**JavaScript tehtäviä**

**Show an alert**

Näytä hälytys

Luo sivu, jossa näkyy viesti “Olen JavaScript!”.

Tee se hiekkalaatikossa tai kiintolevylläsi, sillä ei ole merkitystä, varmista vain, että se toimii.

**Näytä alert omassa scriptissä**

Valitse ratkaisu edelliseen tehtävään Näytä hälytys. Muokkaa sitä muuttamalla skriptin sisältö ulkoiseen tiedostoon Alert.js. Katso että html-sivu ja alert.js on samassa kansiossa..

Avaa sivu ja varmista, että alert toimii.

**Working with variables**

Työskentely muuttujien kanssa 1

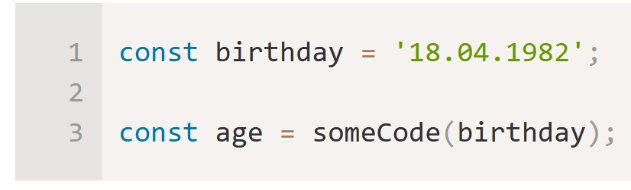
Tee kaksi muuttujaa: admin ja nimi.  
 Anna nimelle arvoksi "John".  
 Kopioi arvo nimestä adminiin.  
 Näytä adminin arvo alertia käyttämällä (on tulostettava ”John”).

**Giving the right name**

Luo muuttuja, jolla on planeettamme nimi. Kuinka nimesit muuttujan?  
Luo muuttuja tallentaaksesi nykyisen verkkosivuston vierailijan nimen. Kuinka nimesit muuttujan?

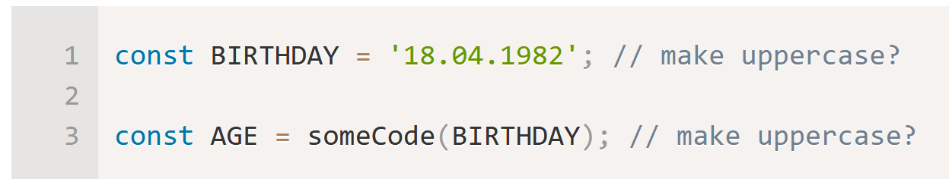
**Iso kirjain vakiona**

Tutki seuraavaa koodia:



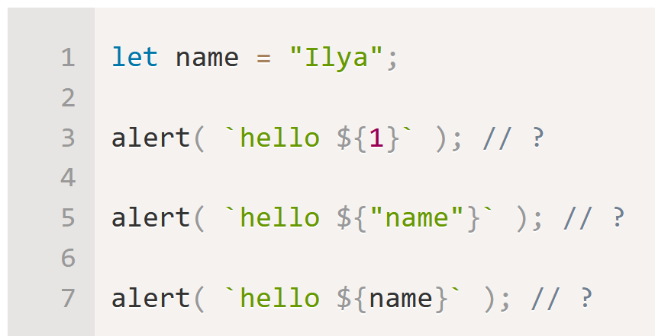
Vakio syntymäpäiväpäivä ja vakio ikä lasketaan syntymäpäivästä jonkin koodin avulla (sillä ei ole merkitystä tässä).

Onko oikein käyttää isoja kirjaimia Birthday muuttujassa? Tai age? Tai molemmissa? Esim. näin?



**Merkkijonot**

Mikä on scriptin tulos? Testaa, jos et tiedä. Laita koodi esim. JsFiddleen

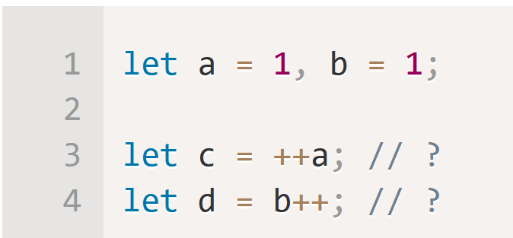


**A simple page**

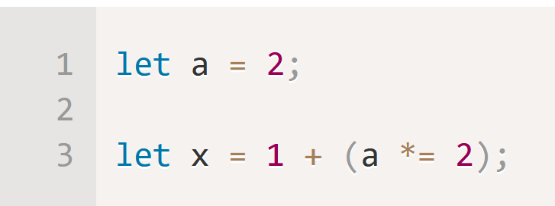
Luo verkkosivu, joka pyytää käyttäjän nimeä ja tulostaa sen.

**The postfix and prefix forms**

Mitkä ovat kaikkien muuttujien a, b, c ja d lopulliset arvot alla olevan koodin jälkeen?



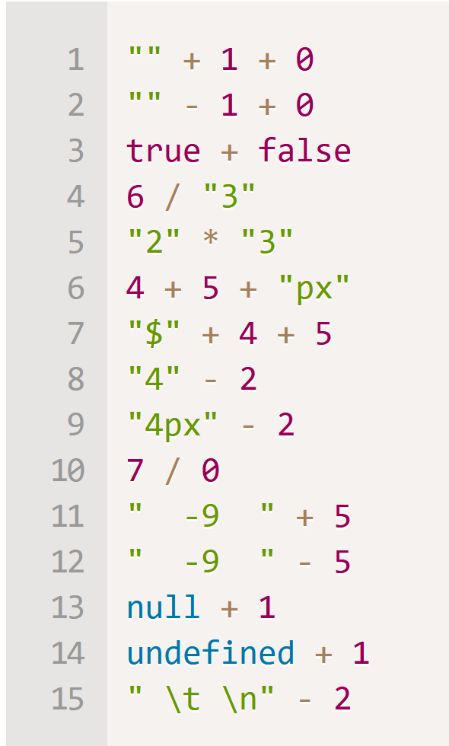
Mitkä ovat a ja x arvot alla olevan koodin jälkeen?



**Lausekkeiden tulokset**

Mitkä ovat tulokset näistä lausekkeista?

Mieti, kirjoita muistiin ja vertaa sitten open vastaukseen.

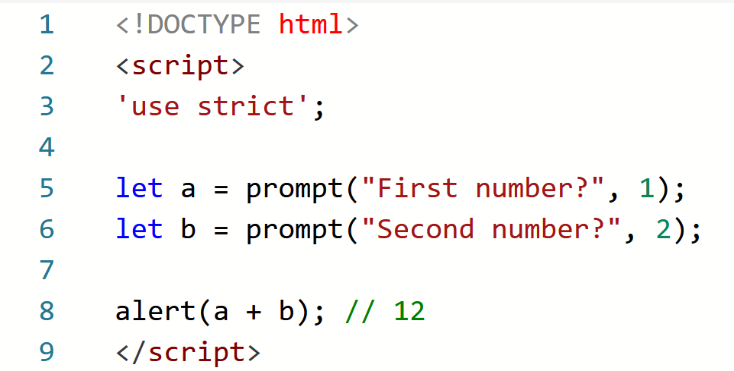


**Korjaa lisäys**

Tässä on koodi, joka kysyy käyttäjältä kahta numeroa ja näyttää niiden summan.

Se toimii väärin. Alla olevan esimerkin vastaus on 12 (oletuskehotusarvoille).

Miksi? Korjaa se. Tuloksen tulisi olla 3.



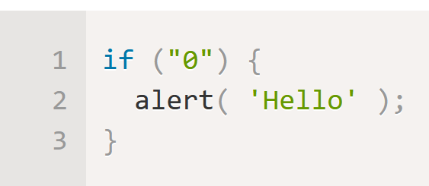
**Vertailut**

Mikä on tulos näille?



