



# Verkkosivujen Rakenne (HTML)\*

CS-C1180 Verkojulkaisemisen perusteet (5 op)

Pauli Laine

[pauli.laine@aalto.fi](mailto:pauli.laine@aalto.fi)

*\*Perustuu Hanna Hämäläisen Antti Tolppasen, Pia Tukkinen ja Hannu Järvisen kalvoihin*

***”Luennon jälkeen osaan tehdä yksinkertaiset verkkosivut”***

# Miksi HTML

- Jotta voidaan tehdä verkkosivuja (joita selaimet avaavat)
- Elementit mahdollistavat **rakenteisen** tiedon esittämisen (ei yhtä pötköä).
- Eri sivut mahdollistavat **sujuvan liikkumisen** tiedon eri osien välillä -> navigointi
- HTML muodostaa rungon, jonka päälle voi lisätä toiminnallisuutta (javascript) ja muotoilua (CSS).

# VJP-kurssin fokus

- Selaimessa tapahtuvan tiedon esittäminen, muotoilu ja toiminnallisuudet
  - Toiminnallisuus ja erityisesti tiedon tallentaminen hyvin rajoitettua
  - Voidaan pitkälti kokeilla omalla koneella (sandbox)
- Tällä kurssilla EI käsitellä palvelimessa tapahtuvaa verkkojulkaisemista jossa:
  - Esim. haetaan tietokannoista tietoa, muotoillaan ja esitetään loppukäyttäjälle
  - Vastataan loppukäyttäjän tekemiin palvelupyyntöihin
  - Koodataan usein PHP:llä tai Microsoft .net:llä

# Sisältö

- Johdanto ja käytännöistä
  - Verkkosivut ja informaatioarkkitehtuuri
  - HTML
    - Historia ja nykypäivä pähkinänkuoressa
  - Koodin rakenne ja käsittely
    - DOM-puu
    - HTML5 elementit
  - Yksinkertaisen sivurungon rakentaminen - Demo
  - Sivuston suunnittelu ja navigointi
  - Informaatioarkkitehtuuri
  - Semantiikka
  - Yhteenveto
-

# Tuntitehtävien luettelo

- Sivuston suunnittelu paperilla
- Tutustuminen kehitystyökaluihin ja kokeilu (ikkunat, lähdekoodin katselu, kehitystyökalut)
- Yksinkertainen linkittämätön sivu, jolla erilaisia elementtejä
- Alasivulle linkattu kotisivu
- Kokonainen sivusto
- Kokonainen sivusto, jossa linkit toteutettu ikonein

# Johdanto ja hyvät käytännöt

- Vain rakenne HTML, muotoilu CSS, toiminnallisuus Javascript
- Ikkunoita kolme, selain, tekstieditori, tietoruutu (huom. läppäriin ergonomia, ruututila). Minä käytän Notepad++
  
- HTML täytyy itse oppia, tagien luetteleminen ei auta
  
- Kokeile->tallenna->arvioi->muuta (opi näppäinkomennot)
- Lähdekoodin tutkiminen ctrl-u
- Selaimen kehitystyökalut

# Käytännössä

- Ikkunat:
  - editori (josta mieluiten näppäinkomennolla selaimeen)
  - selain
  - tietoa HTML-tageista yms
  - tiedostoselain (file explorer)
- Mieti ensin kansiorakenne
- Editoi, tallenna, kokeile



# Miksi tehdään tekstieditorilla eikä käytetä valmiita palveluja?

- Jos haluat vain sivut, voit käyttää esim:
  - Sites.google.com
    - Katso esim. lähdekoodi
  - Wix
  - Kotisivukone
- Jos haluat oppia kuinka sivuja tehdään?

# Sivuston rakenteesta

- Periaatteessa jokainen sivu on yksi tiedosto (esim. yhteystiedot.html)
- Kun tehdään linkkejä, niin laitetaan 'ankkuri' `<a href>` joka viittaa tuohon tiedostoon.
- **Kehitysvaiheessa nämä tiedostot ovat omalla koneella.**
- **Kun sivusto julkaistaan, ne siirretään palvelimelle -> koko maailma voi nähdä ne!**

# INFORMAATIOARKKITEHTUURISTA

- Verkkosivujen tarkoitus
- Selkeä tiedon esittäminen edellyttää usein rakenteista tiedon esittämistä
- Mikä tieto kuuluu mihinkin osioon?
- Miten käyttäjä tietää missä osassa hän on?
- Mitä käyttäjän pitää milloinkin nähdä?
- Miten sivustolla liikutaan

# INFORMAATIOARKKITEHTUURISTA - jatkoa

- Sivuston yhteydet paikallisesti ja muualle
- Onko tarkoitus avata alasivuja, jotka vievät muualle (esim. BBC.COM)
- Tarvitaanko sivun sisäisiä linkkejä?

# Sivustolla liikkumisen periaatteet - informaatioarkkitehtuuri

- INFORMATION ARCHITECTURE IS the science of figuring out what you want your site to do and then constructing a blueprint before you dive in and put the thing together.
- [https://www.wired.com/2010/02/information\\_architecture\\_tutorial/](https://www.wired.com/2010/02/information_architecture_tutorial/)
- Yksinkertaisia esimerkkejä:  
<https://www.webpagefx.com/blog/web-design/information-architecture-101-techniques-and-best-practices/>

# Linkkejä

- <http://webstyleguide.com/wsg3/3-information-architecture/3-site-structure.html>
  - <http://www.webopas.net/vinkit.html>
  - [https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML/Document\\_and\\_website\\_structure](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure)
  - <http://www.htmlquick.com/tutorials/organizing-website.html>
  - <https://code.tutsplus.com/tutorials/28-html5-features-tips-and-techniques-you-must-know--net-13520>
  - <https://fi.wikipedia.org/wiki/Informaatioarkkitehtuuri>
-

# Tehtävä - sivustos suunnittelu paperilla

- Mieti kynää ja paperia apuna käyttäen yksinkertainen sivusto - Kesto 10 minuuttia. Valmistaudu esittelemään sivustosi.
- Voit käyttää myös piirto-ohjelmaa (mieluummin ei)

# *HTML VERKOSSA*

## *Verkkosivustot*



# Verkkosivut omalla koneella

- Omalla koneella kansiossa olevat .html loppuiset tiedostot
- Voidaan avata selaimella
- Eivät näy maailmalle
- Ei tarvita palvelinta
- **MUISTA TALLENTAA** (että voit jatkaa seuraavalla tunnilla!)

