

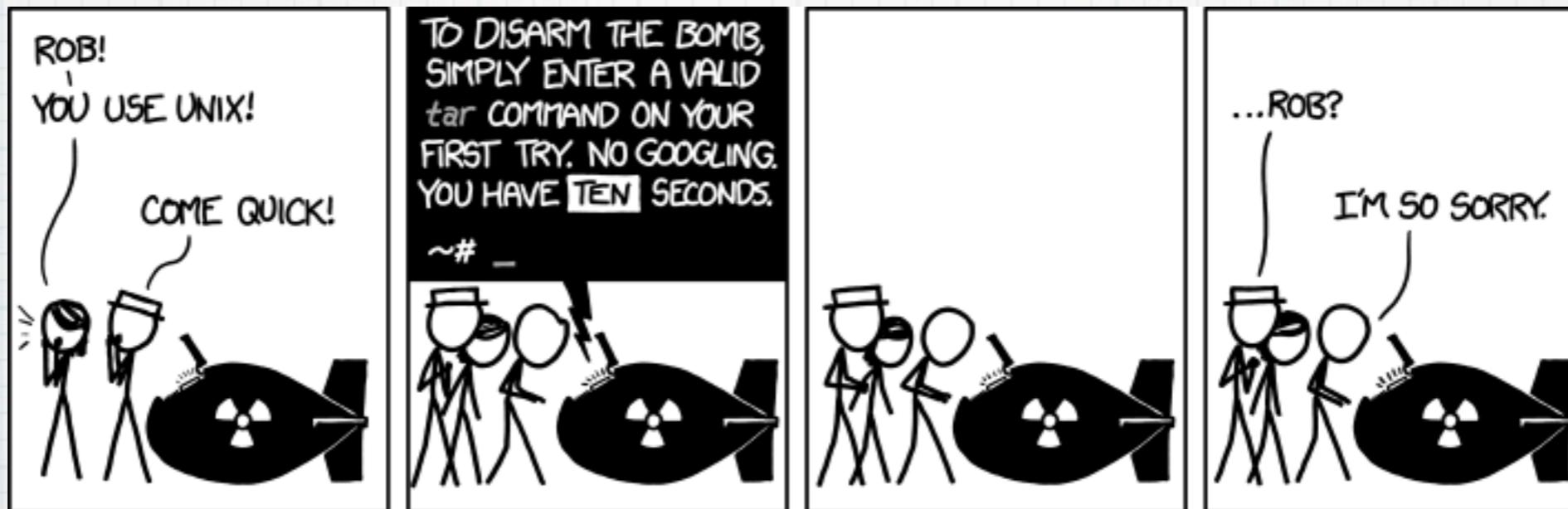
HTML 1

Forelesning 2, Webprosjekt H2013

Alfred Bratterud

Litt XKCD

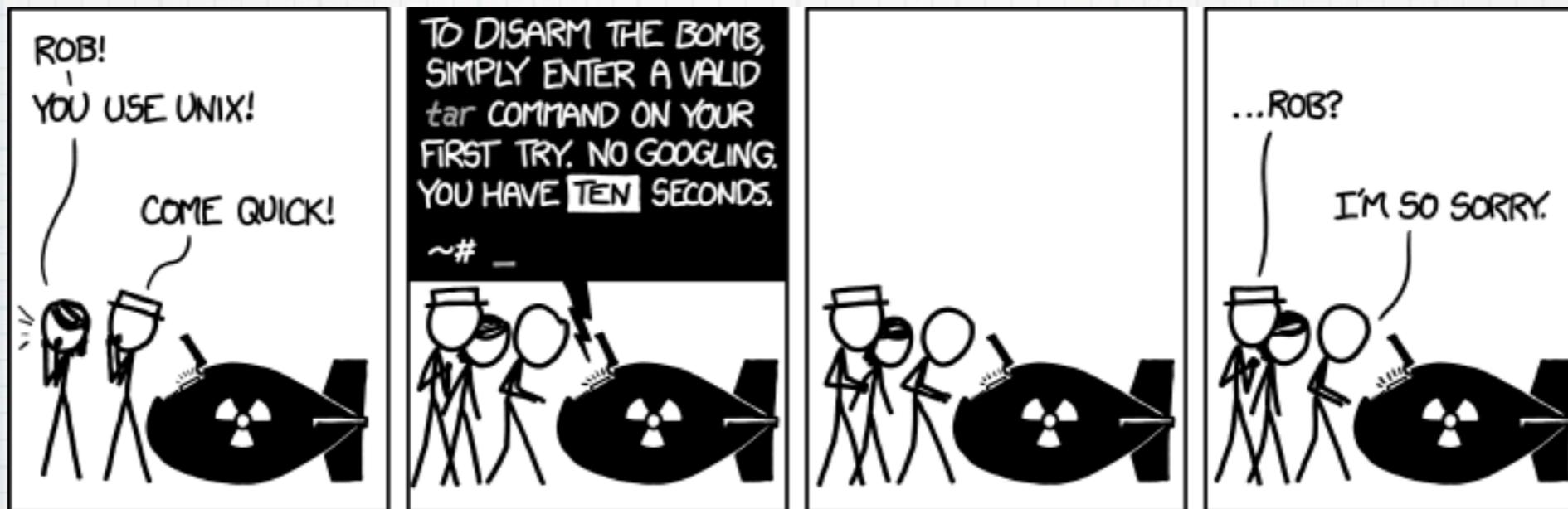
Tar



Permanent link to this comic: <http://xkcd.com/1168/>

Litt XKCD

Tar

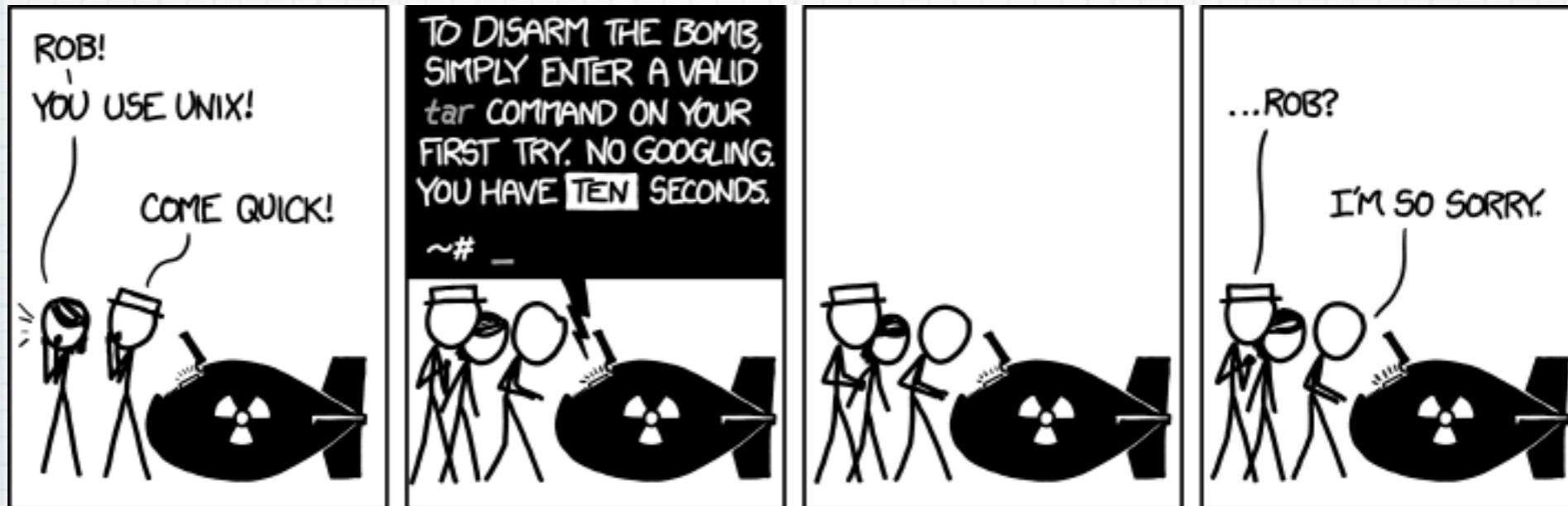


Permanent link to this comic: <http://xkcd.com/1168/>

Mitt første forsøk:

Litt XKCD

Tar



Permanent link to this comic: <http://xkcd.com/1168/>

Mitt første forsøk:

~ alfreb\$ tar --version

Plan for dagen

- * Litt repetisjon
- * Innføre noen flere tagger
- * Validering
- * Lage en enkel, men skikkelig hjemmeside
- * Tegnsett og enkoding (hvis tid)

Hvordan lager man en nettside?

