JavaScript ohjelmointi
DOM tutustuminen alkaa

MARGIT TENNOSAARIN MATERIAALISTA
LAURA JÄRVISEN MUOKKAAMA

DOM peruskäyttö

https://github.com/bc-web-ohjelmistokehitys/WP25K-JS/blob/main/03_viikko/dom_perusk%C3%A4ytt%C3%B6.md

DOM

Document Object Model (DOM) on ohjelmointirajapinta verkkodokumenteille. Se esittää verkkosivun siten, että ohjelmat voivat muokata dokumentin rakennetta, tyyliä ja sisältöä.
Kun opit käyttämään DOM:ia, voit tehdä verkkosivuistasi vuorovaikutteisempia ja laittaa ne reagoimaan käyttäjän toimintaan.

DOMin muokkaus on laaja kokonaisuus, jonka parissa tulemme viettämään useamman tunnin.

…DOM?

Itse DOM on yksinkertainen asia, jonka monimutkainen nimi saattaa tehdä siitä vaikean hahmottaa.

Tärkeintä on ymmärtää, että DOMia manipuloimalla **muokkaamme verkkosivun sisältöä**.

Funktion kutsuminen

Funktion sisällä oleva koodi suoritetaan, kun "jokin" kutsuu (eli aktivoi) funktion:

- **Kun tapahtuma tapahtuu (esimerkiksi käyttäjä klikkaa painiketta)**
- Kun se kutsutaan JavaScript-koodista
- Automaattisesti (itse itseään kutsuva funktio)

Keskitytään nyt ensimmäiseen vaihtoehtoon.

HTML tapahtumat

Tässä on joitakin esimerkkejä HTML-tapahtumista:

- HTML-verkkosivu on latautunut valmiiksi
- HTML-syöttökentän arvoa on muutettu
- HTML-painiketta on klikattu

https://www.w3schools.com/jsref/dom_obj_event.asp

JavaScript voi muuttaa kaikki HTML-elementit sivulla

JavaScript voi muuttaa kaikki HTML-attribuutit sivulla

JavaScript voi muuttaa kaikki CSS-tyylit sivulla

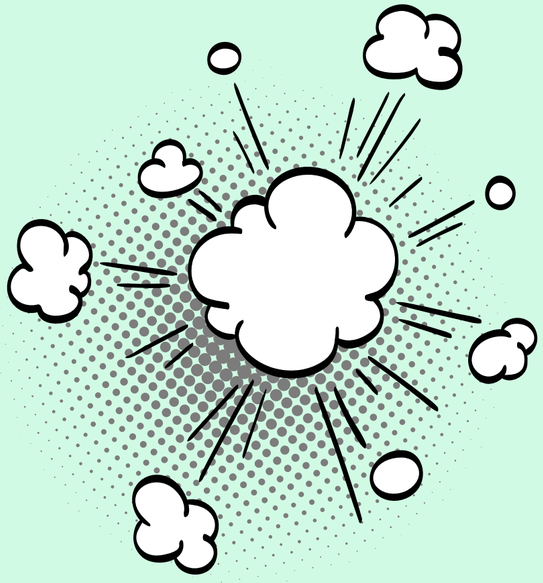
JavaScript voi poistaa olemassa olevat HTML-elementit ja attribuutit