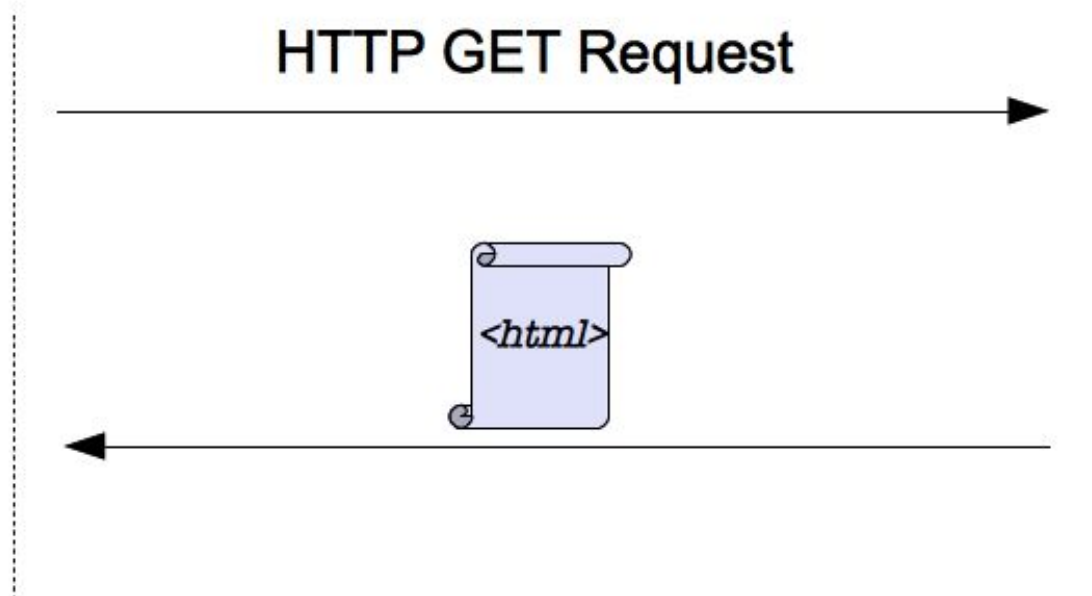
# Verkkosivut

- Internetissä tai intranetissä julkaistu sivu
  - Maailmanlaajuinen tai kohdennettu yleisö
- Julkaistu palvelinkoneella, voidaan kehittää myös paikallisesti
  - Reaaliaikaisesti saatavilla
- Käytetään verkkoselaimella
- Palvelin vastaa käyttäjän pyyntöön ja lähettää koodatun dokumentin, jonka selain muuttaa ihmiselle ymmärrettävään muotoon
- Asiakaspää – palvelinpää

# Asiakaspää - palvelinpää

Asiakaspää  
(HTTP Client, eg.  
Web browser)

Palvelinpää  
(HTTP Server, eg.  
Apache Tomcat)





# HTML 5

- [HyperText Mark-up Language](#)
- Standardoitu kuvauskieli joka määrittelee dokumentin rakenteen
  - Otsikoita, kappaleita, listoja, taulukoita, kuvia, lomakkeita, linkkejä muihin dokumentteihin
- Selaintuki ei ole vielä täydellinen (IE)
  - Selainriippumattomuus vaatii lisätyötä tällä hetkellä.
- Selaintukea voi testata
  - [caniuse.com](http://caniuse.com)
  - [html5test.com](http://html5test.com)

# Sisällön ja ulkoasun erottaminen

- Nykyisin hyvänä käytäntönä pidetään sisällön ja ulkoasun erottamista erillisiin dokumentteihin
  - HTML = sisältö
  - CSS = ulkoasu
  - Lisäksi voidaan lisätä dynaamista toiminnallisuutta ja muokata koodia ohjelmoimalla (yleensä Javascript)
- Etuna helppo päivitettävyyys, ulkoasu voidaan muuttaa kerralla koko sivustolle!
- Voidaan määritellä ulkoasuja eri mediatyypeille käyttäen samaa HTML-dokumenttia.

# Sisältö ja ulkoasu

- Ti 22.1 pelkästään HTML
  - Ei mitään muotoilua
  - Pelkästään sivun/sivuston rakenne
  - Ei toiminnallisuutta
  
- Ti 29.1 CSS joka yhdistetään HTML:n
  - Oman sivuston tyylin kehittäminen

# HTML koodin rakenne ja elementit

## ja yksinkertaisen sivurungon rakentaminen käytännössä

# Pienin verkkosivu - sivuston purkaminen

```
<html>
```

```
  Kukkuu
```

```
</html>
```



# Tunnisteet ja elementit

- HTML on rakenteista tekstiä, joka koostuu tunnisteista ja elementeistä.
- Merkitään kulmasulkeilla `<` ja `>`
- **Tunnisteen, eli tagin** avulla selain osaa näyttää sivun oikein.
  - `<html>`, `<h1>`, `<p>`, jne..
- **Elementti** tarkoittaa yhden tagin ja sen päättötagin muodostamaa kokonaisuutta.
  - `<tag>...</tag>`
- **Tagit ja elementit** voivat olla sisäkkäisiä tai peräkkäisiä.

**TUTORIAL:** <http://www.quackit.com/html/tutorial/>

# Tunnisteet ja elementit

- Vaikka tunnisteita on todella paljon, välttämättömiä on vain muutamia
  - Rakenteen keskeiset elementit
  - Listat
  - Linkit
  - Kuvat

**TUTORIAL:** <http://www.quackit.com/html/tutorial/>

---

# Luetteloita

- 
- Tunnisteluetteloita
  - <https://www.w3schools.com/html/>
    - <https://www.w3schools.com/tags/default.asp>
    -
- Tunnisteilla saattaa olla parametrejä:
  - id,src,onclick jne...jne...

