



PROGRAMMERING FÖR UX-PRODUKTION 02

HTML Grund

Förra gången

- Alla presenterar sig
- Kursen presenteras
- Hur fungerar:
 - Webbbläsare
 - Webserver
 - HTTP
- En sajts beståndsdelar
 - HTML, CSS, Javascript
- 12 Regler + grupparbete

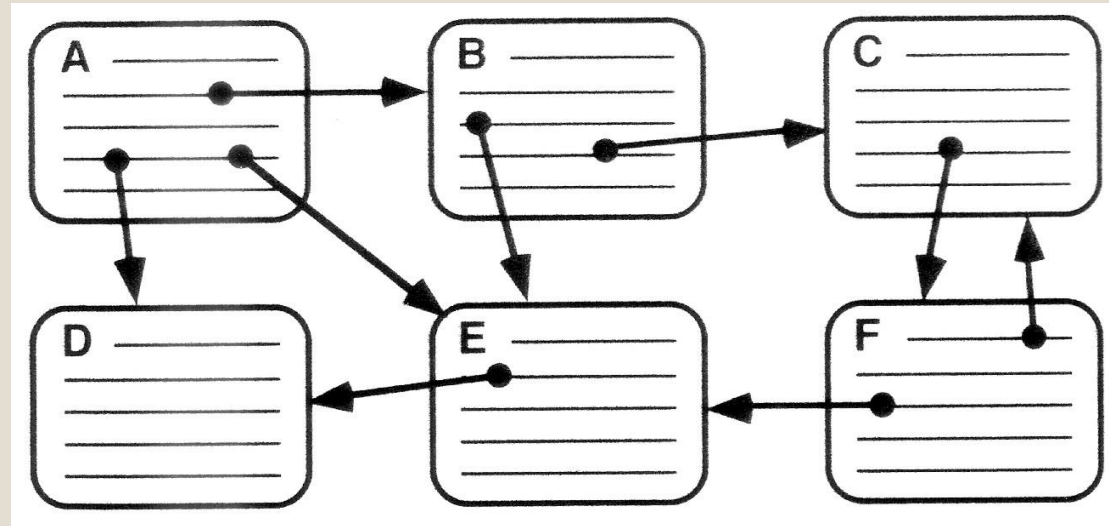
Idag

- “Redovisning” av 12 regler
- HTML - Hur fungerar det
 - HTML-dokumentets struktur
 - Vanligaste HTML-elementen
- Uppgift 1 - Webb sida - Profilsidan

12 Regler - Reflektioner

HTML

- Hypertext markup language
 - Hypertext = text med länkar till andra texter
 - Markup = markering för att strukturera, organisera och tydliggöra element på en webbsida
 - Language = språk
- HTML är ett markeringsspråk som webbläsare använder för att tolka och skriva text, bilder och annat material, till visuella eller hörbara webbsidor.
- Första versionen kom 1991 och bestod av 18 element.
-



Hyper Text

=länkar!

Webbplatsens tre delar



Innehåll



Design



Funktion

Webbsidans tre delar

- **HTML** (Hyper Text Markup Language) beskriver en sidas innehåll (content) och struktur.
"De här elementen ska finnas på sidan, i den här ordningen"
- **CSS** beskriver hur taggar/element ska presenteras.
"Det här elementet ska ha röd textfärg och vara understruket."
- **JavaScript** hanterar användarens interaktion med sidan.
"När jag klickar på knappen ska det här hända."
- **Separation of concerns** innebär att man inte vill blanda HTML, CSS och JavaScript utan ha dem i separata filer. Filerna slutar normalt med .html, .css och .js



HTML – Hypertext Markup Language

- En webbsida består av en textfil i HTML-format, och andra länkade filer.
- Filnamnet slutar ofta på .html. Tex "index.html"
- Innehåller taggar/element:
 - `<tag>Någon form av innehåll</tag>`
 - Exempel: `En viktig text`
- En tag utan innehåll skrivs `<tag/>` (om taggen inte har något innehåll)
 - Exempel: `<hr/>` (horisontal ruler, eller linje)
 - En tag kan ha ett attribut: `<tag attribut="värde">Innehåll</tag>`
 - `<h1>Rubrik på sidan</h1>`
 - Exempel: ``
- Mellanslag i HTML ignoreras(förutom det första)
 - Exempel: `<p> En text </p>` skrivs ut som " En text".



