

Päivä 3: Web-ohjelmointi jatkuu

MIMMIT KOODAA -BOOTCAMP 2020
JANI JÄRVINEN, INTERTECHNO TRAINING OY

Web-tekniologia, jatkoa

CSS-kielen valitsimet

- Valitsimet ovat CSS-kieleen kuuluva osa, joiden avulla valitaan HTML rakenteesta se osa, johon halutaan ulkoasusääntöjen kohdistuvan

- Engl. selectors

- Valitsimia on useita eri tyyppisiä, joista keskeisimmät ovat:

- Elementtivalitsimet, esim. p, h1
 - Luokkavalitsimet, esim. .omaluokka
 - Id-valitsimet, esim. #omaid

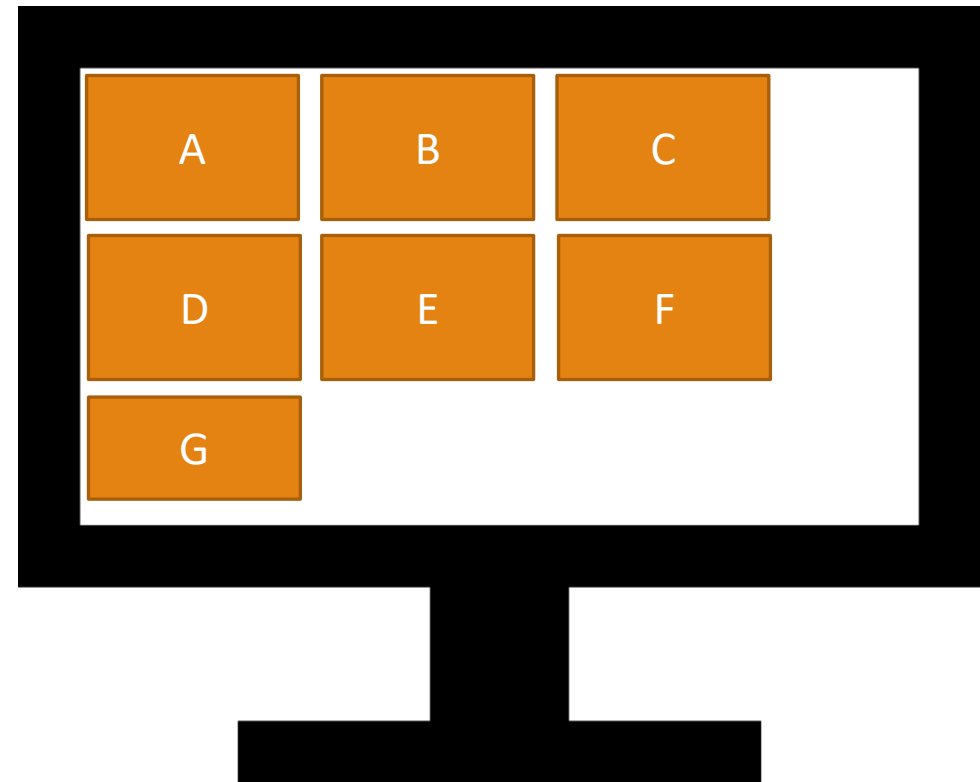
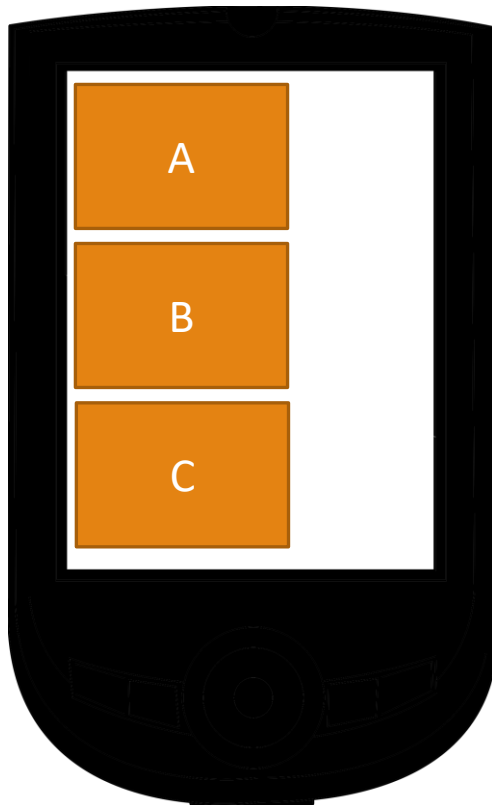
- Esimerkki HTML-määrittelyistä:

```
<h1>Tämä on otsikko</h1>
```

```
<p class="omaluokka">Tämä on esimerkki.</p>
```

```
<p id="omaid">Toinen kappale tekstiä.</p>
```