**TUTORIAL:** <http://www.quackit.com/html/tutorial/>

---

# Peruselementit

- HTML
- Head
- Body
- P
- A
- Div
- Image
- Li
- Button
- Input

# <html>

- Määritellään dokumentin tyyppi
  - [Document Type Declaration](#) (DTD)
- HTML5:ssä yksinkertaisempi, kuin aiemmissa versioissa  
**<!DOCTYPE html>**
- Not Case sensitive kuten ennen! (eli onko isot vai pienet kirjaimet)
- Aiemmissa HTML:n versioissa tarkempi määrittely:  
**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.10//EN"  
http://www.w3c.org/TR/html4/strict.dtd>**
- HTML –tagien sisään tulee koko sivun koodi.

# <head>

- Head -tagissa määritellään dokumentin **metatiedot**.
- Title (muista aina määritellä!)
  - Ilmoittaa sivun otsikon
  - Näkyy selaimen yläpalkissa
- Metatiedot
  - Voidaan merkitä sivulle piilotettuja tietoja
- Linkitetyt tiedostot
  - Sivustoon liittyvät tiedostot, kuten css tai javascript -tiedostot
- Tunnisteikoni
  - URL-kentässä / yläpalkissa näkyvä sivuston logo

# <body>

- Kaikki käyttäjälle sivuilla näytettävä sisältö tulee body-tagin sisään.
- Rakenteen määrittely tapahtuu body-tagin sisällä.
- Ennen body-tagissa määriteltiin dokumentin esitysominaisuuksia. Nykyään kaikki esitysominaisuudet määritellään CSS-tiedostossa.
- Sivusto koostuu body-tagin sisään luotavista elementeistä.
- Sivustolle voidaan määritellä (yhteisiä) globaaleja tai elementtikohtaisia attribuutteja.

# Image

- Taustakuva
- Kuva sivulla
- Kuva linkkinä

```

```



# A

- Anchor eli A on linkkitagi
- Voi viitata samalle sivulle, samaan sivustoon tai ulkopuolelle (internetiin)
- Samalle sivustolle viitattaessa voidaan käyttää suhteellista url:ia (ilman: <https://www....>).

```
<a href="koti.html">Koti</a>
```

```
<a href="https://www.bbc.com" >BBC</a>
```

# Button

- Toimii visuaalisesti, mutta ei tee mitään (vielä)
- Voit nimetä yms.

```
<button>START</button>
```

```
<button id="kaynnistysnappi">START</button>
```

```
<button id="kaynnistysnappi" onclick="kaynnistysfunktio()">START</button>
```

# P

- P eli paragrafi
- Kappale tekstiä

<p>Jotain juttua</p>

# DIV

- Osio, esim. tekstiä, käytetään paljon muotoilun apuna (ensi viikon CSS). Voi sisältää muuta, myös toisia divejä
- Ei sinänsä 'näytä' mitään

```
<div>  
<p> Jotain juttua </p>  
<p> Lisää juttua</p>  
</div>
```

# L

- Lista (asioita, linkkejä yms)
- Voidaan kehystää listamääreillä `ordered list <ol>` tai `unordered list <ul>`

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Milk</li>
```

# iframe

- Avaa url:in samassa ikkunassa, uudessa kehyksessä:
- Paljon tyyliparametrejä
- Esimerkki:

```
<iframe src = "https://www.bbc.com">
```

# Metatiedot

- Kuvailevaa tai määrittelevää tietoa tiedosta
- Metatiedolla pyritään tehostamaan tietovarannon käyttöä
- Parantaa informaation löydettävyyttä
  - Hakukoneet
- Suuri merkitys tiedon automaattisessa järjestelyssä
- Microformats for Google Chrome™

# Favicon.ico tunnisteikoni

- [Favicon](#) (favorites icon) näkyy selaimen osoitekentässä ja usein myös taustalla auki olevan välilehden vasemmassa reunassa sekä kirjanmerkeissä.
- Parantaa sivuston tunnistettavuutta.
- Voit käyttää jotain valmista palvelua ikonin generoimiseen, kuten:
  - <http://www.favicon.cc/>
- Voit generoida kuvan myös sivuston logosta (isompi kuva)
  - Koko 16 x 16 pikseliä

<http://www.w3.org/2005/10/howto-favicon>



**Tehtävä (nämä on jo koulun  
koneella):**

**Asenna Chrome**

**Sublime Text 2/Brackets/Atom/**

**TAI MIELUITEN:**

**Notepad++**

<https://www.google.com/intl/en/chrome/browser/>

<http://www.sublimetext.com/2>

# Selainten kehitystyökalut

## Firefox

[Inspector](#) (sisäänrakennettu tuki)

[Firebug](#) (liitännäinen)

## Safari

[Developer Tools](#) (sisäänrakennettu tuki)

## Chrome

[Developer Tools](#) (sisäänrakennettu tuki)

Näppäinlyhenne: CTR-shift-J

## Kaikki selaimet

[Firebug Lite](#) (JavaScript-kirjasto)

# Validointi

Voit tarkistaa HTML-koodisi oikeellisuuden seuraavalla sivustolla:

<https://validator.w3.org>

# Tuntitehtävä 1

## Tee yksinkertainen sivu jolla on:

- Tekstiä
- Useita kappaleita
- Linkkejä muille sivuille
- Eri tasoisia otsikoita
- Kuvia
- Lista

# HTML suunnitteluprosessi

- Ideointi
- Benchmarkkaus, eli olemassaolevien sivustojen tarkastelu
- Luonnostelu
- Rautalangat/Wireframes eli tietorakenteen ja navigaation kuvaus
- Leiskat/Layout eli graafisen ulkoasun hahmotelma

Haasteita:

- Mikäli graafinen suunnittelija päättää ulkoasun -> voi johtaa vaikeaan toteutettavuuteen. On ehkä tehokkaampaa muokata graafista ilmettä suoraan koodista.
- Sivustolle päätetty yleinen ilme tai käytettävät ulkoasuvalinnat

# Tehtävä - Kotivu-alasivu

- Tee kaksi HTML-tiedostoa, koti.html ja alasivu.html, laita niihin tekstiä, jotta näet kummalla sivulla olet
- Linkkaa sivut toisiinsa, että voit liikkua niiden välillä

# Tehtävä - Kotivu-alasivusto

- Tee useita HTML-tiedostoja, koti.html ja alasivuxxx.html, laita niihin tekstiä, jotta näet kummalla sivulla olet
- Linkkaa sivut toisiinsa, että voit liikkua niiden välillä
- Toteuta alasivuille meno listana
- Jokaisella alasivulla on samassa paikassa linkki kotisivulle.

# Tehtävä - Kotivu-alasivusto - navigointi ikonein

- Tee useita HTML-tiedostoja, koti.html ja alasivuxxx.html, laita niihin tekstiä, jotta näet kummalla sivulla olet
- Linkkaa sivut toisiinsa, että voit liikkua niiden välillä
- Toteuta linkit kuvina
- Toteuta alasivuille meno listana
- Jokaisella alasivulla on samassa paikassa linkki kotisivulle.



# HTML4 elementit

- Tyylittelyyn block tyyliset elementit kuten `<div>` ja inline tyyliset elementit kuten `<span>`
- `<div id="container">`  
  `<table>...`  
  ...  
  `</div>`
- `<p>My brother's name is <span class="namestyle">Michael</span><p>`
- `<h1>..<h6><p><b><br><table>...`
- Lista HTML4 elementeistä: <http://www.w3.org/TR/html4/index/elements.html>

# HTML5 uudet rakenteelliset elementit

# Miksi?

- Auttaa lähdekoodin lukijaa ymmärtämään, miten sivusto on rakennettu
- Voi auttaa myös toiminnallisuuden kehittämisessä

# HTML5 elementit - Ylätunniste <header>

- <header> sisältää joukon sivustoa kuvaavia elementtejä (nimi, logo, kuva) sekä mahdollisesti navigaation.