JavaScript voi lisätä uusia HTML-elementtejä ja attribuutteja

JavaScript voi reagoida kaikkiin olemassa oleviin HTML-tapahtumiin sivulla

JavaScript voi luoda uusia HTML-tapahtumia sivulle

Funktion kutsuminen HTML elementistä



+

tapahtuma

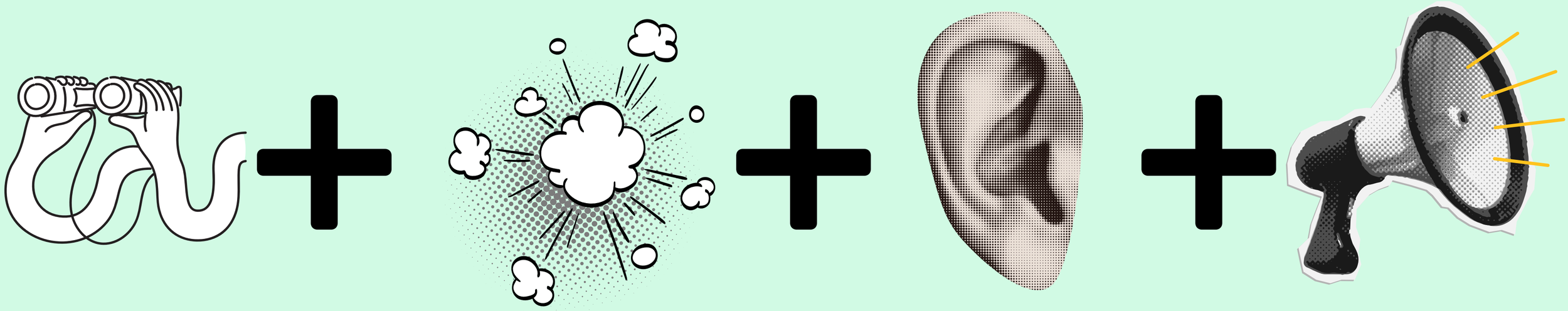
funktiokutsu

Funktion kutsuminen HTML:stä

```
<elementti tapahtuma="jotain JavaScriptiä">

<button onclick="alert('Painiketta klikattu!')">Klikkaa minua!</button>
<button onclick="displayDate()">Paljonko kello on?</button>

<img src="./image.jpg" onclick="displayDate()" />
<div onclick="displayDate()">Paljonko kello on?</div>
```

```
<button onclick="showAlert()">Klikkaa minua</button>

<script>
  function showAlert() {
    alert('Painiketta klikattu!');
  }
</script>
```

Funktion kutsuminen JavaScriptistä



JavaScriptissä määritellään:

- mitä HTML elementtiä tarkkaillaan sekä
- minkälaista tapahtumaa odotetaan.
- Tätä aletaan kuuntelemaan
- ja jos haluttu tapahtuma tapahtuu, kutsutaan funktiota.

Funktion kutsuminen JavaScriptistä: Mikä tai mitkä elementit

HTML-valitsimia käytetään JavaScriptissä HTML-elementtien valitsemiseen ja muokkaamiseen verkkosivulla.

- getElementById()
- getElementsByClassName()
- getElementsByTagName()
- querySelector()
- querySelectorAll()

Valitsimien avulla voit helposti tunnistaa elementtejä niiden yksilöllisten ID-tunnisteiden, luokkien, tagien tai jopa monimutkaisten CSS-valitsimien perusteella.

Mikä tai mitkä elementit

```
// Valitsee elementin, jonka ID on 'myElement'
const element = document.getElementById("myElement");

// Valitsee kaikki elementit, joilla on luokka 'myClass'
const elements = document.getElementsByClassName("myClass");

// Valitsee kaikki <div>-elementit
const divs = document.getElementsByTagName("div");

// Valitsee ensimmäisen elementin, jolla on luokka 'myClass'
const firstElementWithMyClass = document.querySelector(".myClass");

// Valitsee ensimmäisen <p>-elementin <div>-elementin sisällä
const firstPInDiv = document.querySelector("div > p");

// Valitsee kaikki elementit, joilla on luokka 'myClass'
const allElementsWithMyClass = document.querySelectorAll(".myClass");

// Valitsee kaikki <p>-elementit <div>-elementtien sisällä
const allPInDivs = document.querySelectorAll("div > p");
```

Muista D niin kuin DOM

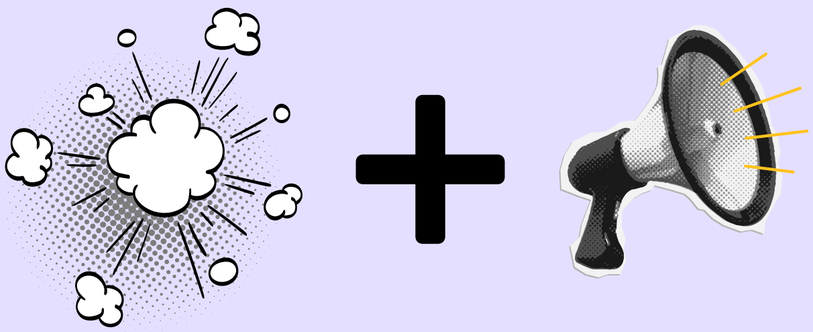
```
const element = document.getElementById("myElement");
```