Flere ulike strategier:

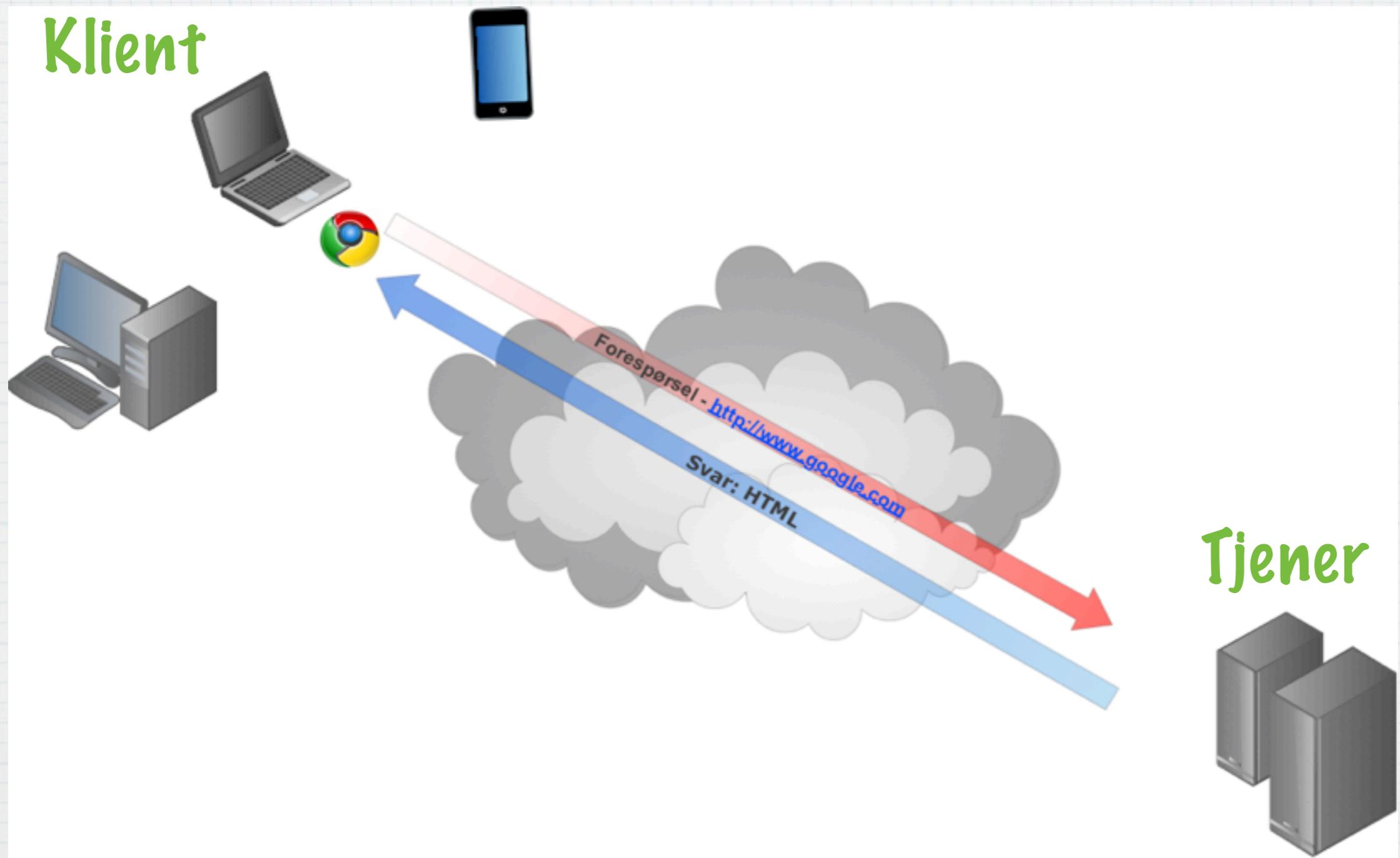
1. Lage alt fra bunnen av
2. Tilpasse og bruke en ferdig løsning
3. Lage hovedstrukturen fra bunnen av, men benytte ferdige komponenter

Bra sted å starte!

Hva er en nettside?

- * HTML-kode
 - * Definerer sidens struktur
- * CSS-kode (Cascading Stylesheets)
 - * definerer HTML-elementenes utseende
- * Grafikk; forgrunns- og bakgrunnsbilder
- * Multimediainnhold

HyperText Transfer Protocol



Nettlesere

- * For å vise en nettside bruker vi en nettleser.
- * Henter innhold fra webtjener
- * Genererer grafisk layout, basert på HTML og CSS
- * Siden det finnes mange nettlesere trenger vi standarder

Standarder

- * Standarder for nettlesere defineres bla. av W3C (World Wide Web Consortium)
- * finnes på <http://www.w3.org>
- * Standarder er viktig, men ikke alltid mulig å tvinge igjennom
- * Vi forholder oss til HTML5 - slik den defineres i validatoren.