```
<header id="banner">
  <div id="logo">
    
  <div id="headline">OTSIKKO</div>
  <div id="headline2">ALIOTSIKKO</div>
  <nav>NAVIGAATIO</nav>
</header>
```

# Navigaatio <nav>

- <nav> elementissä esitetään sivuston päänavigaatio. Kaikkien sivun linkkien ei tarvitse olla nav-elementissä.
- Linkit voivat osoittaa muille sivuston sivuille, tai sivuun itseensä
- Päänavigaatio listamuodossa.

```
<nav>
  <ul>
    <li class="active"><a href="index.html">Etusivu</a>
    <li><a href="tuotteet.html">Tuotteet</a></li>
    <li><a href="jalleenmyyjät.html">Jälleenmyyjät</a></li>
  </ul>
</nav>
```

# Sisältö - <section> ja <article>

- Leipäteksti ja siihen liittyvät kuvat sijoitetaan sektioiden ja artikkelien sisään.
- Section-tagin yleisesti rakenteiselle sisällölle.
- Artikkelitagin käyttökelvoinen esimerkiksi blogiteksteissä tai itsenäisissä artikkeleissa
  - Määrällisesti monta, jäsennettävää tietoa
- Mahdollista tehdä hierarkisia rakenteita
  - Sektion alisektio.
- Sijoiteltavissa ja jäsennettävissä vapaasti CSS:n avulla.

# Alatunniste <footer>

- Tietoa sivustosta
  - Tekijät
  - Tekijänoikeudet
  - Linkki **rekisteriselosteeseen**
  - Käyttöehdot
- Alatunniste on yleensä sama kaikilla sivuston sivuilla
  - Yhtenäisyys
- Usein sivuston alalaidassa, mutta voidaan periaatteessa sijoittaa mihin tahansa.
  - Joka sivulla samassa kohdassa
  - Ei voi laittaa headerin tai toisen footerin sisälle.

# Sivupalkki <aside>

- <aside> -elementti kuvaa osan sivusta, jossa on välillisesti sivustoon liittyvää materiaalia.
- Tämän tyyppiset osiot esitetään usein sivupalkkeina painetussa materiaalissa.
- Esittää lohkoa joka voidaan esittää erotettuna pääsisällöstä

```
<aside id="sidebar">  
  <ul>  
    <li>Kirjaudu sisään</li>  
  </ul>  
</aside>
```



# <img>

- Oman tai verkossa olevan kuvan näyttäminen sivulla
- Oma kuva
  - ``
- Verkosta linkitetty kuva
  - ``

# Form ja submit

Emme tällä kurssilla käytä submitia, koska se olettaa palvelinpäähän sijoitettua toimintaa (esim. php.scriptiä, joka vie tiedon vaikka tietokantaan).

```
<form action="/action_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

# Input

```
<input type="text" name="lastname" value="Mouse"><br><br>
```

Voit kirjoittaa kenttään, mutta kirjoitetulle tekstille EI tapahdu mitään! (Tulee tapahtumaan vasta Javascriptin yhteydessä).

# Video

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  Your browser does not support the video tag.  
</video>
```

[https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5\\_video](https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_video)

# Meta viewport

Kertoo laitteen näytön koon. Helpottaa esim. mobiililaitteille suunnitteluss (liittyy enemmän CSS:n, mutta on HTML-tagi).