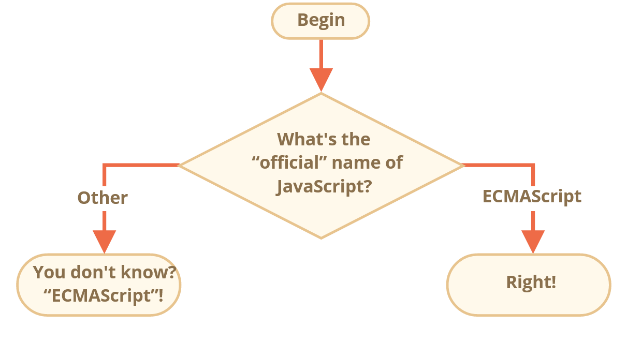
**if (merkkijono nolla)**

Näytetäänkö alert?



**JavaScriptin nimi**

Kirjoita if..else -konstruktilla koodi, joka kysyy: 'Mikä on JavaScriptin' virallinen 'nimi?'



**Näytä kyltti**

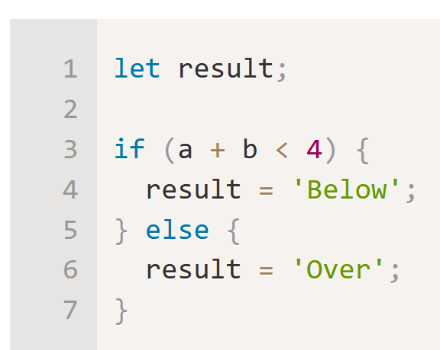
Kirjoita if..else -komennolla koodi, joka saa numeron nopeasti ja näkyy alertissa:

1, jos arvo on suurempi kuin nolla,  
 -1, jos vähemmän kuin nolla,  
 0, jos yhtä suuri kuin nolla.

Tässä tehtävässä oletamme, että syöte on aina numero.

**Korvaa 'jos' osaksi '?'**

Kirjoita tämä uudelleen, jossa käytät ehdollista operaattoria '?':



**Kirjoita "if..else" lauseet uudelleen "?"**

Luettavuuden vuoksi on suositeltavaa jakaa koodi useisiin riveihin.



**Jos ehto**

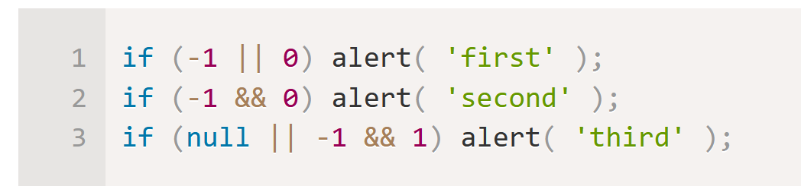
Kirjoita ”jos” -ehto tarkistaaksesi, että ikä on välillä 14–90.

Reunaehdot on 14 tai 90.

**Kysymys "jos"**

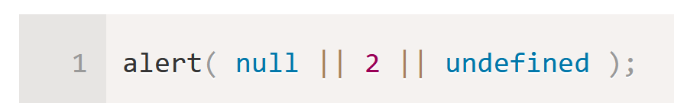
Mitkä näistä alerteista aiotaan suorittaa?

Mitkä ovat lausekkeiden tulokset, jos (...)?



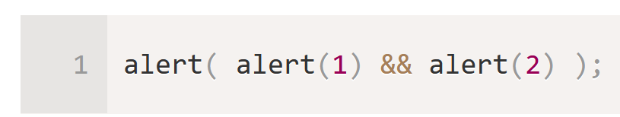
**AND, OR jne**

Mitä tai koodissa tapahtuu?



**What is the result of AND?**

Mitä tämä koodi tekee?



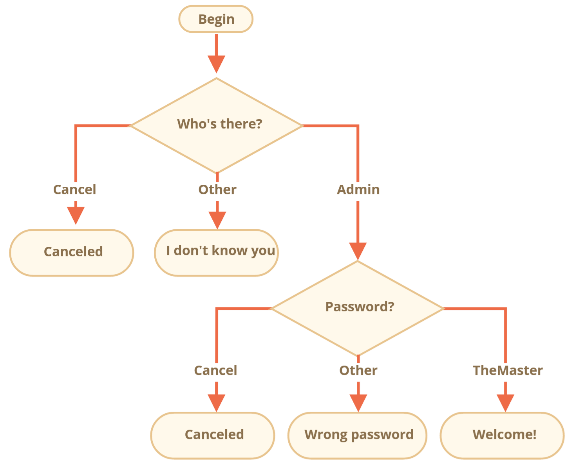
**Tarkista kirjautuminen**

Kirjoita koodi, joka pyytää kirjautumista, ja kehote.

Jos vierailija kirjoittaa "Järjestelmänvalvoja", pyydä salasanaa, jos syöte on tyhjä rivi tai Esc - näytä "Peruutettu", jos kyseessä on toinen merkkijono - teksti sitten "En tiedä sinua".

**Salasana tarkistetaan seuraavasti:**

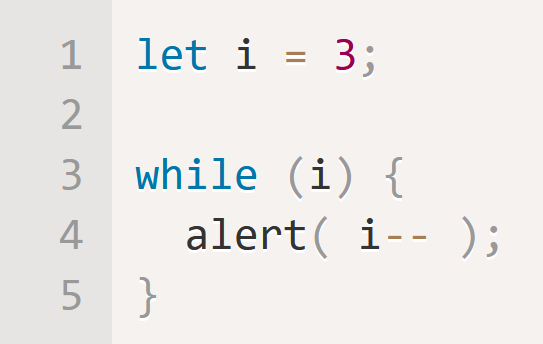
Jos se on sama kuin TheMaster, näytä sitten "Welcome!",  
 Toinen merkkijono - näytä “Väärä salasana”,  
 Jos tyhjä merkkijono tai peruutettu syöte, näytä “Peruutettu”



Käytä sisäkkäisiä, jos lohkoja. Huomaa koodin yleinen luettavuus.

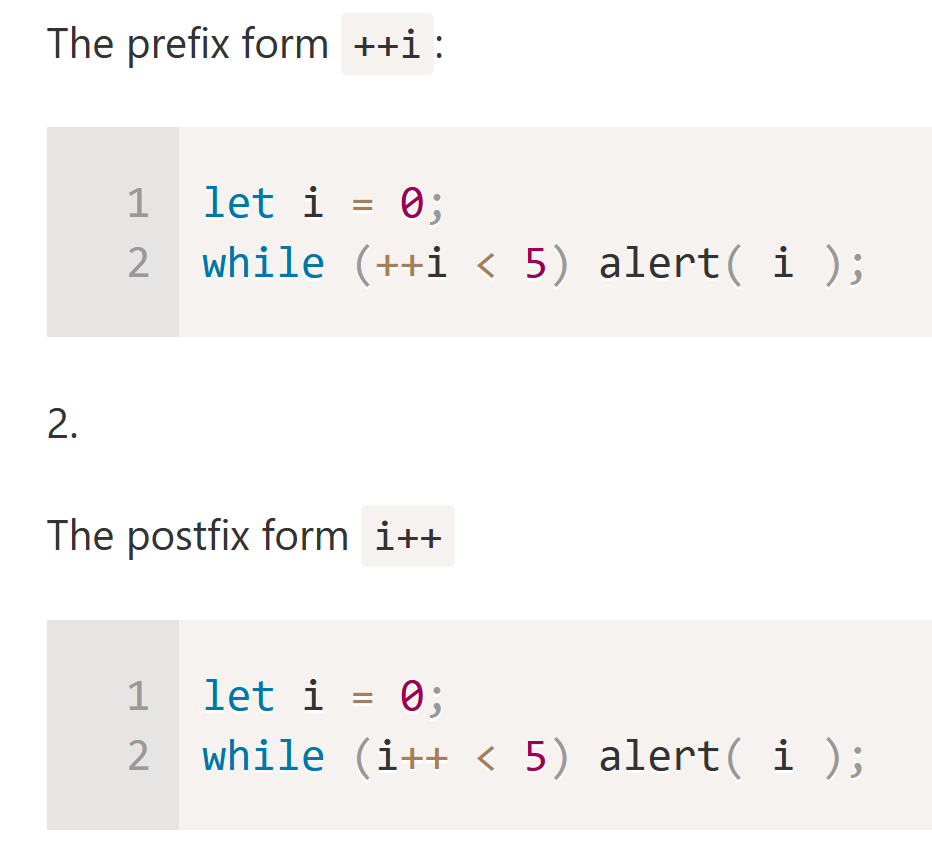
Vihje: tyhjän syötteen lähettäminen kehoteelle tuottaa tyhjän merkkijonon ''. Painamalla ESC-painiketta kehotuksen aikana, palauttaa arvon 0.

