

JavaScript ohjelmointi

DOM tutustuminen jatkuu

MARGIT TENNOSAARIN MATERIAALISTA
LAURA JÄRVISEN MUOKKAAMA

DOM

DOMia manipuloimalla **muokkaamme verkkosivun sisältöä**.

Muokkaus tapahtuu reaktiona johonkin tapahtumaan HTML:ssä.

Tarkastellaan HTML tapahtumia 1

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

Tarkastele yllä olevasta linkistä listaa HTML-tapahtumista.

- Kirjoita itsellesi ylös kaikki ne, joita et ymmärrä.
- Käykää opettajan johdolla läpi koko lista ja selvittäkää ne, joita joku ei ymmärtänyt. (On ok, jos opettajakaan ei tiedä ihan kaikkia ulkoa.)

Tarkastellaan HTML tapahtumia 2

Mihin 5 tapahtumaan olet itse huomannut verkkosivun reagoivan eniten / useimmin?

Onko jotain tapahtumaa, jota et muista koskaan käyttäneesi tai huomannut sivun niihin koskaan reagoivan?

Yleisimmät tapahtumat ovat:

1.click

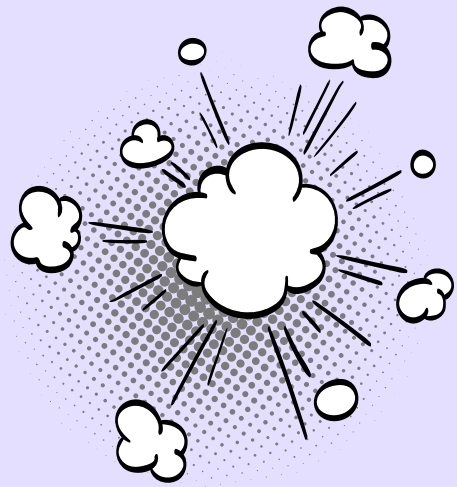
2.mouseover

3.keydown

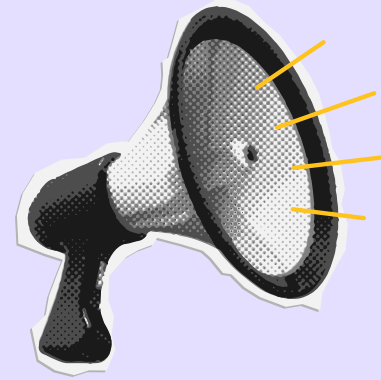
4.submit

5.load

Funktion kutsuminen HTMLstä



+

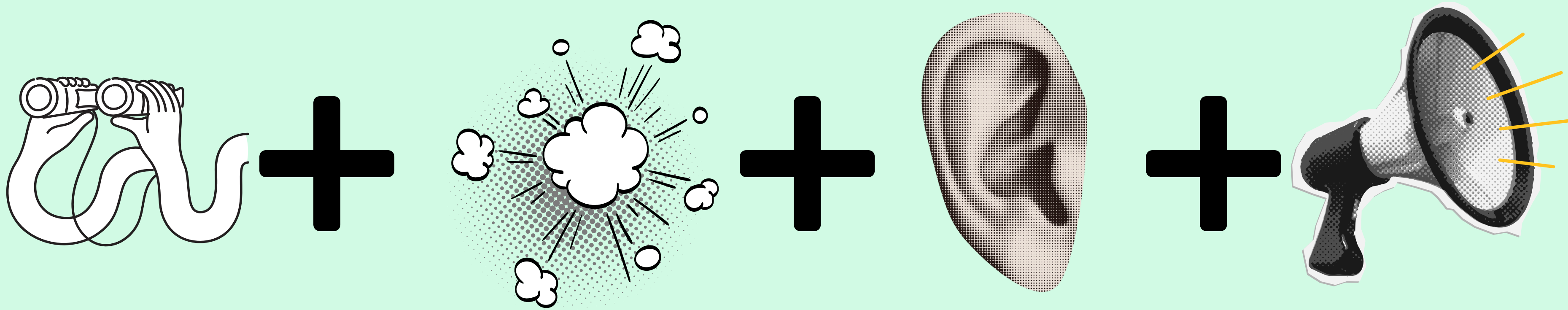


```
  
<div onclick="displayDate()">Paljonko kello on?</div>
```

Elementtiin lisätään tapahtumankäsittelijä, joka kertoo mihin tapahtumaan reagoidaan sekä funktio, jota sillä kutsutaan.

Näin voi tehdä, suosittelen silti toista tapaa eli...