```
<meta name="viewport"  
content="width=device-width,  
initial-scale=1.0">
```

# Ulkopuolisen sisällön upottaminen

- `<iframe>`

- Esim Youtube

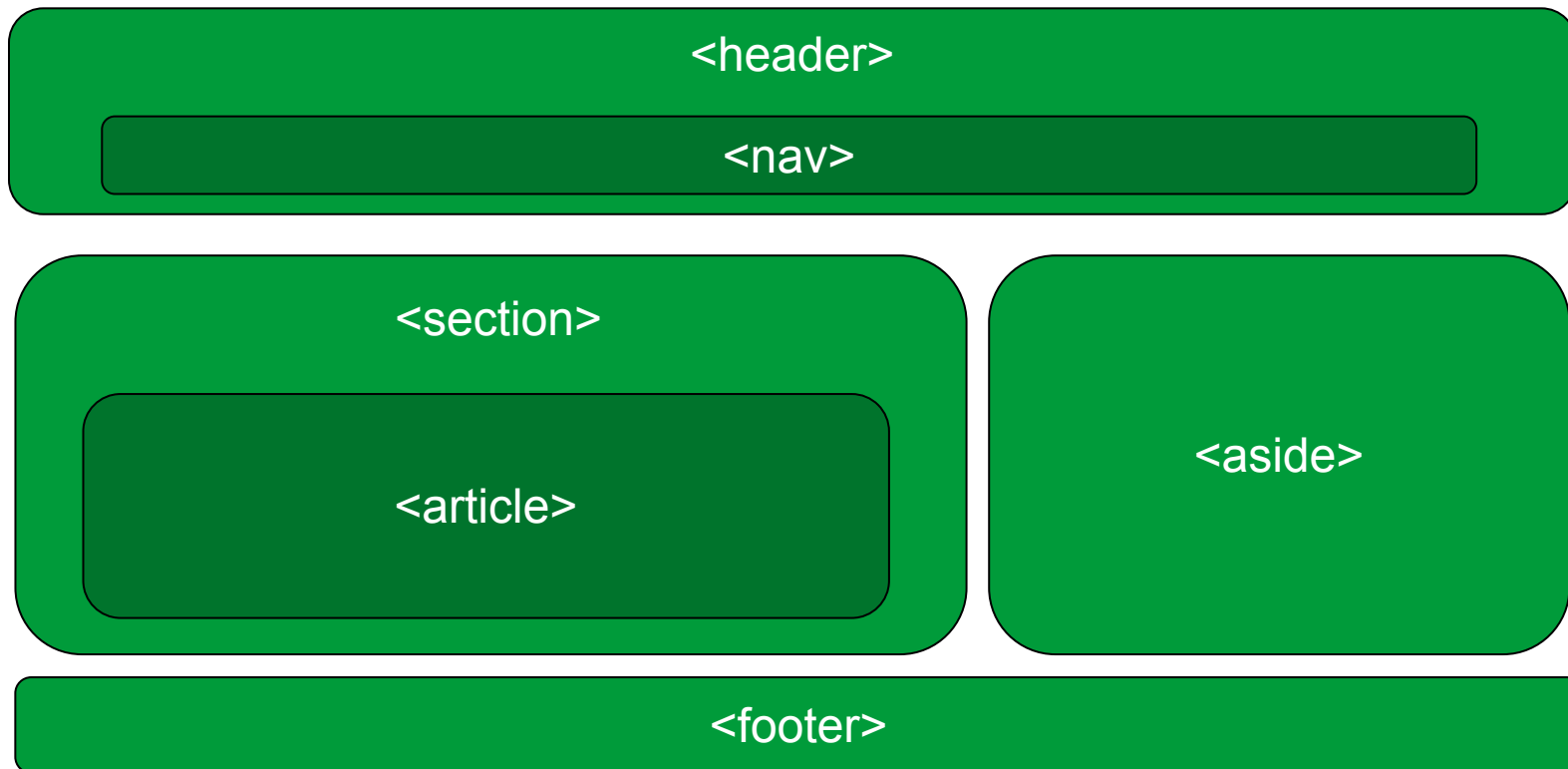
- ```
<iframe width="560" height="315" src="https://www.youtube.com/embed/NqM84D18Pqw"
frameborder="0" allowfullscreen></iframe>
```

- `<object>` ja `<embed>`

- Esim Youtube (vanha tyyli)