**Last loop value**



### Which values does the while loop show?

Mitkä arvot while-silmukka näyttää? tärkeys: 4 Kirjoita jokaiselle silmukka-iteraatiolle arvo, jonka se tuottaa, ja vertaa sitä sitten ratkaisuun. Molemmat silmukat tuovat alertiin samoja arvoja vai eivätkö?



**For**

Mitkä arvot näkyvät "for" -silmukalla? tärkeys: 4 Kirjoita jokaiselle silmukalle, mitkä arvot se näyttää. Vertaa sitten vastaukseen. Molemmat silmukat laittavat alertiin samoja arvoja vai eivätkö?



**Syötä parilliset numerot silmukassa**

Käytä silmukkaa for for parillisten numeroiden antamiseen 2-10.

**Toista, kunnes tulo on oikea.**

Kirjoita silmukka, joka pyytää yli 100: ta lukua. Jos vierailija syöttää toisen numeron, pyydä häntä syöttämään uudelleen. Silmukan on kysyttävä numeroa, kunnes joko kävijä syöttää yli 100-luvun tai peruu syötteen / syöttää tyhjän rivin. Tässä voidaan olettaa, että kävijä syöttää vain numeroita. Tässä tehtävässä ei tarvitse käyttää erityistä käsittelyä ei-numeeriselle syötteelle.

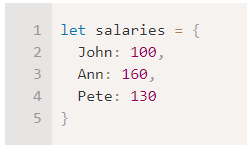
**Objektit eli oliot**

Kirjoita koodi, yksi rivi kullekin toiminnolle:

Luo tyhjä olion käyttäjä.  
Lisää ominaisuuden nimi arvoksi John.  
Lisää ominaisuuden sukunimi arvoksi Smith.  
Muuta nimen arvoksi Pete.  
Poista ominaisuus nimi oliosta.

**Tarkista isEmpty**  
  
Kirjoita funktio isEmpty (obj), joka palauttaa arvon true, jos objektilla ei ole ominaisuuksia, muuten epätosi.

Pitäisi toimia näin:



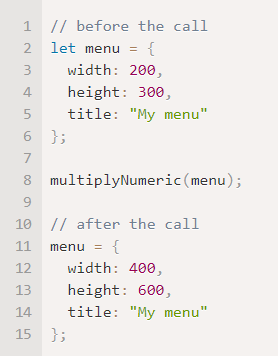
Kirjoita koodi summaamaan kaikki palkat ja tallenna muuttuja summaan.

Pitäisi olla 390 yllä olevassa esimerkissä (kun lasket kaikki palkat yhteen).

Jos palkat ovat tyhjät, tuloksen on oltava 0.

**Multiply numeric property values by 2**

Luo funktio multiplyNumeric (obj), joka kertoo obj: n kaikki numeeriset ominaisuusarvot 2: lla.  
Esimerkiksi:



Huomaa, että multiplyNumericin ei tarvitse palauttaa mitään.

P.S. Käytä typeof:ia tarkistaaksesi numeron.

**Object method this**

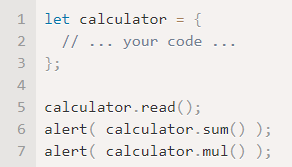
**Using "this" in object literal**

Tässä funktio makeUser palauttaa objektin.  
Mikä on seurausta sen viitteestä? Miksi?



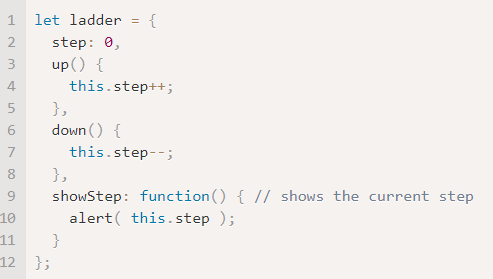
**Luo laskin**  
  
Luo objektilaskin kolmella tavalla:

read () pyytää kahta arvoa ja tallentaa ne kohteen ominaisuuksiksi.  
summa () palauttaa tallennettujen arvojen summan.  
mul () kertoo tallennetut arvot ja palauttaa tuloksen.

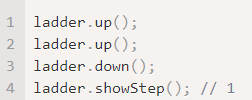


**Ketjutus**

Tikkaalla on esine, jonka avulla voi mennä ylös ja alas:



Jos meidän on nyt soitettava useita puheluja peräkkäin, voimme tehdä sen näin:



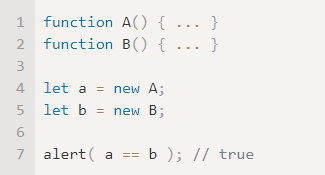
Muokkaa up-, down- ja showStep-koodia, jotta puhelut voidaan ketjuttaa seuraavasti:



Sellaista lähestymistapaa käytetään laajasti kaikissa JavaScript-kirjastoissa.

**Constructor**

Kaksi toimintoa - yksi objekti  
  
Onko mahdollista luoda funktioita A ja B, kuten new A () == new B ()?

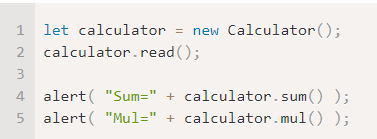


Jos on, kerro sitten esimerkki.

**Luo uusi laskin**  
  
Luo konstruktorifunktio Laskin, joka luo objekteja 3 menetelmällä:

read () pyytää kahta arvoa kehotteen avulla ja muistaa ne kohteen ominaisuuksissa.  
summa () palauttaa näiden ominaisuuksien summan.  
mul () palauttaa näiden ominaisuuksien kertolaskun.

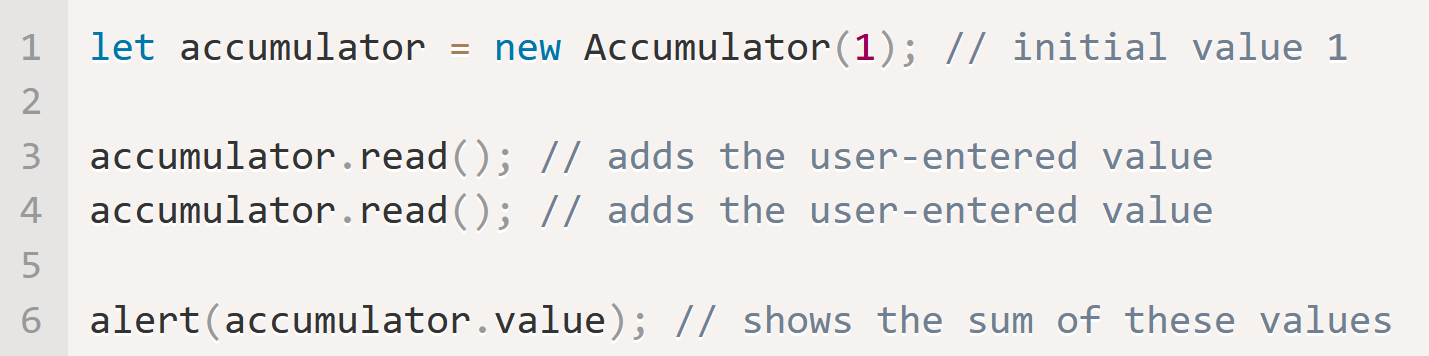
Esimerkiksi:



### Create new Accumulator

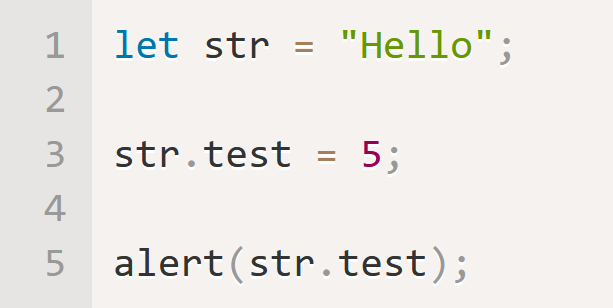
Luo konstruktoritoiminto Accumulator (StartValue)

objektin tulisi: Tallentaa nykyinen arvo ominaisuuden arvoon. Aloitusarvo asetetaan konstruktorin startValue argumentille. Read () -menetelmän tulisi käyttää kehotetta uuden luvun lukemiseen ja arvon lisäämiseen. Toisin sanoen arvo-ominaisuus on kaikkien käyttäjän syöttämien arvojen summa alkuperäisen arvon startValue kanssa.



## **Can I add a string property?**

Kirjoita koodi:



Mitä luulet toimiiko koodi? Mitä siinä näytetään?

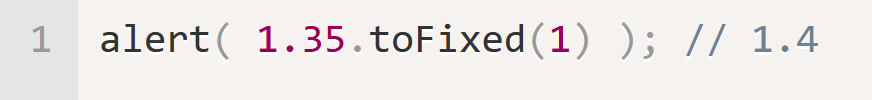
Number

### Sum numbers from the visitor

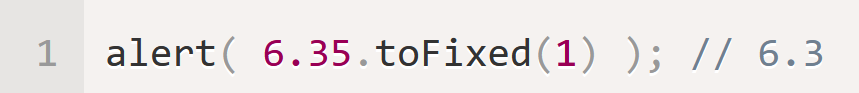
Luo scripti, joka kehottaa käyttäjää syöttämään kaksi numeroa ja näyttää sitten niiden summan.

## **Why 6.35.toFixed(1) == 6.3?**

Math.round ja toFixed pyöristetään lähimpään numeroon.



Miksi 6.35 on pyöristetty 6.3 eikä 6.4:ään



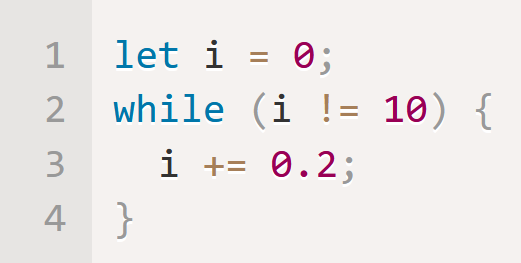
Miten pyöristää 6.35 oikealla tavalla?

### Repeat until the input is a number

### Luo funktio readNumber, joka pyytää lukua, kunnes käyttäjä antaa kelvollisen numeerisen arvon. Tuloksena oleva arvo on palautettava numerona. Käyttäjä voi myös pysäyttää prosessin syöttämällä tyhjän rivin tai painamalla "PERUUTA". Tällöin funktion pitäisi palauttaa nolla.

**Satunnainen ääretön silmukka**

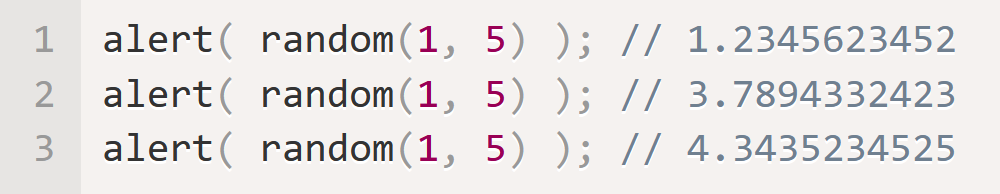
Tämä silmukka on ääretön. Se ei lopu koskaan. Miksi?



**Satunnainen luku min-max**

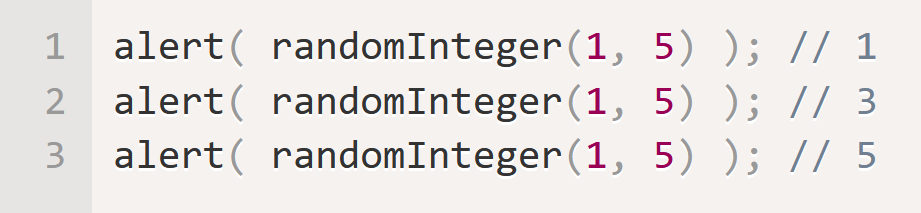
Sisäänrakennettu funktio Math.random () luo satunnaisarvon 0: sta 1: een (ei 1: tä). Kirjoita funktio satunnainen (min, max) satunnaisen liukuluvun muodostamiseksi minuutista maksimiin (ilman max: ää).

Esim. sen toiminnasta.



**Satunnainen kokonaisluku välillä min - max**

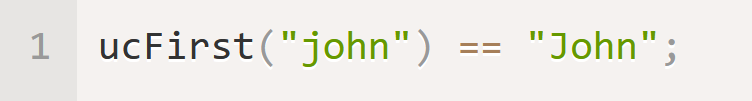
Luo funktio randomInteger (min, max), joka tuottaa satunnaisen kokonaisluvun välillä min - max, sisältäen sekä min- että max-arvot. Kaikkien numeroiden välillä min..max on oltava sama todennäköisyys.



String

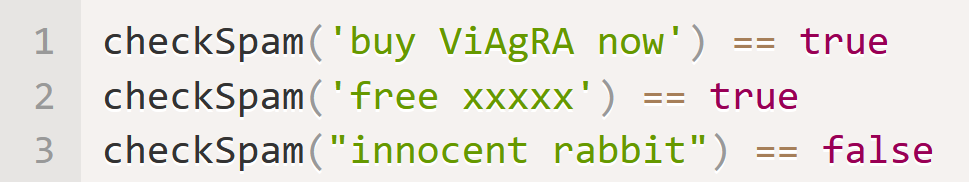
**Eka merkki isoilla kirjaimilla**

Kirjoita funktio ucFirst (str), joka palauttaa merkkijonon ison ensimmäisen merkin, esimerkiksi:



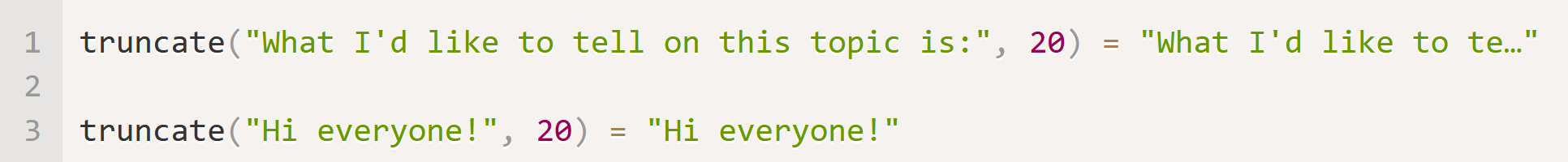
**Tarkista roskaposti**

Kirjoita funktio checkSpam (str), joka palauttaa arvon true, jos str sisältää 'viagra' tai 'XXX', muuten false. Toiminnon on oltava kirjainkoosta riippumaton

:

**Katkaise teksti**

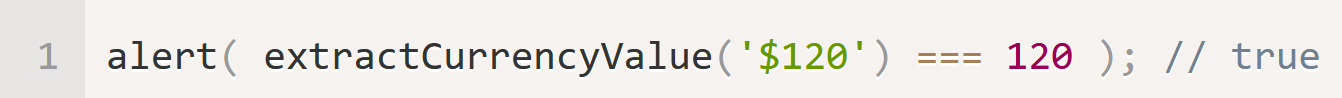
Luo funktion katkaisu (str, maxlength), joka tarkistaa merkkijonon pituuden ja, jos se ylittää maksimipituuden - korvaa str: n pään ellipsin merkillä "…", jotta sen pituus olisi yhtä suuri kuin maksimipituus. Funktion tuloksen tulisi olla katkaistu (tarvittaessa) merkkijono.