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

```
// Valitse elementin, jonka ID on 'myElement'  
const element = document.getElementById("myElement");  
  
// Valitse kaikki elementit, joilla on luokka 'myClass'  
const elements = document.getElementsByClassName("myClass");  
  
// Valitse kaikki <div>-elementit  
const divs = document.getElementsByTagName("div");  
  
// Valitse ensimmäisen elementin, jolla on luokka 'myClass'  
const firstElementWithMyClass = document.querySelector(".myClass");  
  
// Valitse ensimmäisen <p>-elementin <div>-elementin sisällä  
const firstPInDiv = document.querySelector("div > p");  
  
// Valitse kaikki elementit, joilla on luokka 'myClass'  
const allElementsWithMyClass = document.querySelectorAll(".myClass");  
  
// Valitse kaikki <p>-elementit <div>-elementtien sisällä  
const allPInDivs = document.querySelectorAll("div > p");
```

HTML elementin muuttaminen

```
elementti.innerHTML = "<p>New Content</p>";
```

Koko HTML-sisällön muokkaaminen. Asettaa (tai lukee) HTML-koodia merkkijonona.

```
elementti.textContent = "New Text Content";
```

Asettaa pelkän tekstin, ei tulkkaa HTML-tageja. Kaikki annettu sisältö käsitellään puhtaana tekstinä.

```
elementti.innerText = "Visible Text Content";
```

Näyttää vain näkyvän tekstin, ottaa huomioon CSS:n kuten display: none. Käsittelee rivinvaihtoja ja välejä visuaalisesti.

Lisää DOM manipulaatiometodeja

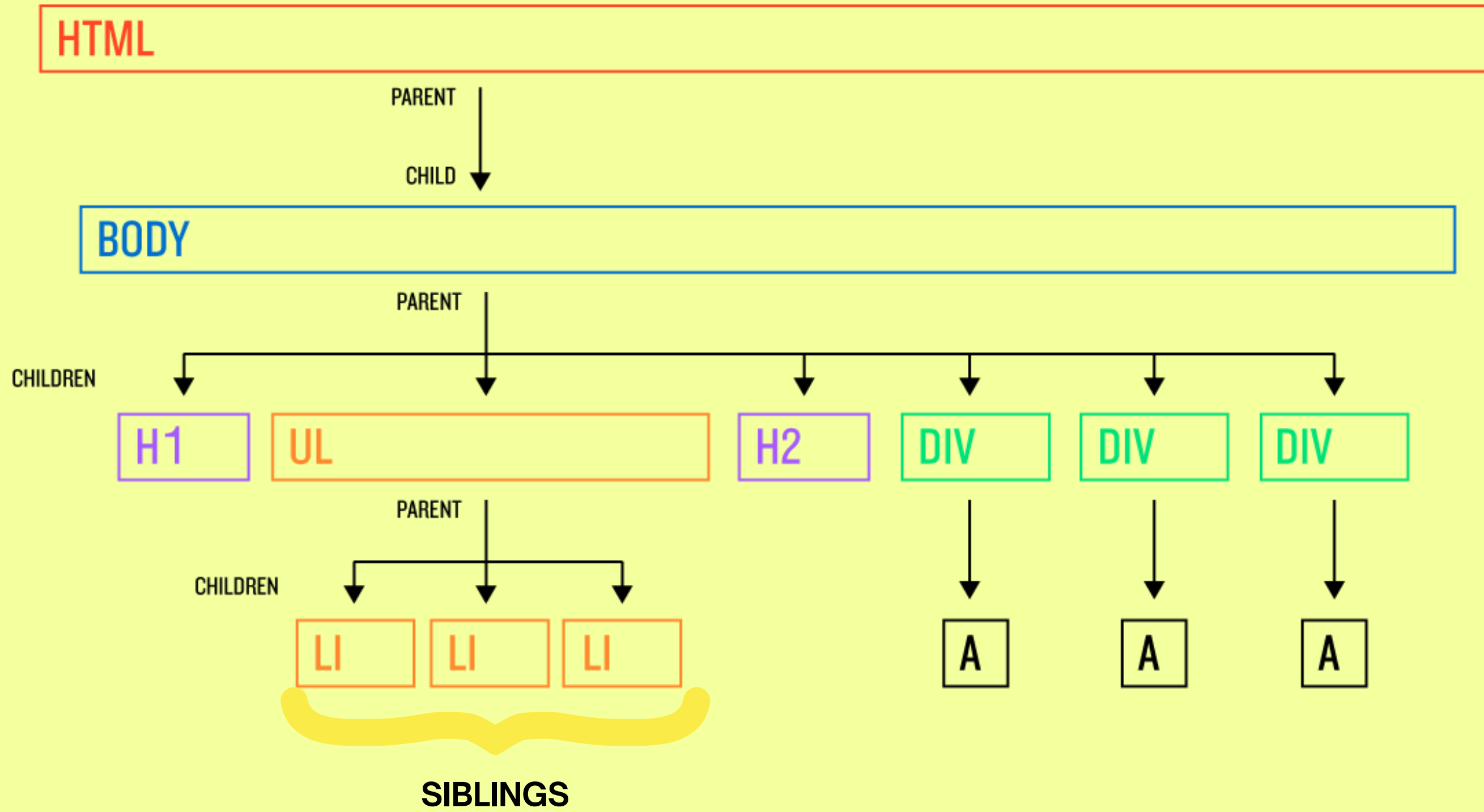
appendChild: Lisää uuden lapsielementin.