- ```
<object width="560" height="315">
  <param name="movie" value="https://www.youtube.com/v/NqM84D18Pqw?
  hl=en_GB&version=3"></param>
  <param name="allowFullScreen" value="true"></param>
  <param name="allowscriptaccess" value="always"></param>
  <embed src="https://www.youtube.com/v/NqM84D18Pqw?hl=en_GB&version=3"
  type="application/x-shockwave-flash" width="560" height="315" allowscriptaccess="always"
  allowfullscreen="true"></embed>
</object>
```

# Yksinkertainen verkkosivun rakenne HTML5:llä



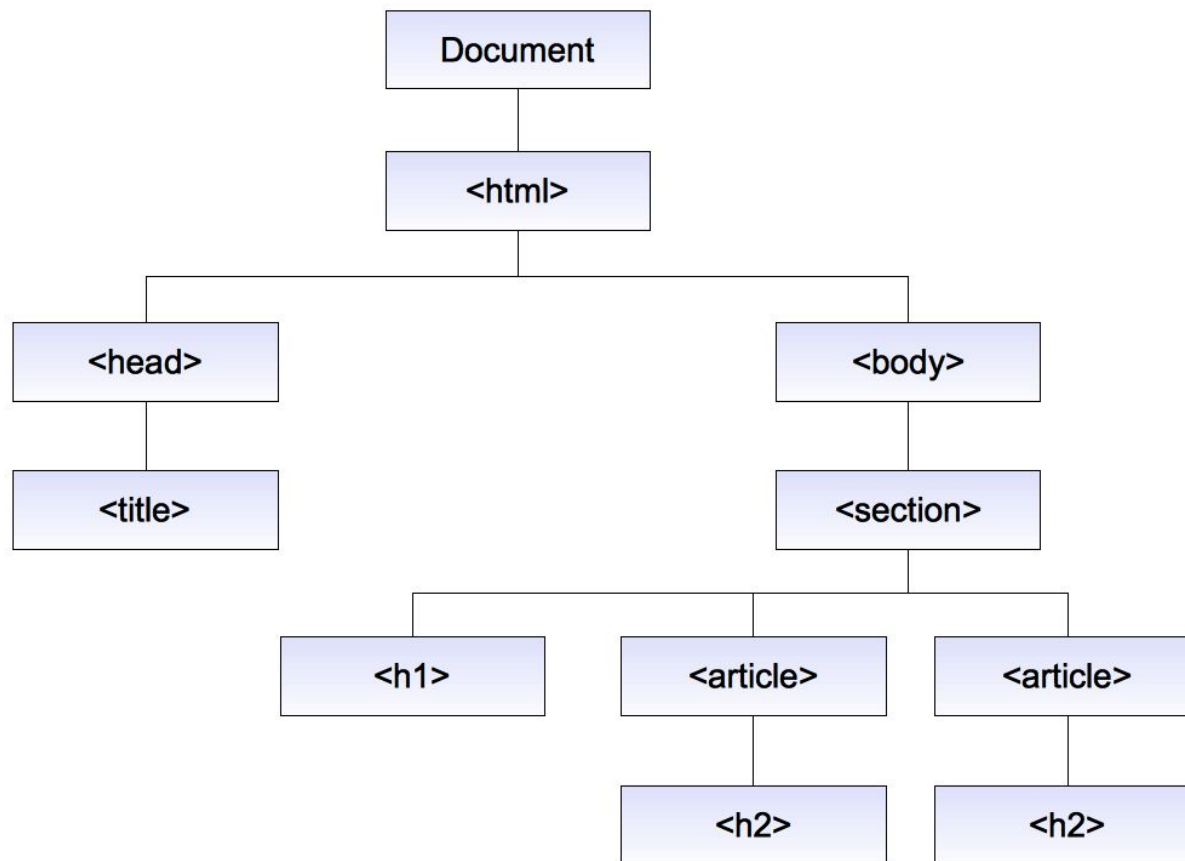
# DOM-puu



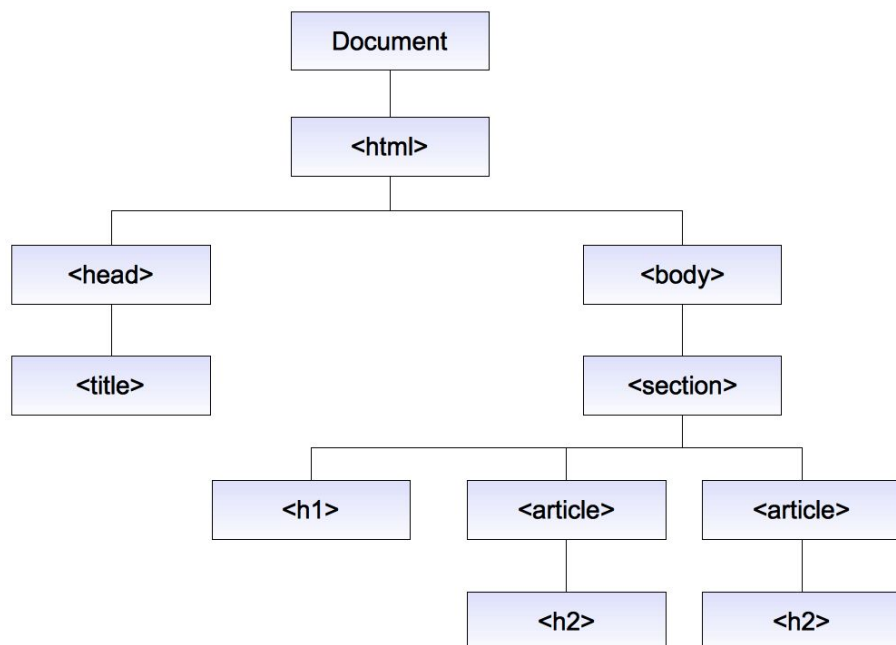
# Document Object Model (DOM)

- Tapa kuvata rakenteisen dokumentin, kuten HTML:n rakenne hierarkisena puuna