**Pura rahat**

Amerikassa on kustannukset muodossa "$ 120". Eli: dollarin merkki tulee ensin ja sitten numero. Luo funktio extractCurrencyValue (str), joka poimii numeerisen arvon tällaisesta merkkijonosta ja palauttaa sen.

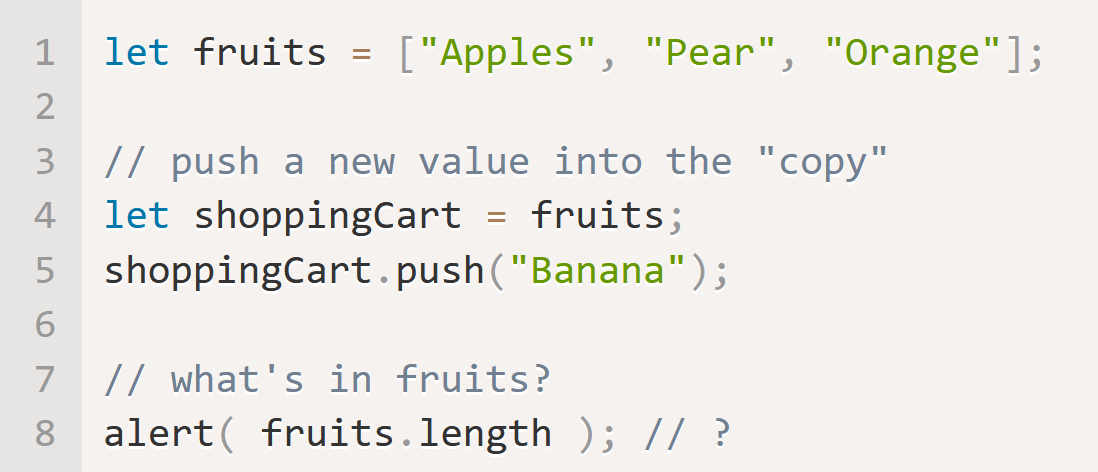
esim.



Arrays, loops

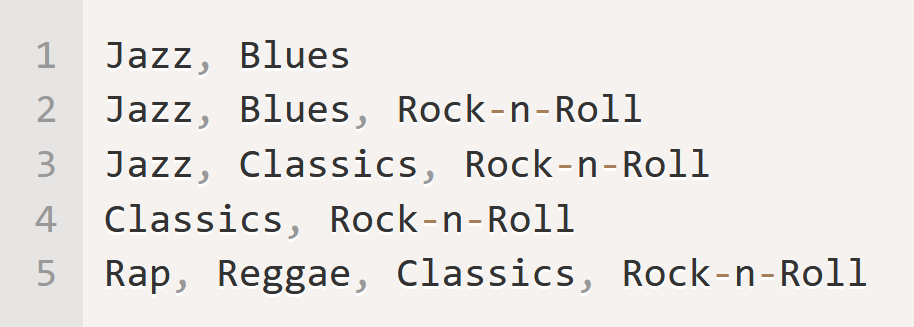
## **Is array copied?**

Mitä koodi tekee?



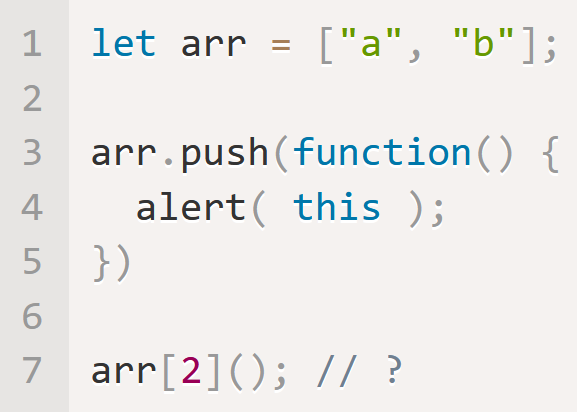
Kokeillaan viittä taulukko toimintoa. Luo taulukotyylit kohteilla "Jazz" ja "Blues". Liitä ”Rock-n-Roll” loppuun. Korvaa keskellä oleva arvo arvolla “Classics”. Ota Rap ja Reggae mukaan ryhmään.

Tee kuvan mukainen muutos….



### Calling in an array context

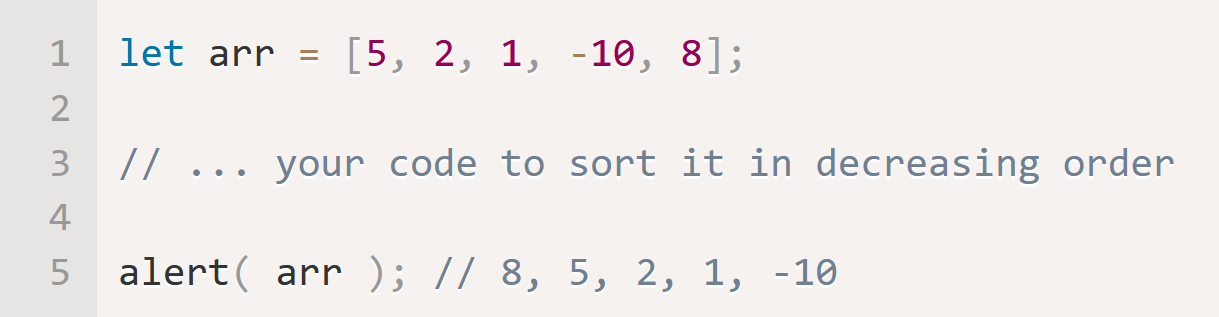
Mikä on tulos ja miksi?



