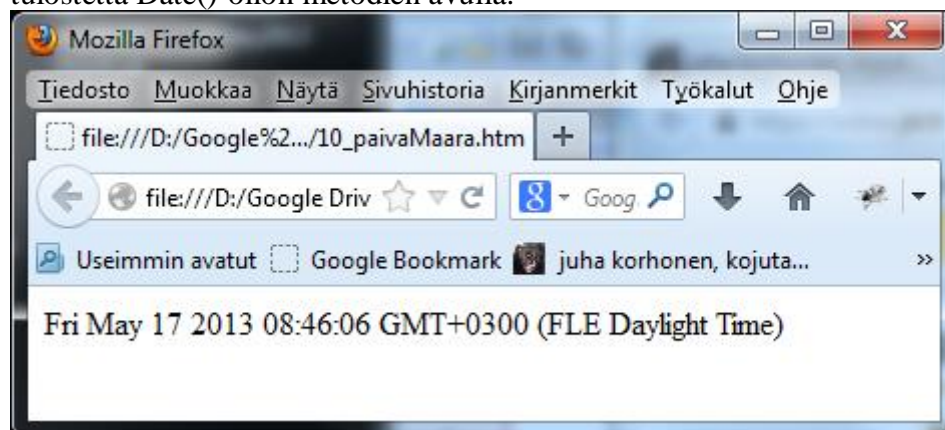
Seuraavassa on lueteltu yleisimmät Date()-olion metodit:

- Date.getDate() //palauttaa kuukauden päivän (1-31)
- Date.getDay() // palauttaa viikon päivän (0=sunnuntai)
- Date.getHours() // palauttaa tunnit(0-23)
- Date.getMinutes() // palauttaa minuutit (0-59)
- Date.getMonth() // palauttaa kuukaudet (0-11; 0 = tammikuu)
- Date.getSeconds() // palauttaa sekunnit (0-59)
- Date.getTime() // palauttaa ajanvälin Epoch aikanykyisyys (ms:na)
- Date.getYear() // palauttaa vuoden????
- Date.getFullYear() // palauttaa vuoden

Tehtävä 76. Lisätään html-sivulle päivämäärä. Tallenna sivu K:\Kotisivu\JavaScript-kansioon **kalenteri.html** -tiedostonimellä.

```
/* Luodaan Tanaan-päivämääräolio. */  
var Tanaan = new Date();  
  
/* Seuraavalle riville lause, joka tulostaa päivämäärän */
```

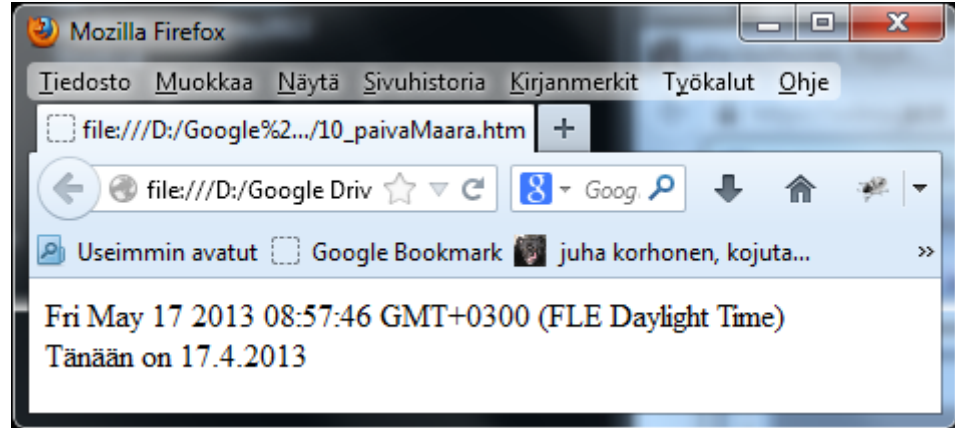
Tuloste näyttää hiukan liiankin tarkalta? Seuraavassa tehtävässä siistitään tulostetta Date()-olion metodien avulla.



Tehtävä 77. Pyydystetään seuraavassa kuukauden päivä, kuukauden numero ja vuosi ja esitetään ne muodossa totutussa 17.5.2013.

```
/* Haetaan meneillään oleva vuosi muuttujaan vuosi. */  
var Tanaan = new Date();  
var vuosi = Tanaan.getFullYear()
```

Nyt voit tulostaa vuosi-muuttujan aivan kuten minkä tahansa muuttujan. Hae vielä meneillään olevan päivän ja kuukauden numero ja tulosta ne muodossa.



Tehtävä 78. Lisätään ohjelmaan ominaisuus, joka tulostaa sivulle, mikä viikonpäivä on menossa.