- Määrittelee miten dokumenttia hallitaan, käsitellään ja muokataan
- Ohjelmoijat voivat rakentaa dokumentteja, navigoida rakenteessa, poistaa tai lisätä elementtejä sekä muokata niitä.
- Voidaan käyttää minkä tahansa ohjelmointikielen kanssa. Tarkoittaa sitä, että esim. Javascriptillä voi luoda ja poistaa DOM-elementtejä 'lennossa'!

# DOM-puu - esimerkki

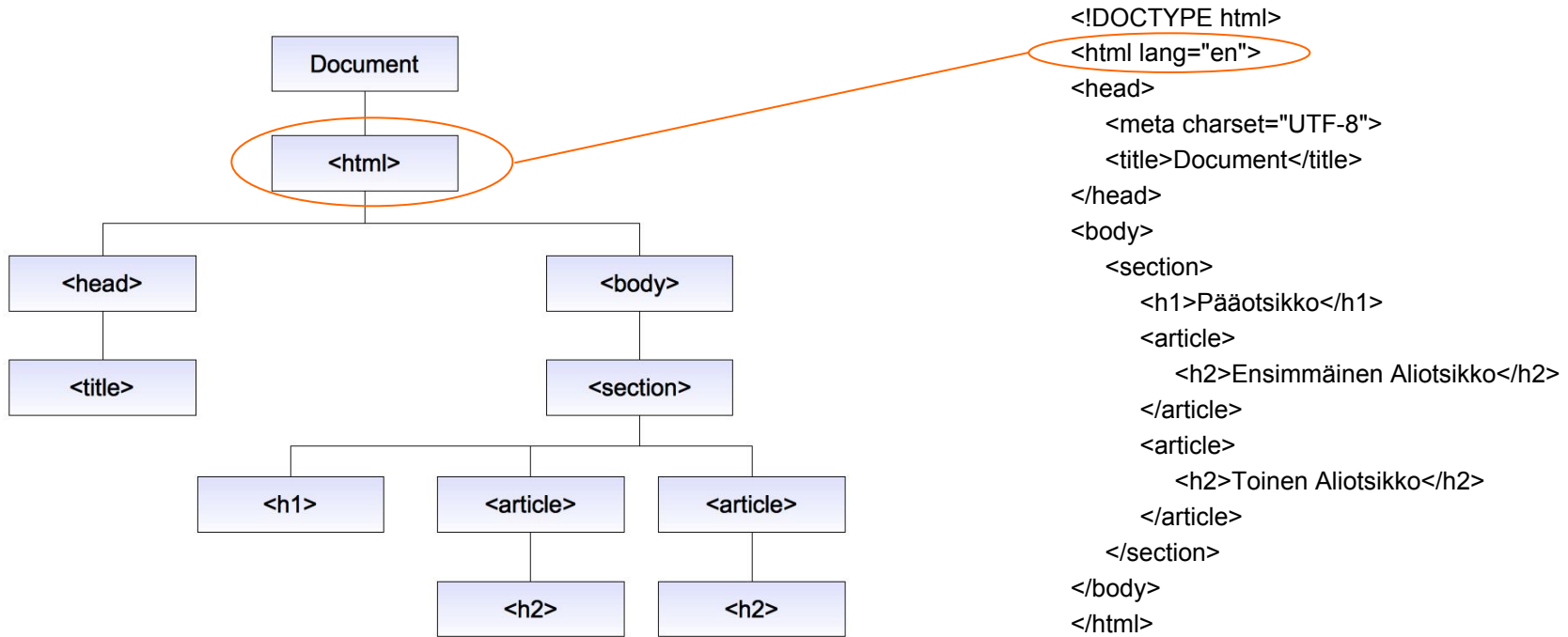


# DOM-puu - esimerkki

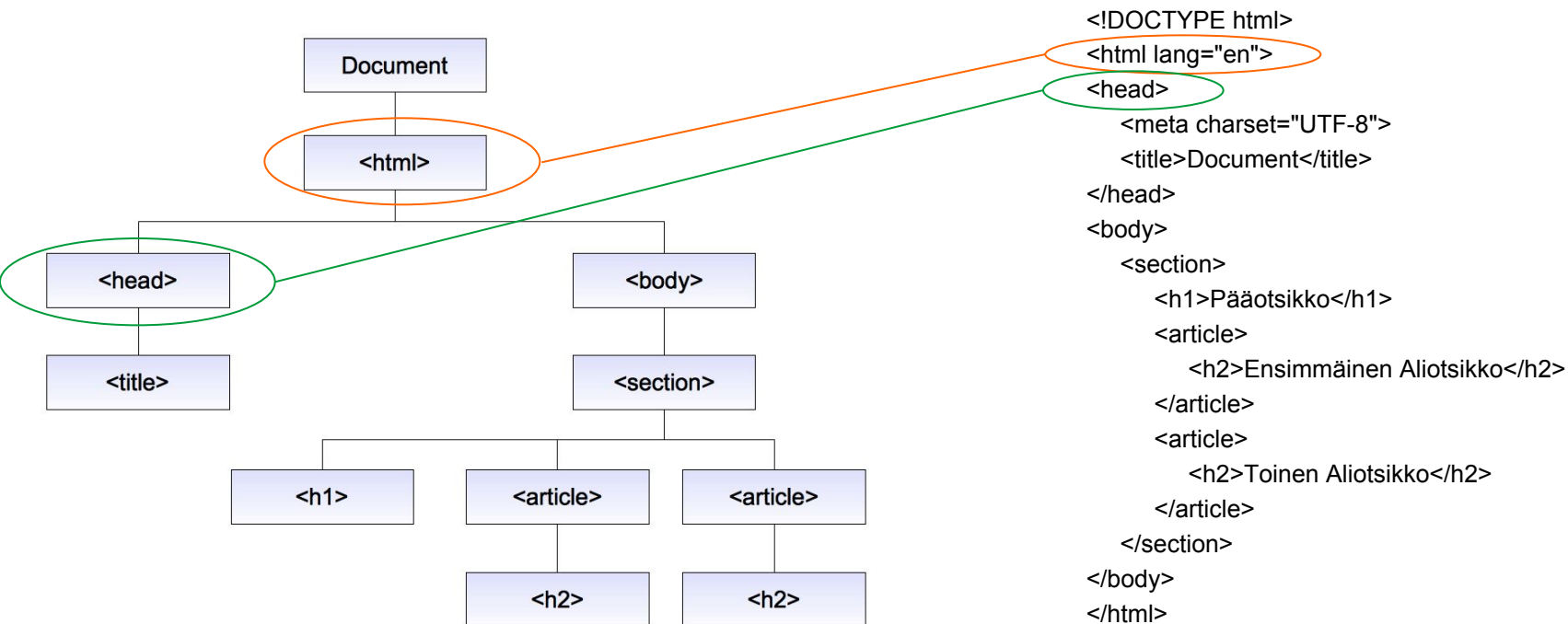


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <section>
    <h1>Pääotsikko</h1>
    <article>
      <h2>Ensimmäinen Aliotsikko</h2>
    </article>
    <article>
      <h2>Toinen Aliotsikko</h2>
    </article>
  </section>
</body>
</html>
```