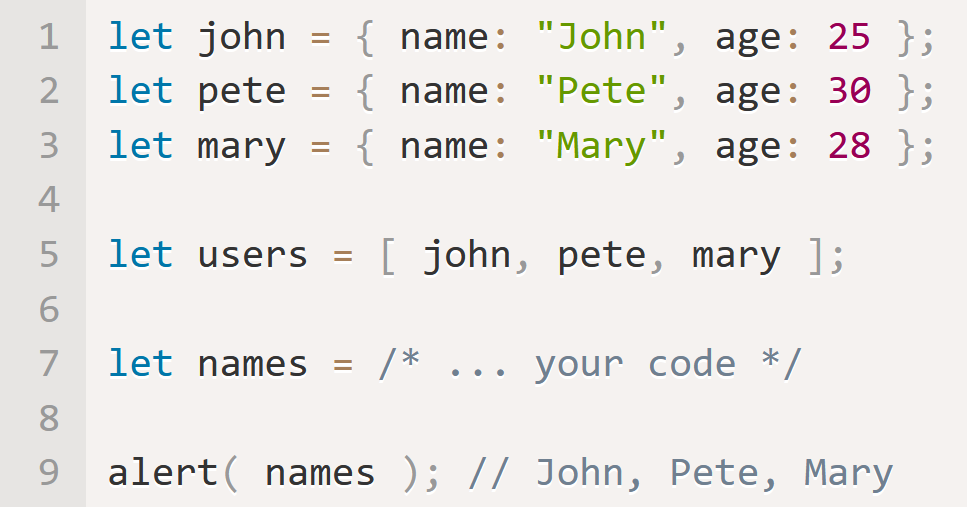
**Syötettyjen numeroiden summa**

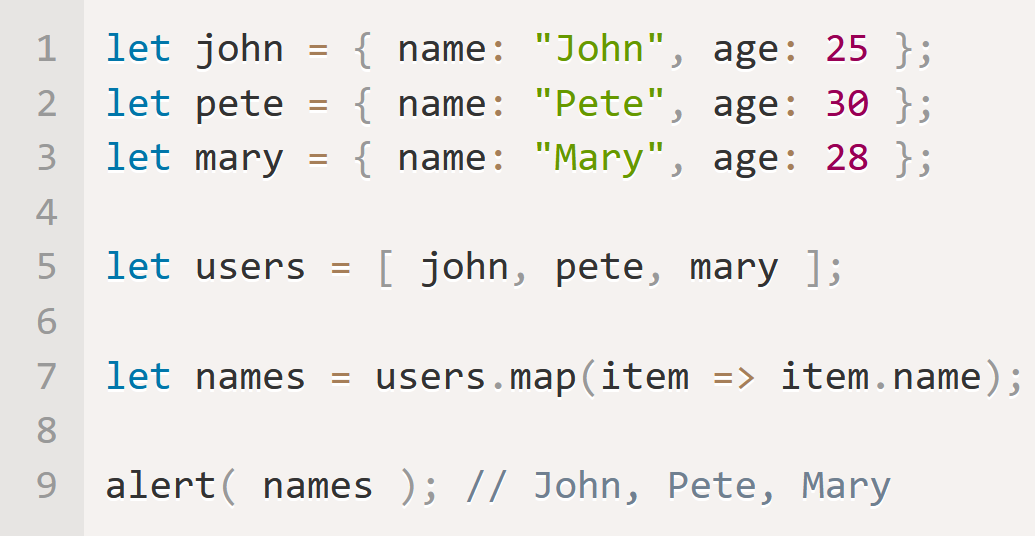
Kirjoita funktio sumInput (), joka: Kysyy käyttäjältä arvoja kehotteen avulla ja tallentaa arvot taulukkoon. Lopettaa kysymyksen, kun käyttäjä syöttää ei-numeerisen arvon, tyhjän merkkijonon tai painaa “Peruuta”. Laskee ja palauttaa array itemien summan. P.S. Nolla 0 on kelvollinen luku, älä lopeta syötettä nollalla.

**Lajittele laskevassa järjestyksessä**



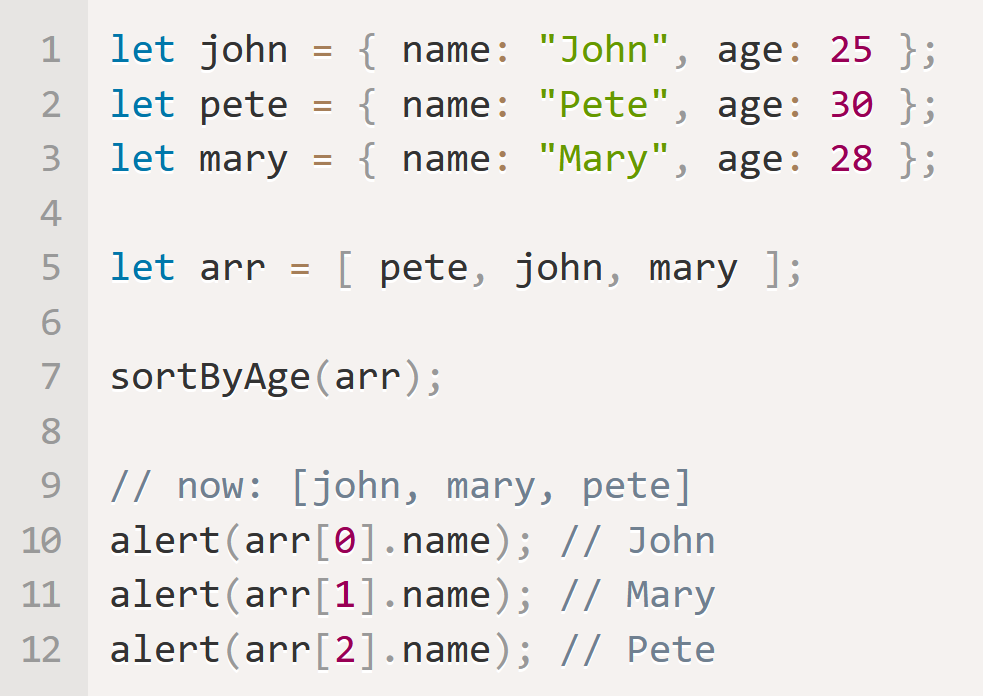
Mitä eroa?





**Lajittele käyttäjät iän mukaan**

Kirjoita funktio sortByAge (users), joka saa joukon objekteja ikäominaisuudella ja lajittelee ne iän mukaan.

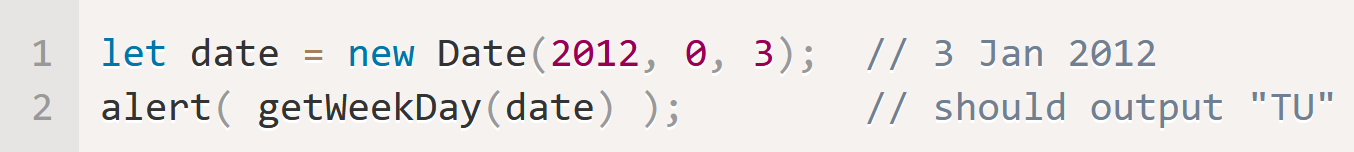


Date and time

**Luo päivämäärä**

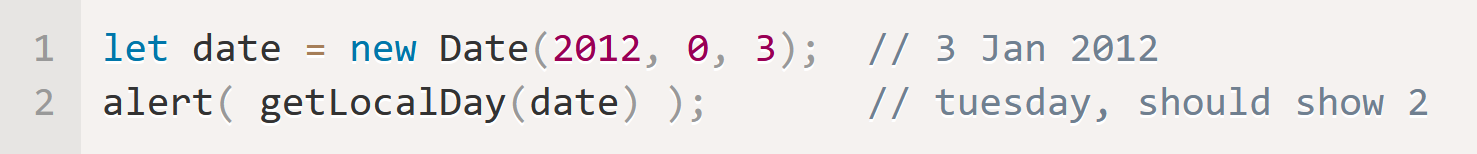
Luo päivämääräobjekti päivämäärälle: 20. helmikuuta 2012, 3.12. Aikavyöhyke on paikallinen. Näytä se alertin avulla.