Ett element

Börja
markering

Avsluta
markering

`<h1>` Farmor Britts pannkakor `</h1>`

De flesta taggar kommer i **par**
Undantag är ex. *img*

HTML - dokumentet

- `<!DOCTYPE html>`
 - Inleder ett HTML-dokument
- `html`
 - `<html>` och `</html>` Anges i början och slutet av hela dokumentet
- `head`
 - `<head>` och `</head>` Används för information om sidan (metadata)
 - Inuti head-taggen anges t ex sidans title
 - `<title>Sidans titel</title>`
- `body`
 - `<body>` och `</body>` Anges före och efter sidans synliga innehåll



Grundelement

Dokumenttyp

```
<!DOCTYPE html>
```

Här börjar det

```
<html>
```

Instruktioner till webbläsaren

```
<head>
```

Teckenuppsättning

```
<meta charset="UTF-8">
```

Sidans titel (syns överst i webbläsare)

```
<title> Hello World! </title>
```

```
</head>
```

Här börjar innehållet

```
<body>
```

...innehållet som kommer synas på
webbsidan...

```
</body>
```

```
</html>
```

HTML-dokumentet i sin helhet

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sidans titel</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <p>En text om något intressant</p>
  </body>
</html>
```

Filnamnet på första sidan av en sajt ska vara: **index.html**

Eller snarare såhär...

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sidans namn, t ex Om oss eller Start</title>
  </head>
  <body>
    ...Innehållet
  </body>
</html>
```

Meta-taggar

```
<meta charset="UTF-8">
```

Teckenuppsättning. Stödjer t ex svenska tecken ÅÄÖ

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

Säger till Microsoft Internet Explorer att använda den senaste render-motorn, Edge

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

width=device-width anger sidans bredd så att den följer enhetens skärmbredd (som varierar beroende på enheten).

initial-scale=1.0 anger den första zoomnivån när sidan först läses in av webbläsaren.

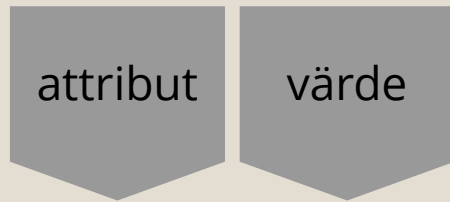
Farmor Britts pannkakor

Ett element är nästan alltid ett block som tar upp hela bredden av sin *container*.

Undantag är ex. *img*,
det är ett s.k. inline-element

Attribut

Ger en tagg fler egenskaper



`<h1 class="large"> Farmor Britts pannkakor </h1>`

Olika taggar kan ha olika attribut
ex. `img` har `src`

HTML-taggar

- h1 - h6 Rubriker (h = heading)
- p Textparagraf
- a Länkar, a står för "anchor". Attribut: href
- strong, b, i, u Olika sätt att betona text
- br, hr Radbrytning och linje
- **div**, span Taggar för formattering och layout
- img Bilder. Attribut: src
- **ul**, **ol**, **li** Listor
- table, tr, td Tabell för data
- br Radbrytning

Bild



Huvudrubrik

Farmor Britts pannkakor

Brödtext

Gör traditionella tunna pannkakor genom att blanda mjöl, mjölk och ägg och lite salt till en jämn smet. Stek smeten till tunna och smarriga pannkakor och servera med en söt sylt.

Underrubrik

Ingredienser

- 2 1/2 dl vetemjöl
- 1/2 tsk salt
- 6 dl mjölk
- 3 ägg
- 3 msk smör
- sylt eller bär

Gör så här:

1. Blanda mjöl och salt i en bunke. Vispa i hälften av mjölken och vispa till en slät smet. Vispa i resten av mjölken och äggen.
2. Smält smöret i stekpannan och vispa ner i smeten. Stek tunna pannkakor av smeten i en stek- eller pannkakspanna.
3. Servera med sylt eller bär.