Esimerkki asemoinnista



CSS Media Query – eli mediakyselyt

- Monilla sivustoilla lähtökohta responsiivisuudelle
- Tarvittaessa sama asia voidaan toteuttaa JavaScriptillä, mutta tämä on sekä hitaampi että kömpelömpi ratkaisu
- Voidaan toteuttaa joko CSS-tiedostojen sisällä tai <link rel="stylesheet"> -määrittelyksellä:

```
@media screen and (min-width:  
600px) {  
    body {  
        background-color: blue;  
    }  
}
```

```
<link rel="stylesheet"  
media="mediatype and (min-width:  
600px)" href="blue.css">
```

Canvas-esimerkki

```
<canvas id="mycanvas" width="300"  
height="300"></canvas>
```

```
<script type="text/javascript">
```

```
var myCanvas =  
document.getElementById("mycanvas");
```

```
var myContext =  
myCanvas.getContext("2d");
```

```
myContext.fillStyle = '#00f';
```

```
myContext.strokeStyle = '#F63806';
```

```
myContext.lineWidth = 4;
```

```
myContext.fillRect(0,0,300,300);
```

```
myContext.strokeRect(0,0,300,300);
```

```
myContext.moveTo(150,0);
```

```
myContext.lineTo(0,150);
```

```
myContext.lineTo(150,300);
```

```
myContext.lineTo(300,150);
```

```
myContext.lineTo(150,0);
```

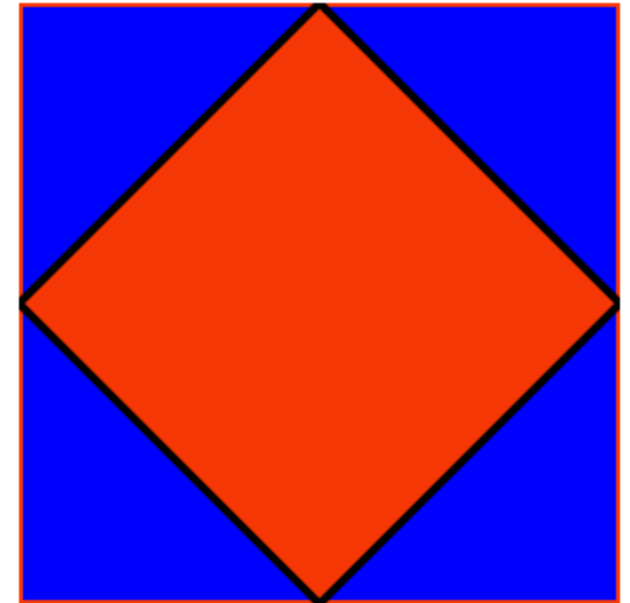
```
myContext.fillStyle = '#F63806';
```

```
myContext.strokeStyle = "#000";
```

```
myContext.fill();
```

```
myContext.stroke();
```

```
</script>
```



Progressive Web Applications (PWA)

- Alun perin Googlelta lähtöisin oleva määritys, joka mahdollistaa web-sovellusten paremman integraation käyttöjärjestelmän kanssa sekä toisaalta esimerkiksi offline-tuen
- Sovellukset voivat tuoda ominaisuuksia lisää ”progressiivisesti” eli tilanteen ja päätelaitteen kykyjen mukaan
- PWA-sovellukset ovat käyttökelpoisia silloinkin, kun verkkoyhteyksiä ei ole tai ne ovat heikkolaatuisia
- Tulevaisuudessa on mahdollista, että PWA-sovelluksia voidaan asentaa Android-järjestelmiin Googlen Play -kaupan ulkopuolelta, Microsoft on jo mahdollistanut tämän omassa Windows Storessaan
- Tarkemmin: <https://developers.google.com/web/progressive-web-apps/>

WebAssembly (Wasm)

- Uusi tapa julkaista web-sovelluksia
- Binäärinen tiedostoformaatti, jolla voidaan jaella esimerkiksi C/C++ -ohjelmointikiellä tehtyjä sovelluksia webin ylitse selainympäristöihin
- Laaja selaintuki, vaikka standardi on vasta tuore
- Perustuu virtuaalikoneeseen, jota ajetaan esimerkiksi selaimen sisällä
- W3C-organisaation standardiehdotus
- Tarkemmin: <https://webassembly.org/>

Git-versionhallinta

Perustetaan tili GitHubiin

- Perustetaan oma tili GitHubiin
 - Osoitteena voi käyttää esimerkiksi henkilökohtaista sähköpostiosoitetta
 - Voit tietenkin käyttää myös aiemmin luotua tiliäsi!
- Siirry GitHubiin web-selaimella ja valitse ”Sign up” -toiminto
- Tilin luonnin jälkeen tarvitset koodille tallennuspaikan eli *repositoryn* eli *repon*, suomeksi säilö
- Voit lisäksi valita, tuleeko koodisi näkyviin kaikille (Public), vai pysyykö se yksityisenä (Private)