**Näytä viikonpäivä**

Kirjoita funktio getWeekDay (päivämäärä) näyttämään viikonpäivä lyhyessä muodossa: 'MO', 'TU', 'WE', 'TH', 'FR', 'SA', 'SU' tai suomeksi…

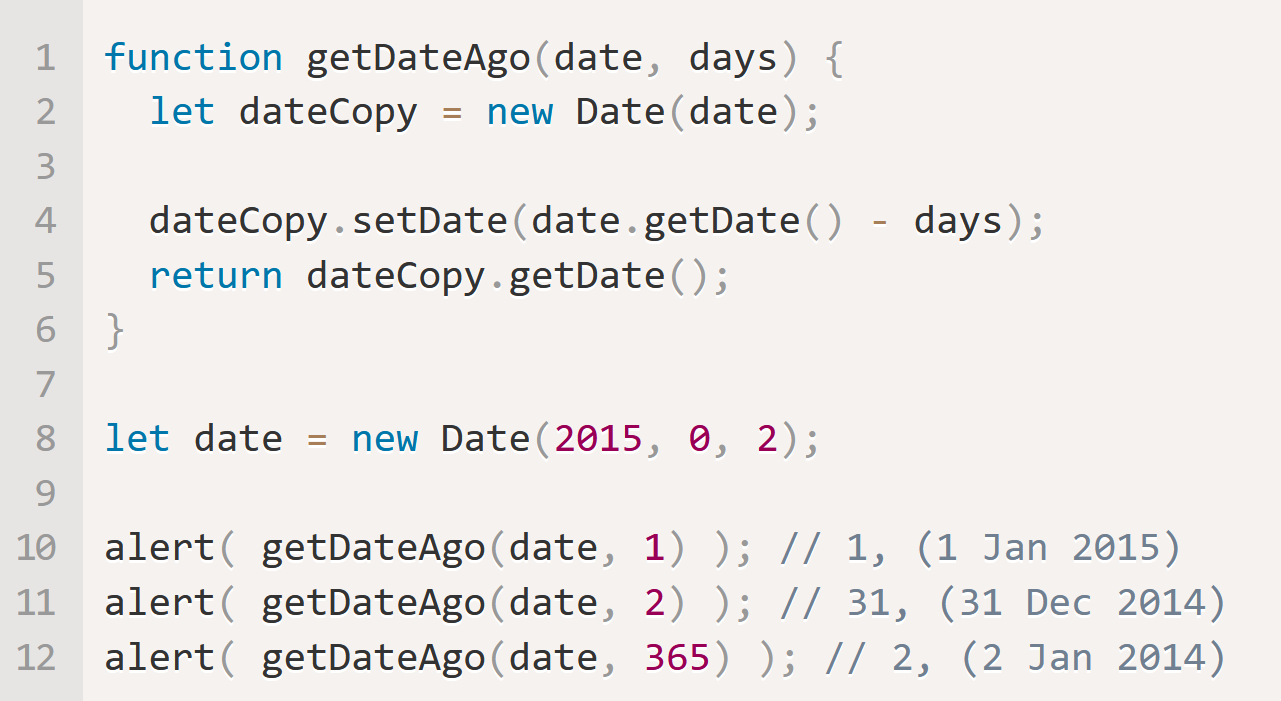
**Euroopan arkipäivä**

Euroopan maissa viikonpäivät alkavat maanantaista (numero 1), sitten tiistaina (numero 2) ja sunnuntaihin (numero 7). Kirjoita funktio getLocalDay (päivämäärä), joka palauttaa päivämäärän viikonpäivän nimen.



**Mitä koodissa tapahtuu?**

Toteuta koodi.



**Kuukauden viimeinen päivä?**

Kirjoita funktio getLastDayOfMonth (vuosi, kuukausi), joka palauttaa kuukauden viimeisen päivän. Joskus on helmikuun 30., 31. tai jopa 28./29.

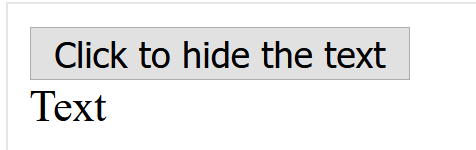
Parametrit: vuosi - nelinumeroinen vuosi, esimerkiksi 2012. kuukausi - kuukausi, 0-11.

Esimerkiksi getLastDayOfMonth (2012, 1) = 29 (karkausvuosi, helmikuu).

Events, tapahtumat

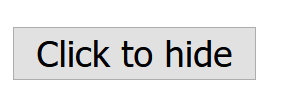
**Piilota napsautus**

Lisää JavaScript painikkeeseen, jotta <div id = "text"> katoaa, kun napsautat sitä.



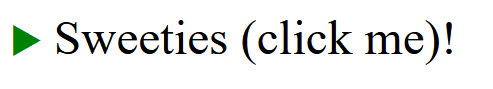
**Piilota itsesi**

Luo painike, joka piilottaa itsensä napsautuksella.

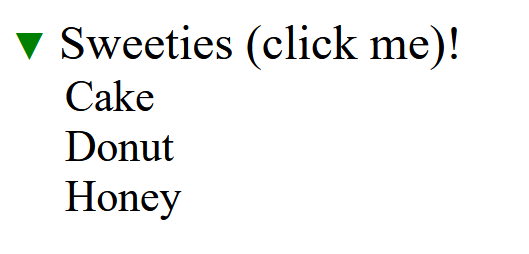
vaikka näin: 

## **Create a sliding menu**

Luo valikko, joka avautuu ja sulkeutuu, kun napsautat kolmiota



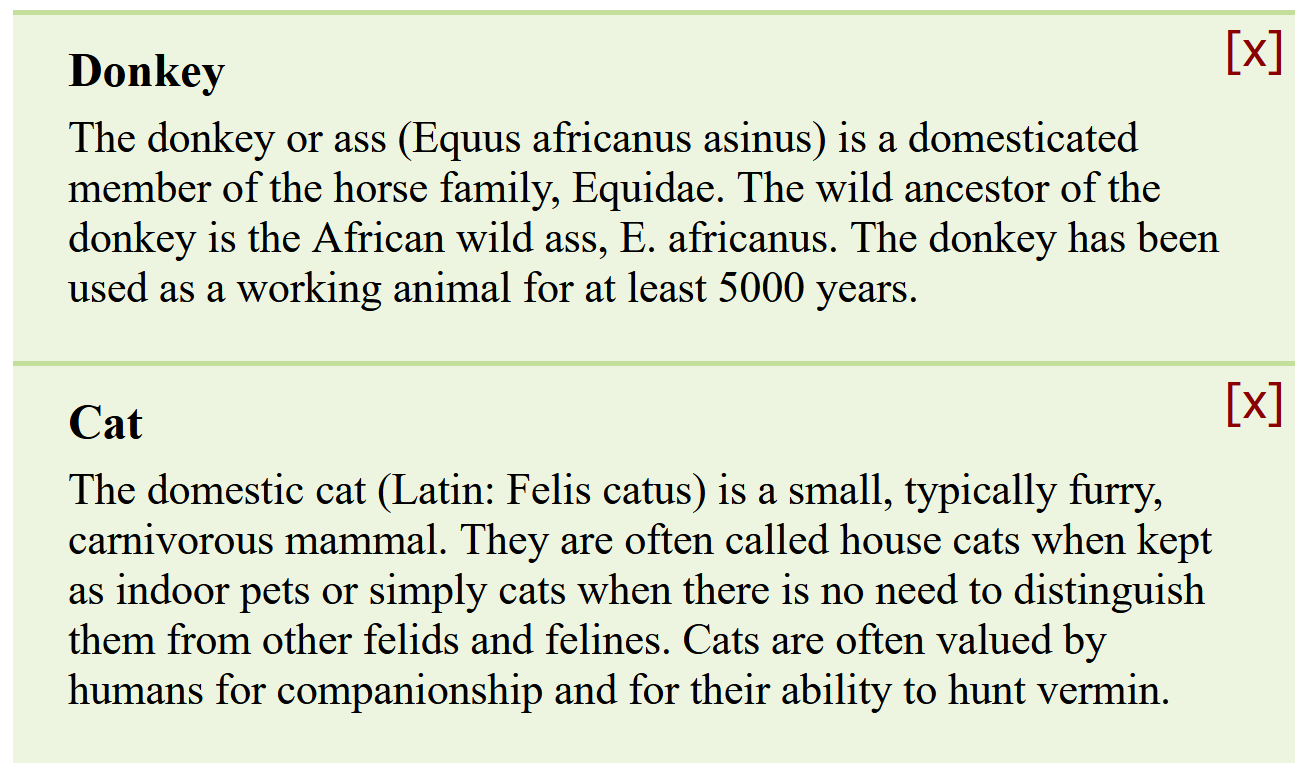
Tulos seuraavalla sivulla:



Lisää sulkemispainike

Käytä JavaScriptiä lisätäksesi sulkemispainikkeen kunkin viestin oikeaan yläkulmaan. Tuloksen pitäisi näyttää tältä:

Kun painat sulkemispainiketta x, laaatikko häviää näkyvistä.