Punktlista

Numrerad lista



<h1>

Farmor Britts pannkakor

<p>

Gör traditionella tunna pannkakor genom att blanda mjöl, mjölk och ägg och lite salt till en jämn smet. Stek smeten till tunna och smarriga pannkakor och servera med en söt sylt.

<h2>

Ingredienser

- 2 1/2 dl vetemjöl
- 1/2 tsk salt
- 6 dl mjölk
- 3 ägg
- 3 msk smör
- sylt eller bär

Gör så här:

1. Blanda mjöl och salt i en bunke. Vispa i hälften av mjölken och vispa till en slät smet. Vispa i resten av mjölken och äggen.
2. Smält smöret i stekpannan och vispa ner i smeten. Stek tunna pannkakor av smeten i en stek- eller pannkakspanna.
3. Servera med sylt eller bär.

Heading

- I HTML finns det fördefinierade rubriktaggar.
- Dessa är h1-h6.
- Exempel:
 - `<h1>En stor rubrik</h1>`
 - `<h4>En underrubrik</h4>`
- Med h-taggen får man ett par egenskaper på köpet:
 - Större text
 - En radbrytning efteråt

Paragraph

- Precis som i en bok eller tidning delar man ofta upp text i stycken, eller paragrafer.
- Det är lämpligt att även göra detta på en hemsida.
- Exempel:
 - `<p>En text om vad som helst, och som har lite längre innehåll...</p>`

Anchor

- Anchor, eller länk är en viktig del av en webbsida. Det gör att man kan klicka på t ex en text och transporteras till en ny sida.
- Exempel:
 - `Klicka för att komma till en annan sida`
 - `Klicka här för att läsa nyheter`

Markera text

- Fet stil, kursiv eller understruken text hjälper till att förtydliga ett textavsnitt.
- Strong eller b: Fet stil (bold)
 - Exempel
 - Det är `viktigt` att använda `handsprit` på händerna.
 - Det är **viktigt** att använda **handsprit** på händerna
- i – italic, kursiv text.
 - Exempel: Fartyget `<i>Titanic</i>` sjönk 1912.
 - Fartyget *Titanic* sjönk 1912.
- u – underline
 - `<u>Glöm inte</u>` att deklarerera.
 - Glöm inte att deklarerera.

Horisontal ruler

- Ett enkelt sätt att avdela olika delar av en webbsida är med en linje.
- HTML-tag: `<hr>`

Division

- En av de absolut viktigaste taggen är `<div>`
- Den används för att rama in innehåll, för att sedan påverka utseendet med hjälp av CSS.
- Den används också som behållare för HTML som skrivs av Javascript.
- Exempel:
 - `<div>Hejsan svejsan</div>`
 - `<div class="design">Hejsan svejsan</div>`
 - `<div id="content"></div>`
- Block-objekt.
 - `<div>Det var en gång en gubbe</div><div> som satt på en stubbe</div>`
 - Det var en gång en gubbe
 - som satt på en stubbe

Span

- Används för text, men fungerar på liknande sätt som div.
- Inline-objekt.
 - `Det var en gång en gubbe som satt på en stubbe`
 - Det var en gång en gubbe som satt på en stubbe

Image

- För att infoga bilder, så används taggen img.
- Exempel:
 - ``
 - ``

Lists

- Listor i punktform.
- ul – Unordered list
- ol - Ordered list (nummererad)
- li – List Item

Exempel:

- ``
 - `Java`
 - `C#`
 - `Python`
- ``

Tables

- En tabell är lämplig att använda om man vill visa en lista, eller tabell på en webbsida
- table – Anger att en tabell ska byggas
- tr – Table row
- td – Table data
- Th – table header
- Exempel:

```
<table>
  <tr>
    <th>Sensor</th>
    <th>Mätvärde</th>
  </tr>
  <tr>
    <td>Temp</td>
    <td>8</td>
  </tr>
  <tr>
    <td>Hum</td>
    <td>84</td>
  </tr>
</table>
```

Line break

- `Hej`
- `Hopp`
- ...blir HejHopp på webbsidan.
- För att göra en radbrytning måste taggen `br` användas.
- `Hej
Hopp`
- ...blir
- Hej
- Hopp

Flöde (Flow)