removeChild: Poistaa olemassa olevan lapsielementin.

setAttribute: Asettaa tai muokkaa attribuuttia.

classList-metodit: Lisää, poistaa, vaihtaa tai tarkistaa luokkia.

HTML sukulaisuhteet



HTML-attribuuttien muuttaminen

```
element.setAttribute("class", "newClass"); // Asettaa uuden class-attribuutin  
let className = element.getAttribute("class"); // Hakee class-attribuutin arvon  
element.removeAttribute("class"); // Poistaa class-attribuutin
```

CSS tyylien muuttaminen

```
element.style.color = "red"; // Asettaa tekstin väriksi punaisen
element.style.fontSize = "16px"; // Asettaa fonttikooksi 16px

element.className = "newClassName"; // Asettaa elementin class-attribuutiksi 'newClassName'

// classList
element.classList.add("newClass"); // Lisää 'newClass'-luokan

element.classList.remove("oldClass"); // Poistaa 'oldClass'-luokan
element.classList.toggle("active"); // Lisää tai poistaa 'active'-luokan riippuen sen nykytilasta

element.classList.contains("active"); // Tarkistaa, onko 'active'-luokka käytössä
```

HTML elementin ja attribuutin poistaminen

```
element.remove(); // Poistaa elementin suoraan
```

```
parentElement.removeChild(childElement); // Poistaa lapsielementin vanhemmastaan
```

Uuden HTML elementin ja attribuutin lisääminen

```
let newElement = document.createElement("div"); // Luo uuden <div>-elementin

parentElement.appendChild(newElement); // Lisää uuden elementin vanhemman loppuun

parentElement.insertBefore(newElement, referenceElement);
// Lisää uuden elementin ennen viite-elementtiä

parentElement.append(newElement, "Some text"); // Lisää elementin ja tekstin vanhemman loppuun

parentElement.prepend(newElement, "Some text"); // Lisää elementin ja tekstin vanhemman alkuun
```


Lomakkeet ja käyttäjän syötteet

Lomakkeet tulevat teille tällä viikolla HTML ja CSS -tunneilla, tässä jo muutama asia niihin liittyen.

Syöte tekstikentästä

```
<input type="text" id="username" placeholder="Syötä nimesi" />
<button id="submitBtn">Lähetä</button>
<p id="output"></p>
```

```
function displayUsername() {
  const inputValue = document.getElementById("username").value;
  document.getElementById("output").textContent =
    "Käyttäjä syötti: " + inputValue;
}

document.getElementById("submitBtn").addEventListener("click", displayUsername);
```

Syötteen live-seuraaminen

```
<input type="text" id="liveInput" placeholder="Aloita kirjoittaminen..." />
<p id="liveOutput"></p>
```

Heissa

Kirjoitit: Heissa

```
function showLiveText(event) {
  document.getElementById("liveOutput").textContent =
    "Kirjoitit: " + event.target.value;
}

document.getElementById("liveInput").addEventListener("input", showLiveText);
```

Lomakkeen käsittely JavaScriptillä osa 1

```
<form id="userForm">
  <input type="text" id="name" placeholder="Nimesi" />
  <input type="email" id="email" placeholder="Sähköpostiosoitteesi" />
  <button type="submit">Lähetä</button>
</form>
<p id="formOutput"></p>
```

Lomakkeen käsittely JavaScriptillä osa 2

```
function processForm(event) {  
  event.preventDefault(); // Estetään lomakkeen oletuslähetys  
  const name = document.getElementById("name").value;  
  const email = document.getElementById("email").value;  
  document.getElementById(  
    "formOutput"  
  ).textContent = `Lähetetty nimi: ${name}, Sähköposti: ${email}`;  
}  
  
document.getElementById("userForm").addEventListener("submit", processForm);
```

Lähetetty nimi: Opettaja, Sähköposti: ope@bc.fi

Harjoitellaan

Tehtäviä on lisätty viikon 4 harjoitukseen.

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

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