### Tee tällainen Carousel

<!DOCTYPE html>

<head>

  <meta charset="utf-8">

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div id="carousel" class="carousel">

    <button class="arrow prev">⇦</button>

    <div class="gallery">

      <ul class="images">

        <li><img src="https://en.js.cx/carousel/1.png"></li>

        <li><img src="https://en.js.cx/carousel/2.png"></li>

        <li><img src="https://en.js.cx/carousel/3.png"></li>

        <li><img src="https://en.js.cx/carousel/4.png"></li>

        <li><img src="https://en.js.cx/carousel/5.png"></li>

        <li><img src="https://en.js.cx/carousel/6.png"></li>

        <li><img src="https://en.js.cx/carousel/7.png"></li>

        <li><img src="https://en.js.cx/carousel/8.png"></li>

        <li><img src="https://en.js.cx/carousel/9.png"></li>

        <li><img src="https://en.js.cx/carousel/10.png"></li>

      </ul>

    </div>

    <button class="arrow next">⇨</button>

  </div>

  <script>

    /\* label the images, just for convenience, to visually track them \*/

    let i = 1;

    for(let li of carousel.querySelectorAll('li')) {

      li.style.position = 'relative';

      li.insertAdjacentHTML('beforeend', `<span style="position:absolute;left:0;top:0">${i}</span>`);

      i++;

    }

    /\* configuration \*/

    let width = 130; // image width

    let count = 3; // visible images count

    let list = carousel.querySelector('ul');

    let listElems = carousel.querySelectorAll('li');

    let position = 0; // ribbon scroll position

    carousel.querySelector('.prev').onclick = function() {

      // shift left

      position += width \* count;

      // can't move to the left too much, end of images

      position = Math.min(position, 0)

      list.style.marginLeft = position + 'px';

    };

    carousel.querySelector('.next').onclick = function() {

      // shift right

      position -= width \* count;

      // can only shift the ribbbon for (total ribbon length - visible count) images

      position = Math.max(position, -width \* (listElems.length - count));

      list.style.marginLeft = position + 'px';

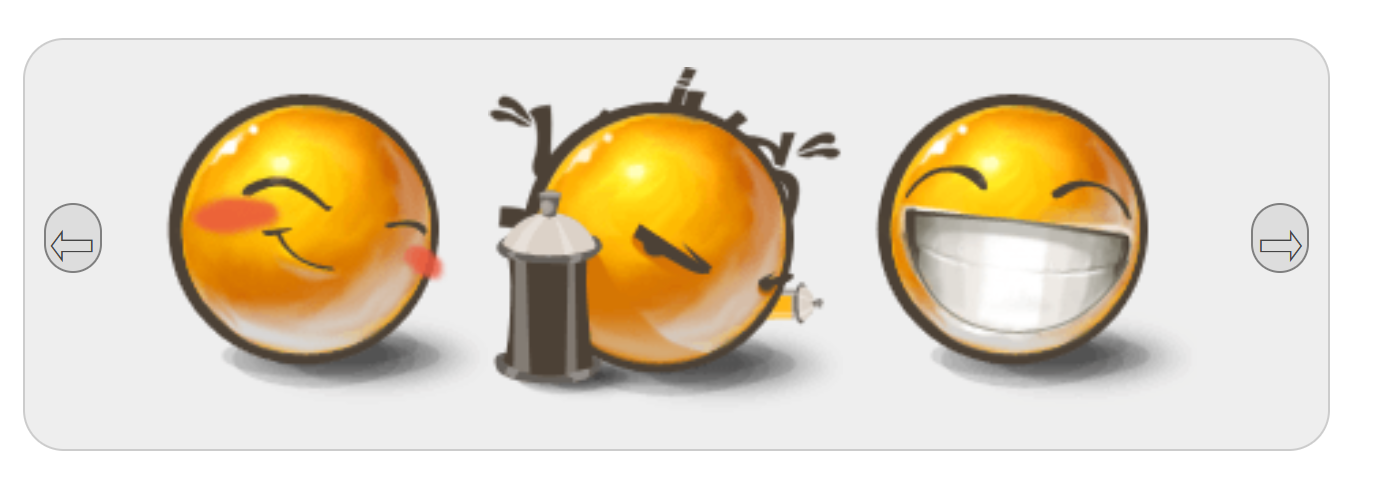
    };

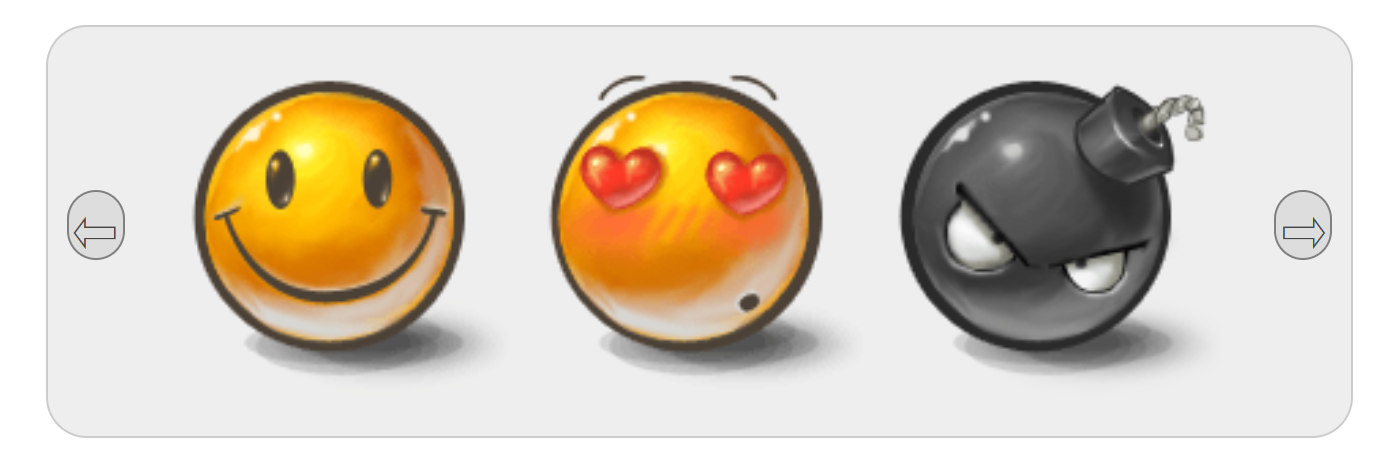
  </script>

</body>

</html>

Lopputulos näyttänee tältä ja tältä (ja vielä muutamalta muulta)?





Tee tehtävät täältä, tehtävät ovat sivun alareunassa:

Event

<https://javascript.info/event-delegation>

Default browser action

<https://javascript.info/default-browser-action>

Mouse events

<https://javascript.info/mouse-events-basics>

mousemove, mouseover, mouseout, mouseenter, mouseleave

<https://javascript.info/mousemove-mouseover-mouseout-mouseenter-mouseleave>

mouse drag and drop

<https://javascript.info/mouse-drag-and-drop>

Keyboard events

<https://javascript.info/keyboard-events>

onscroll

<https://javascript.info/onscroll>

form elements

<https://javascript.info/form-elements>

focus blur

<https://javascript.info/focus-blur>

events change input

<https://javascript.info/events-change-input>

forms-submit

<https://javascript.info/forms-submit>

Kokeilepas myös noita…

<https://javascript.info/css-animations>

<https://javascript.info/js-animation>