Elementens ordning

bubble

<header>

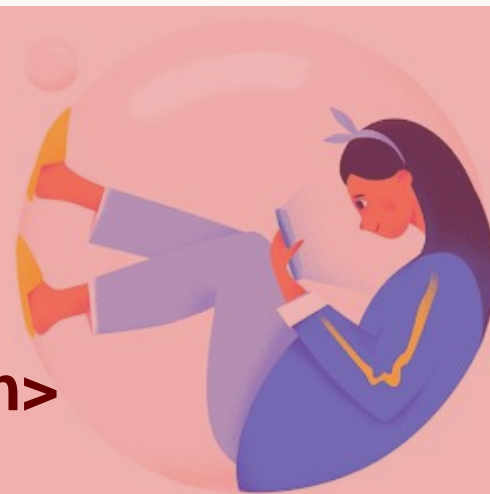
[Support](#)

[Log In](#)

[Sign Up](#)

A fresh **social**
networking
environment

<main>



Be creative and express your thoughts through photos and videos. Never miss out and discover what's happening with the people you follow.

Download on the
App Store

Get it on
Google Play

By Outcrowd

Struktur-element

Block-element som märker upp struktur, inte innehåll

Fungerar tekniskt på samma sätt som en DIV, dock semantisk skillnad

Vanliga element

Typografi

h1-h3

p

em

Struktur

main

nav

section

article

header

footer

aside

div

Media

img

video

audio

Övrigt

a (länk)

Semantik

Rätt tagg på rätt plats.

Viktigt för tillgängligheten (accessibility)

ex. Formulärbeteenden desktop / mobil / screen reader

Semantiska element

<article>

<main>

<aside>

<mark>

<details>

<nav>

<figcaption>

<section>

<figure>

<summary>

<footer>

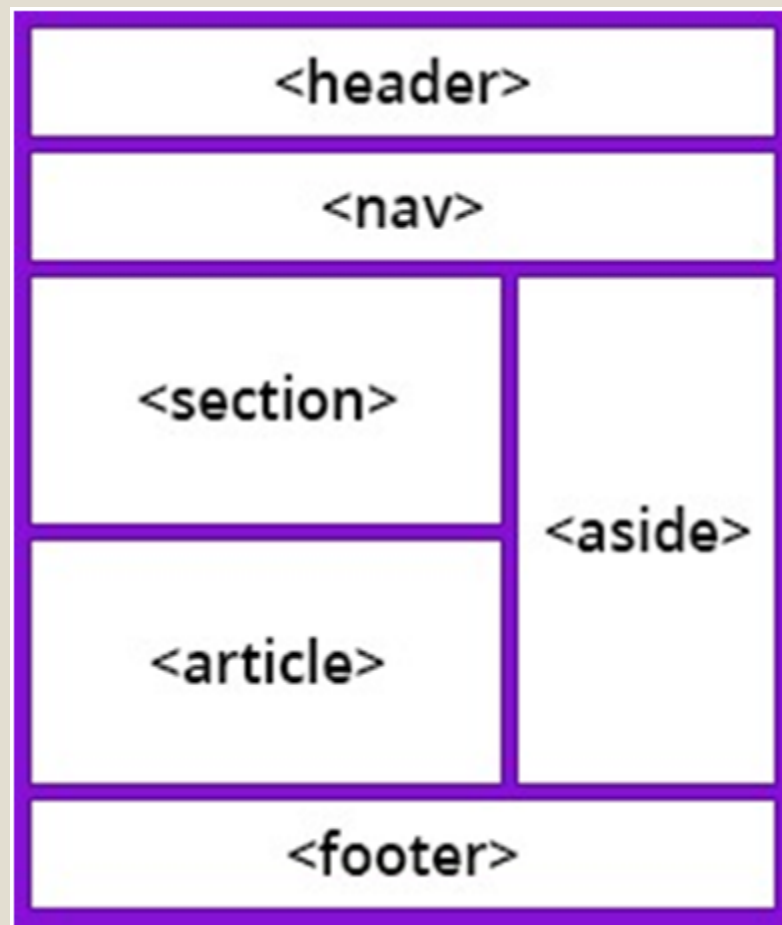
<time>

<header>

+100 till typ...