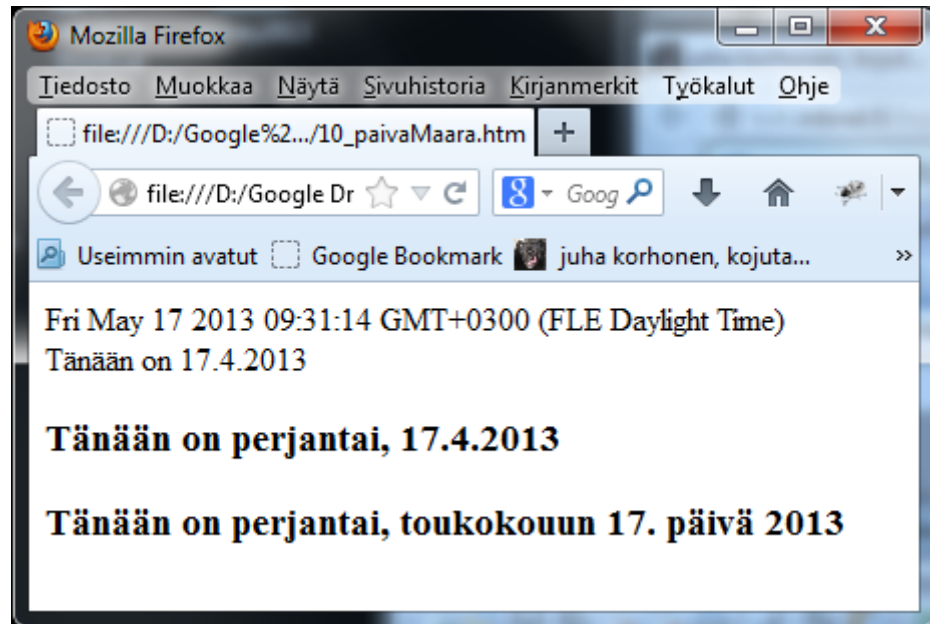
Date()-olio ei (*tietenkään*) suoraan anna viikonpäivän nimeä, eikä varsinkaan suomeksi, joten joudumme rakentamaan funktion, joka muuttaa Date.getDay()-funktion antaman numeron viikonpäiväksi.

Tee esimerkiksi seuraava funktio, joka palauttaa päivän nimen paluuarvossaan

```
function palautaPaiva(paivanNumero) {  
    if(paivanNumero === 5) return "perjantai";  
    .  
    .  
    .  
    .  
    .  
}
```

Huom! Päivän numeron saat selville Date()-metodilla getDay().

Tehtävä 79. Lisää ohjelmaasi samaan tapaan myös meneillään olevan kuukauden tulostus.



Tehtävä 80. Lisätään sivulle copyright-merkintä (c) 2011-2013 , jonka loppupäiväystä (tai vuotta) ei tarvitse muuttaa, vaan se muuttuu automaattisesti.

Tehtävä 81. Tehtävänä on selvittää minä viikonpäivä henkilö (1.1.1996) on syntynyt. Date()-oliolle päivämäärän voi antaa myös seuraavasti.

```
var syntymaPaiva = new Date(1996,0,1);
```

Edellä olevan määrittelyn jälkeen syntymaPaiva()-oliolta voidaan kysyä mikä päivä on menossa.

Tehtävä 82. Lisää ohjelmaan ominaisuus, joka kysyy päivämäärään. Käytä esimerkiksi aiemmin luomaamme lueLuku()-funktiota.

```
function lueLuku() {  
  var input=0;  
  input = prompt("Syötä luku");  
  return parseInt(input);  
}
```

Lopuksi tulostetaan sivulle viikonpäivän jona henkilö on syntynyt.

Tehtävä 83. Lasketaan monesko vuoden päivä on menossa.

Vihje: luo kuukaudet[]-taulukko, jossa on kaikkien kuukausien päivien lukumäärät.

```
kuukaudet[31, 28, 31, ..., 31];
```

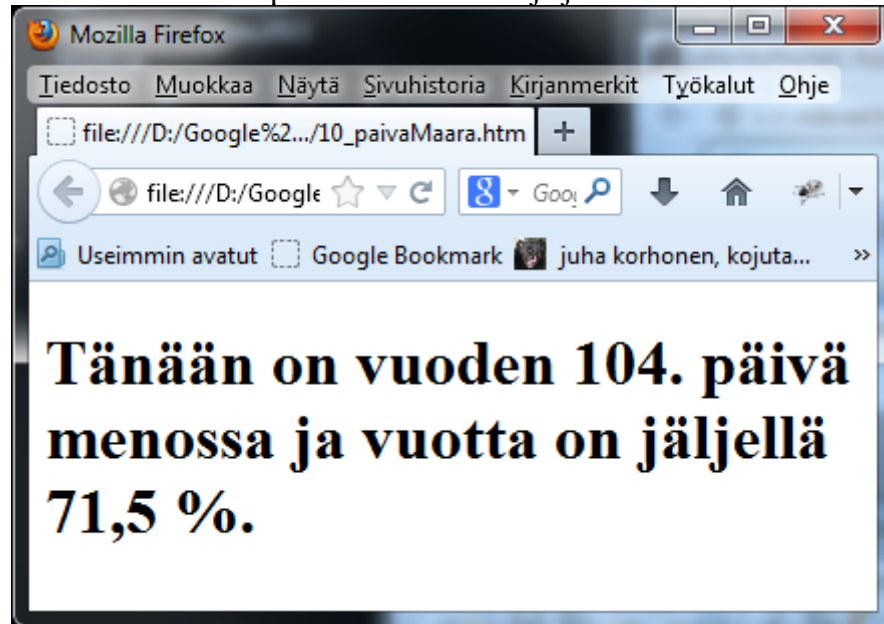
Tämän jälkeen teet funktion, joka laskee vuoden alusta kaikki päivät yhteen, aina menossa olevaan kuukauteen saakka.

Tulosta päivien lukumäärä sivulle.

Tarkista ohjelmasi toimivuus laskemalla koko vuoden päivämäärät. Kokeile vuosia 2012 (366 päivää) ja 2013 (365 päivää).

Tehtävä 84.

Lasketaan montako prosenttia vuotta on jäljellä.

**Tehtävä 85.**

Tee funktio, joka pyöristää tuloksen esimerkiksi yhteen tai kahteen desimaaliin.