Verktøy

- * Windows:

- * TextPad

- * NotePad

- * Mac: (men kun med syntaksmerking)

- * TextMate

- * BBEdit /
TextWrangler

- * Linux:

- * Kate

- * Gedit

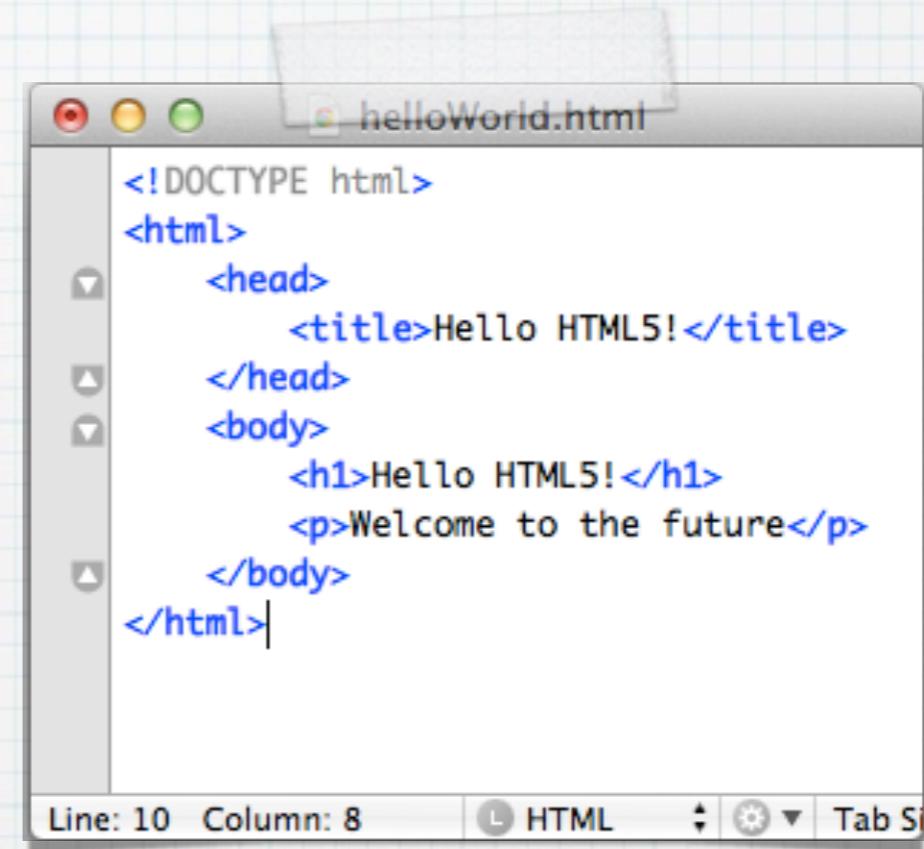
- * emacs

- * Generelle:

- * jEdit

HTML-kode

- * HTML-filer er tekstfiler, med HTML-tagger
- * Tagger starter med <tagnavn> og avsluttes med </tagnavn>
- * <!DOCTYPE ...> angir HTML-versjon