Struktur-elementens semantik

ehh...är det en *section* eller en *article*?



bubble

<header>

Support

Log In

Sign Up

A fresh **social
networking**
environment

<section>



<section>

Be creative and express your thoughts through photos and videos. Never miss out and discover what's happening with the people you follow.



Download on the
App Store



Get it on
Google Play



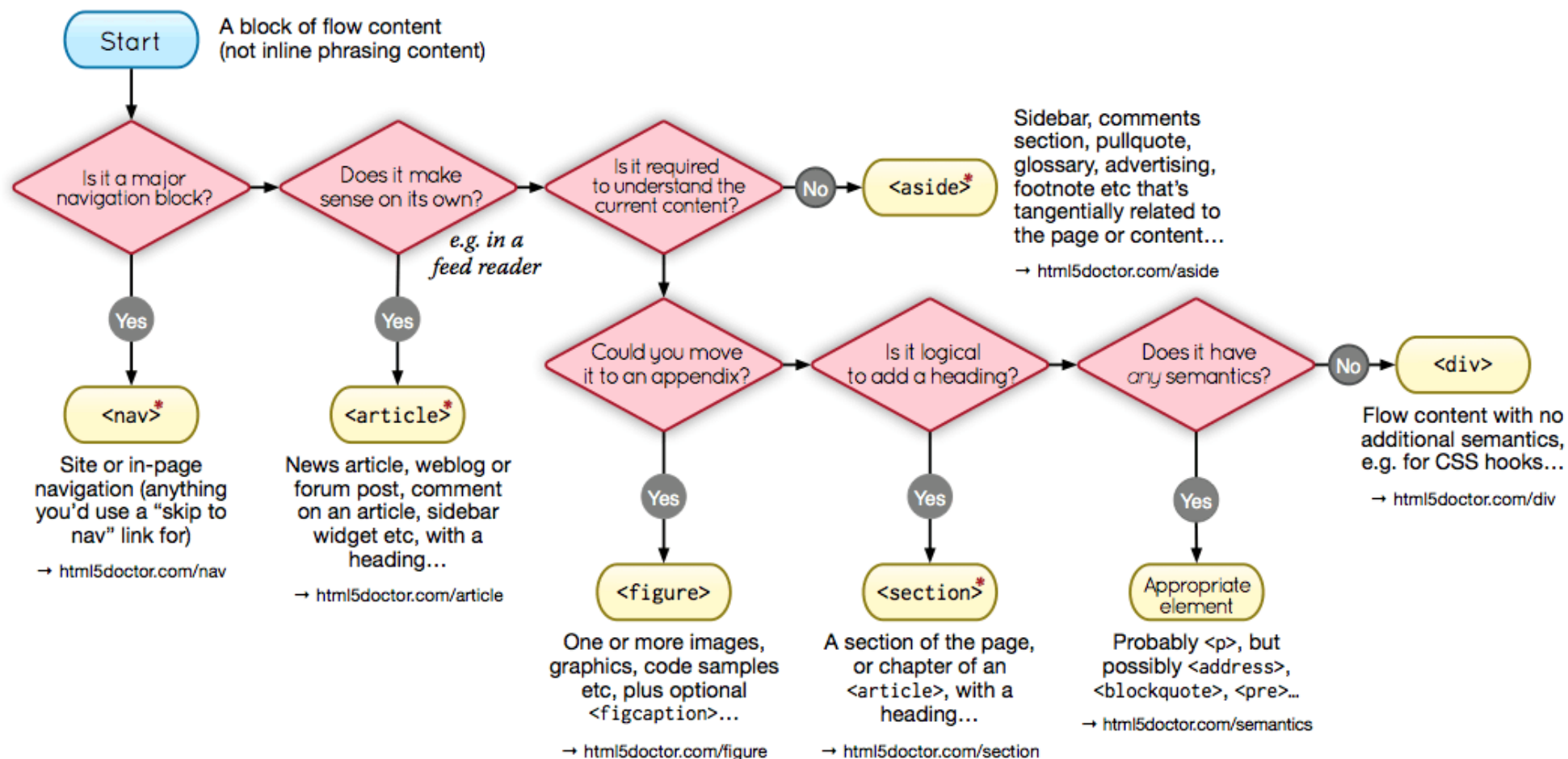
Doctor

HTML5 Element Flowchart

Sectioning content elements and friends

By @riddle & @boblet

www.html5doctor.com



*Sectioning content element

These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline
→ html5doctor.com/outline