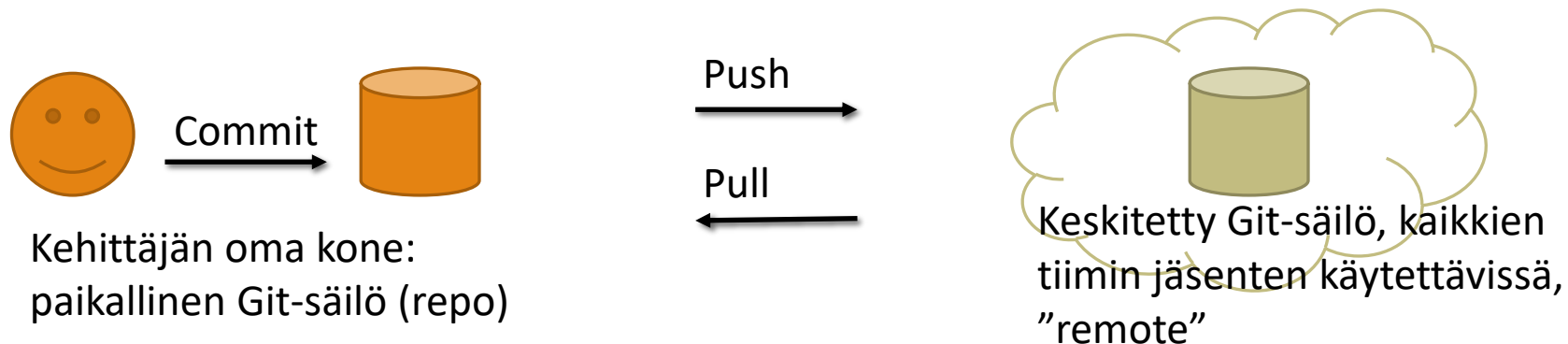
The image shows a composite of two screenshots from the GitHub website. The top screenshot is the GitHub homepage, featuring the slogan 'Where software is built' and a sign-up form with fields for 'Pick a username', 'Your email', and 'Create a password'. The bottom screenshot is the 'Create a new repository' form, which includes fields for 'Owner' (jani-jarvinen) and 'Repository name' (firsttest), a 'Description (optional)' text area, and radio buttons to select between 'Public' and 'Private' visibility. There is also a checkbox for 'Initialize this repository with a README'.

Git-versionhallinta

- Git on alun perin Linuxin kehittämistä varten suunniteltu versionhallintajärjestelmä, joka on nykyään hyvin suosittu
- Git on lähtökohtaisesti komentorivipohjainen ohjelma, mutta usein sitä käytetään jonkin koodieditorin kautta
- Perustuu hajautettuun (distributed) arkkitehtuuriin
 - Kukin kehittäjä työskentelee oman paikallisen versionhallintansa kanssa, mutta synkronoi tiedot palvelimen kanssa säännöllisesti
- Keskeinen käsite: repository eli repo, suomeksi säilytyspaikka tai säilö
- Latauslinkki
 - <https://git-scm.com/downloads>

Pilvipohjaiset Git-versionhallintapalvelut

- Git-versionhallinta on lähtökohtaisesti aina paikallinen järjestelmä, josta voidaan kuitenkin kopioida tietoja nk. etäpalvelimelle (engl. remote)
- Kun Gitin kanssa tehdään tiimityötä, on erityisesti tärkeää muistaa tallentaa omat muutokset etäpalvelimelle riittävän usein: tähän käytetään push-komentoa
- Markkinoilla on saatavilla useita sekä maksuttomia että maksullisia pilvipalveluita, joita voidaan käyttää Git-versionhallinnan etäpalvelimenä
 - Esimerkiksi GitHub (Microsoft), Bitbucket (Atlassian), Gitlab, Unfuddle, jne.
 - Kurssilla käytämme esimerkkinä GitHubia, <https://github.com/>



Tärkeimmät Git-komennot

- Alustus
 - Init: alustaa hakemiston Git-säilytyspaikaksi
 - Clone: kloonaa eli lataa verkosta jonkin Git-säilytyspaikan tiedot
- Tietojen haku ja lähetys palvelimelle
 - Fetch: hakee palvelimelta tiedot uusimmista tiedostoista
 - Pull: uusimpien tiedostojen lataaminen omalle koneelle
 - Push: ominen muutosten lähettäminen takaisin palvelimelle
- Tallennusten tekeminen paikallisesti
 - Commit: muutosten tallentaminen paikallisesti osaksi versionhallintaa
- Haarat
 - Branch: muodostaa uuden haaran paikallisesti
 - Checkout: vaihtaa haaraa (tämän jälkeen tyypillisesti Pull)
- Tilatieto
 - Status: kertoo, missä tilassa paikalliset tiedostot ovat

Tärkeitä Git-taitoja

- Kun Gitin käyttöä opetellaan, on tärkeää hallita vähintään seuraavia perustietoja
- Gitin alustaminen käyttöön
 - Git init -komento
- Olemassa olevan Git-säilön (repon) kloonaaminen omalle koneelle
 - Git clone -komento
- Muutosten tallentaminen omaan Git-tietokantaan
 - Git commit -komento
- Tilan tarkistaminen ja uusien tiedostojen lisääminen
 - Git status- ja Git add -komennot
- Historiatietojen katsominen
 - Git log -komento