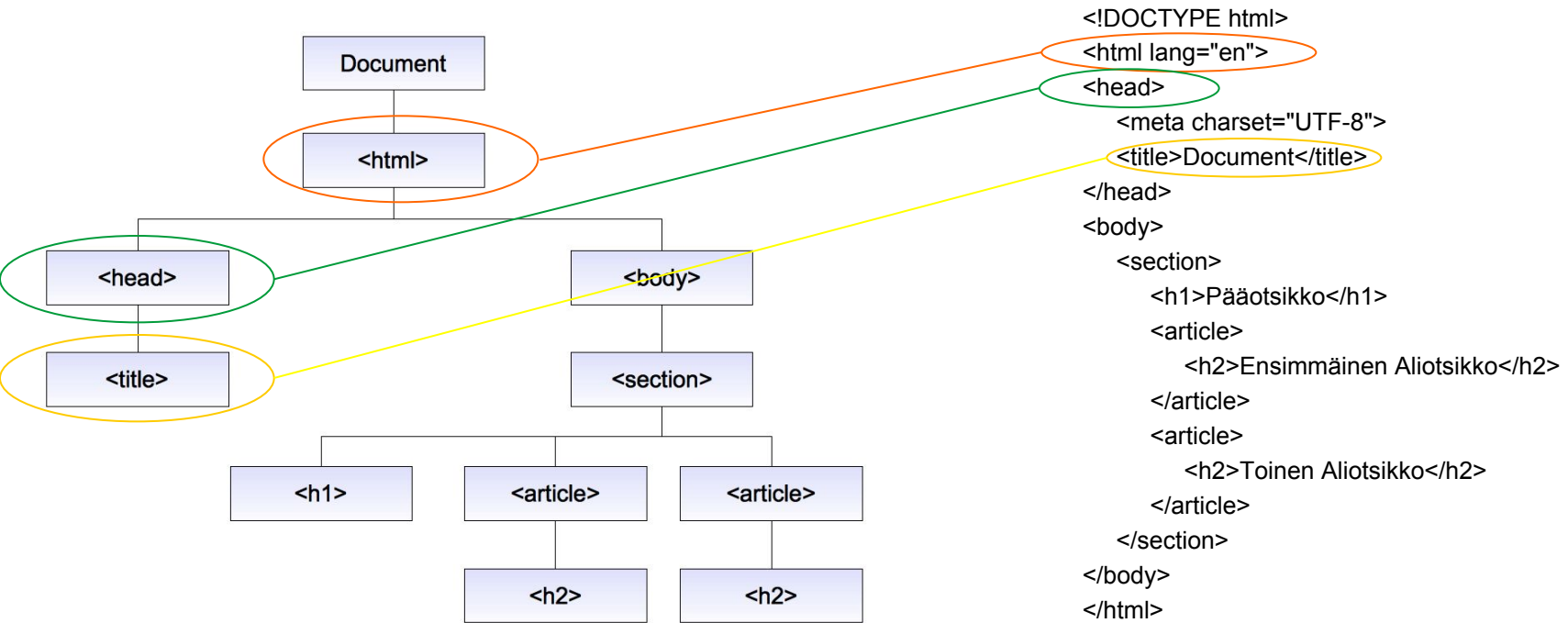
# DOM-puu - esimerkki



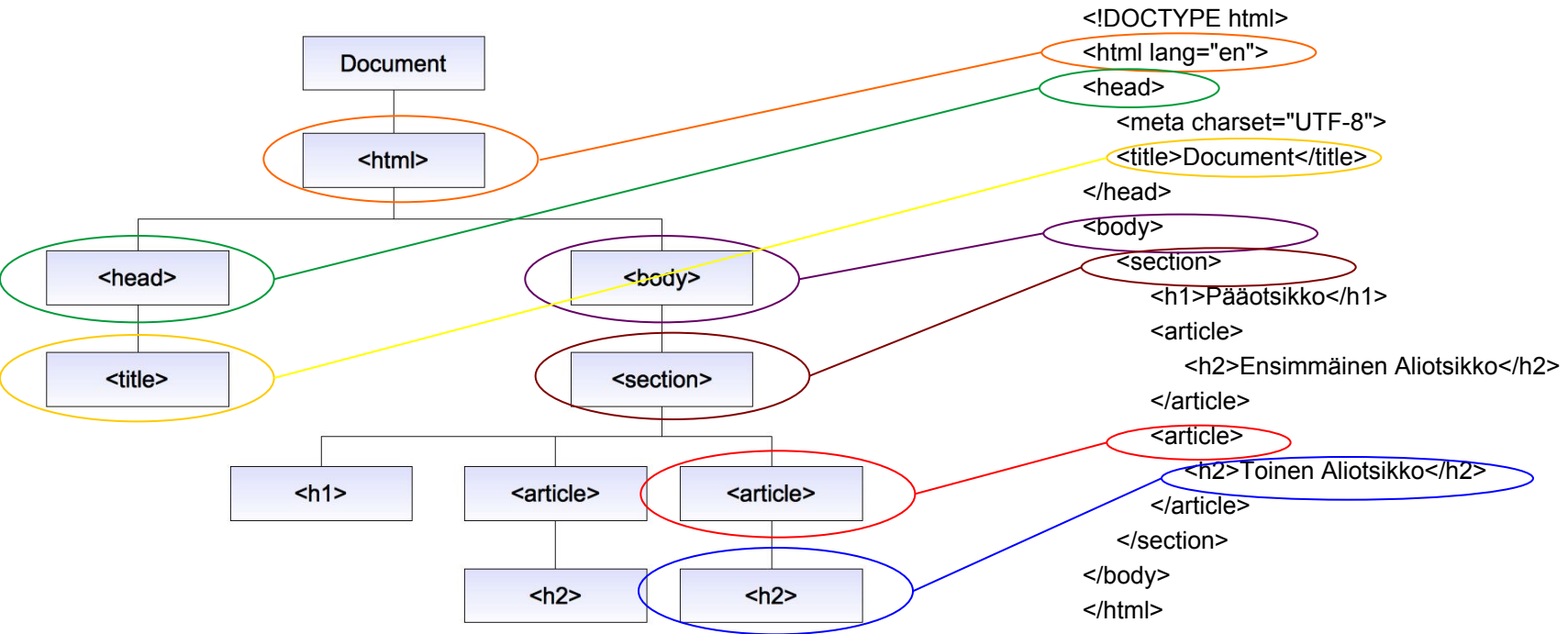
# DOM-puu - esimerkki



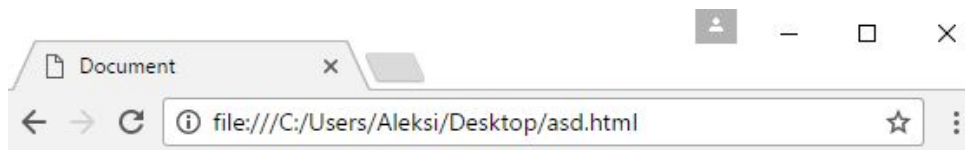
# DOM-puu - esimerkki



# DOM-puu - esimerkki



# DOM-puu - esimerkki



## Pääotsikko

### Ensimmäinen aliotsikko

### Toinen aliotsikko

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <section>
    <h1>Pääotsikko</h1>
    <article>
      <h2>Ensimmäinen Aliotsikko</h2>
    </article>
    <article>
      <h2>Toinen Aliotsikko</h2>
    </article>
  </section>
</body>
</html>
```



# Yhteenveto

- Verkkosivut
  - Hyvät käytänteet, editori/selain kehitysympäristö
- HTML
  - Historia ja nykypäivä pähkinänkuoressa
- Koodin rakenne ja käsittely
  - HTML5 elementit
  - DOM-puu
- Yksinkertaisen sivurungon rakentaminen
- Semantiikka
  - RDFa, mikrodata, mikroformaattit

# Lisämateriaalia – Cheat Sheets

- <http://www.cheatography.com/davechild/cheat-sheets/html-character-entities/>
- <http://www.addedbytes.com/download/html-cheat-sheet-v1/png/>
- [http://www.inmotionhosting.com/img/infographics/html5\\_cheat\\_sheet\\_tags.png](http://www.inmotionhosting.com/img/infographics/html5_cheat_sheet_tags.png)
- <http://woorkup.com/wp-content/uploads/2009/12/HTML5-Visual-Cheat-Sheet1.pdf>
- <http://media.smashingmagazine.com/wp-content/uploads/images/html5-cheat-sheet/html5-cheat-sheet.pdf>

# Tuntitehtävien luettelo

- Sivuston suunnittelu paperilla
- Tutustuminen kehitystyökaluihin ja kokeilu (ikkunat, lähdekoodin katselu, kehitystyökalut)
- Yksinkertainen linkittämätön sivu, jolla erilaisia elementtejä
- Alasivulle linkattu kotisivu
- Kokonainen sivusto
- Kokonainen sivusto, jossa linkit toteutettu ikonein

# Kiitos!

**Kysymyksiä? Kommentteja?**

pauli.laine@aalto.fi