2011-07-22 v1.5

For more information:
www.html5doctor.com/semantics

Exempel

```
◦ <header>
◦   Logotyp och företagsnamn
◦ </header>
◦ <nav>
◦   Menyn
◦ </nav>
◦ <main>
◦   <article>
◦     En massa information
◦   </article>
◦ </main>
◦ <footer>
◦   Kontaktinformation
◦ </footer>
```

Så lite HTML som möjligt
utan att kvalitet och resultat kompromissas

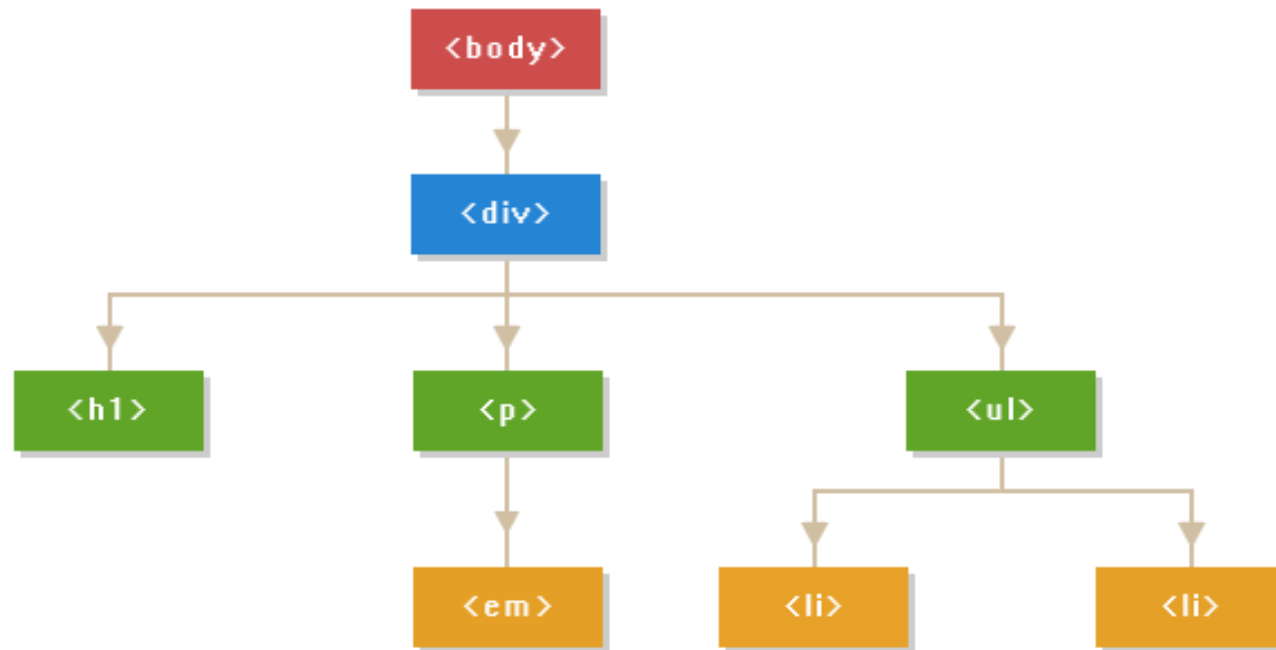
Document Object Model

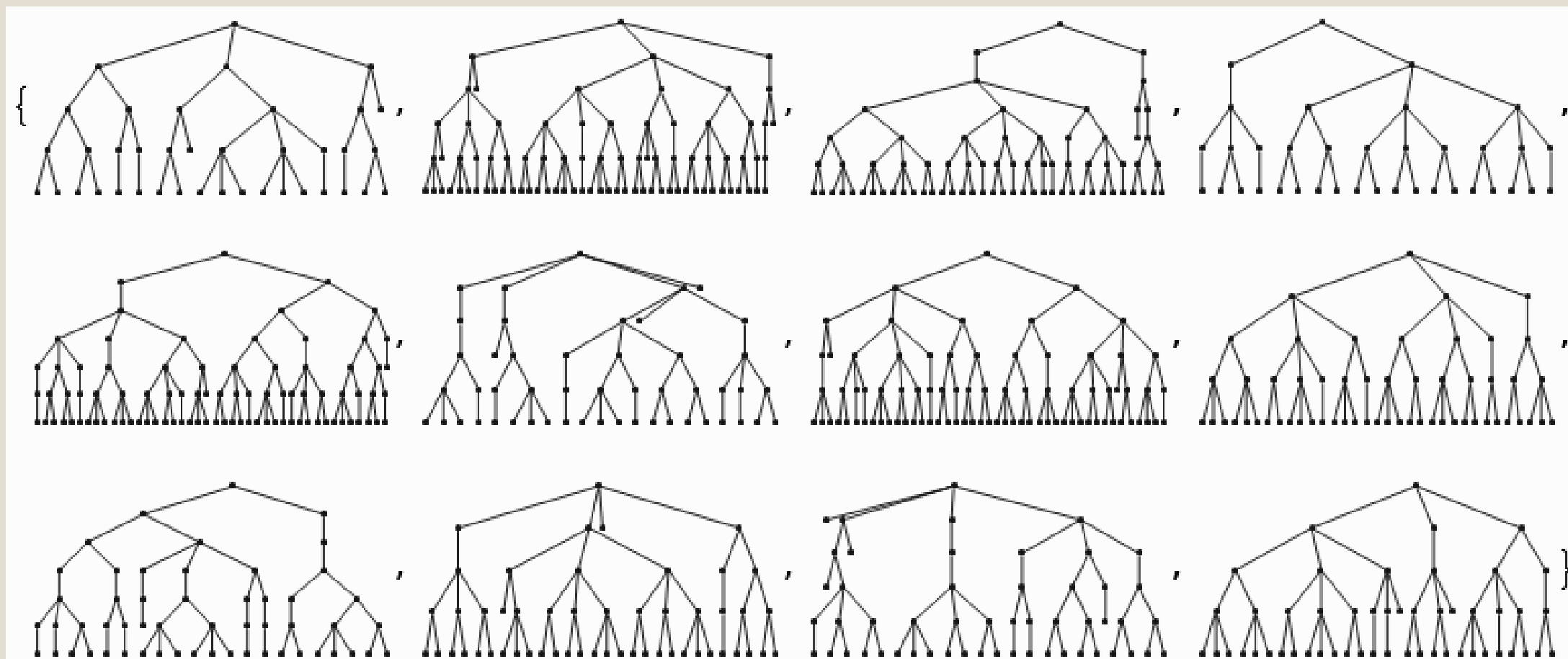
Kärt barn, många namn

- Dommen
- DOM-trädet
- HTML-trädet

DOMmen

Alla element bildar en trädstruktur





DOM-trädet varierar mycket i struktur

Uppgift 1 - Profilsajt

- I projekt 1: Profil, så ska ni skapa en html sida som ska se ut som en profilsida, fylld med information som är relevant för just en profilsida (vad eller vems profil väljer ni själva).
- Kraven för att klara uppgiften:
 - Minst en bild
 - Använd er av headers och paragrafer
 - Minst en lista
 - Minst en tabell, med t ex data om den profil ni valt: Längd, ålder osv
 - Minst 2 länkar till en utomstående sida (t.ex. google.com)
 - Minst två sidor, som ska gå att klicka sig emellan, t ex Huvudsida om profilen, samt en extra sida om personens påhittade (eller verkliga) CV/meriter, i en lista.
 - Använda symantisk HTML (HTML5) (header, nav, main, article, footer, istället för DIV)
 - Design och layout med hjälp av CSS (fredagens lektion)
- Enskild uppgift, men okej att samarbeta i grupprum

Kod-exempel - HTML

- För html-sidor använder vi med fördel Visual Studio Code.
- Kod: index.html

Resten av dagen

- Börja jobba med uppgift 1, och fyll på element, men vänta med design tills imorgon.