Funktion kutsuminen JavaScriptistä: Tapahtuman kuuntelu ja funktiokutsu

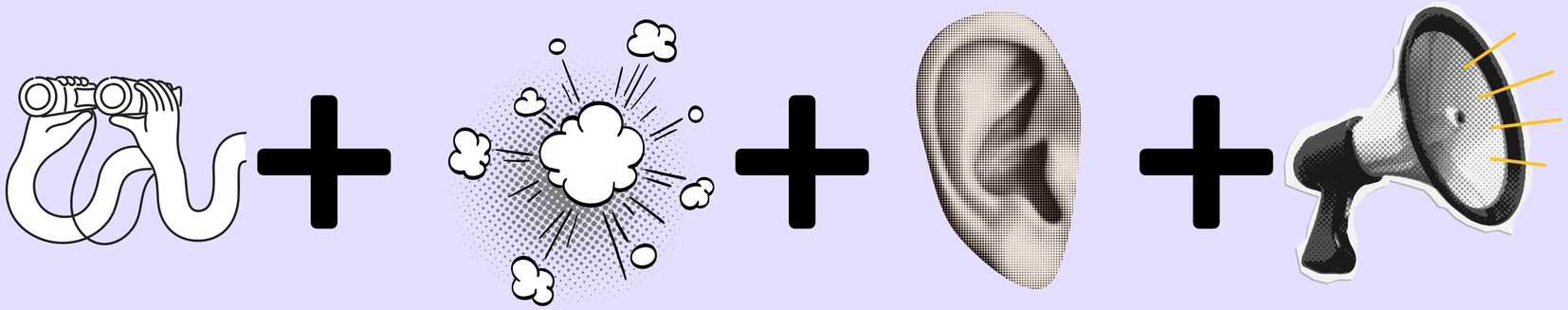
Funktio voidaan määritellä osana tätä rimpsua, jolloin sille ei tarvitse antaa nimeä.

```
document.getElementById("myButton").addEventListener("click", function () {
  alert("Button clicked!");
});

document.getElementById("myButton").addEventListener("click", myFunction);
```

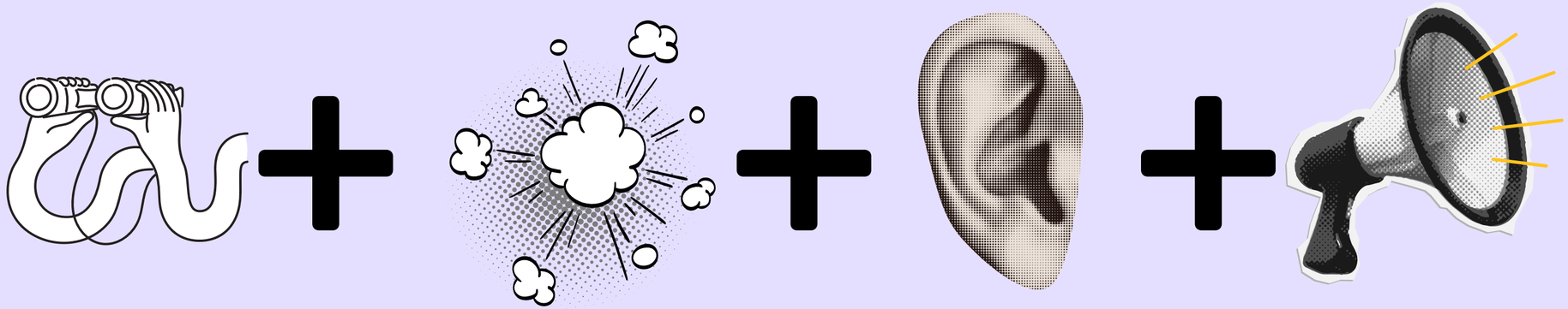
Voidaan kutsua nimeltä myös muualla määriteltyä funktiota.



Toinen näistä näyttää yksinkertaisemmalta



Kumpi on parempi?



Miksi?

- Pitää HTML:n siistinä ja JavaScriptin järjestyksessä.
- Mahdollistaa useiden tapahtumakuuntelijoiden lisäämisen samalle elementille.
- Helpottaa tapahtumakäsittelijöiden poistamista tai päivittämistä.

HTML tekstisisällön muuttaminen

Palataan tähän myöhemmin, mutta ensimmäisissä tehtävissä tarvittava tekstisisällön muuttaminen tapahtuu näin:

```
// elementti.textContent = "Uusi sisältö";

document.getElementById("tervehdys").textContent = "Tervepä terve!";
```

+

```
<body>
  <h1>esimerkki</h1>
  <p id="tervehdys">koodi</p>
```

=

Harjoitellaan

Kolme ensimmäistä tehtävää on lisätty Githubiin.

https://github.com/bc-web-ohjelmistokehitys/WP25K-JS

Huom! Opettele itse - älä pyydä tekoälyä ratkaisemaan tehtävää puolestasi.