The screenshot shows a code editor window titled "helloWorld.html". The code in the editor is:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML5!</title>
  </head>
  <body>
    <h1>Hello HTML5!</h1>
    <p>Welcome to the future</p>
  </body>
</html>
```

The status bar at the bottom of the editor shows "Line: 10 Column: 8".

HTML-tagger

- * **<html>** Alt pakkes inn i html-taggen
</html>
- * **<head>**
Dette er “hodet”. Her ligger informasjon browseren trenger før “rendering”
</head>
- * **<body>**
Dette er “Kroppen”. Her ligger alt innholdet. **</body>**

Flere tagger

- * `` (emphasis) / `<i>`
- * `` (strong emphasis) / ``
- * ~~``~~
- * `<sub>` (subscript)
- * `<sup>` (superscript)
- * `<meta>`
- * `<audio>`, `<video>`
- * `<div>`
- * `<canvas>`, `<time>`
- * `<header>`, `<footer>` og `<section>`

Flere tagger

- * `` (emphasis) / ~~`<i>`~~
- * `` (strong emphasis) / ~~``~~
- * ~~``~~
- * `<sub>` (subscript)
- * `<sup>` (superscript)
- * `<meta>`
- * `<audio>, <video>` Ny!
- * `<div>`
- * `<canvas>, <time>` Ny!
- * `<header>, <footer>` og `<section>` Nye!

Logiske v.s. fysiske tagger

- * En logisk tag sier bare noe om det logiske innholdet. Feks. **<p>** for paragraph
- * En fysisk tag sier noe om de “fysiske” egenskapene, feks. **** for bold.
- * Vi ønsker logiske og semantiske tagger. Feks. **** for emphasize
- * CSS håndterer de “fysiske” egenskapene

Atributter (legenskaper)

- * Atributter settes slik: ``
- * `id="alfanumeriskID"` - brukes i svært mange tagger.
- * `width="xx"` sette bredde på elementet.
- * `tabindex="x"` setter tab-rekkefølge
- * Mange, mange andre. Slå opp hver tag!

Interne lenker

- * Vi bruker "id"-atributtet i (feks.) a-taggen til å lage et anker
- * Vi bruker "#" fulgt av id, i href-attributtet, for å lenke til et anker
 - * ``
 - * ``

Interne lenker

* Eksempel:

http://en.wikipedia.org/wiki/Css#CSS_2

Undersider

- * Det er alltid lurt å dele opp koden
- * Filer i samme mappe refereres til med "href" til filnavn
- * Ellers: Unix mappe-syntaks
 - * href=".//subfolder1/fil1.html"
 - * href=".//siblingfolder1/fil1.html"

Undersider forts.

- * Absolutt v.s. relativ filsti:
 - * <http://domene.no/mappe/fil.html>
 - * ./mappe/fil.html
- * Bruk alltid relativ sti der du kan
- * Hvorfor? Fordi navn endres.

Kommentarer

- * Noen ganger vil man ha kommentarer i HTML-koden
- * Kommentarer synes ikke på selve siden
- * Syntaks:

<!-- Dette er generert av PHP -->

- * Alt mellom “” blir “usynlig”

Tagger fra oppgavene

- * Header (overskrift):
`<h1>tom. <h6>`
- * Paragraph (Avsnitt):
`<p> ... </p>`
- * Anchor (Anker):
`<a ... > `
- * Image (Bilde)
``
- * list (ordnet/uordnet)
` / `

Tagger for uken

- * Header (overskrift):
`<h1> tom. <h6>`
- * Paragraph (Avsnitt):
`<p> ... </p>`
- * Anchor (Anker):
`<a ... > `
- * Image (Bilde)
``
- * list (ordnet/uordnet)
` / `
- * Linjeskift: `
`
- * `<table>` (tabell) med
`<tr>, <th> og <td>`
- * `<meta>`
- * `<!DOCTYPE >`

Avslutting av tagger

- * I XHTML må alle tagger lukkes
- * Med HTML5 er det unntak for "tomme" tagger, bla. :
 - * br, hr, img, meta
 - * Disse "kan" lukkes direkte -
, men skal ikke ha egen lukke-tag. (**IKKE
 ?? </br>**)
 - * I HTML5 er også bare start-tag OK for disse:

- * Fullstendig liste her:
<http://www.w3.org/TR/html-markup/syntax.html#void-element>

Demo!

`raspberry_pi.html`

Validering

- * Alle sidene vi lager i kurset skal validere med HTML5-validator
 - * <http://validator.w3.org/>
- * Vi aksepterer ingen "Errors", men blå "Warnings" går greit.
- * Du kan enten lime inn koden direkte, eller laste opp filen
- * Validerer det jeg nettopp lagde? (Oh-oh...)

Tegnsett

Tekst hentet fra disk eller nettverk:

1110011001110010011011000110100101100111

Hvordan tolker vi dette om til leselig tekst?

Tegnsett

Tekst hentet fra disk eller nettverk:

1110011001110010011011000110100101100111

- * Det opprinnelige tegnsettet heter ASCII
(American Standard Code for Information Interchange)
- * 7 bit pr. tegn - totalt 128 tegn
- * Leser en og en byte - 1 bit til overs
- * A-Z, a-z, 0-9, (), [], {}, !"#\$%, etc.

Tegnsett

Tekst hentet fra disk eller nettverk:

0110011001110010011011000110100101100111

- * Det opprinnelige tegnsettet heter ASCII
(American Standard Code for Information Interchange)
- * 7 bit pr. tegn - totalt 128 tegn
- * Leser en og en byte - 1 bit til overs
- * A-Z, a-z, 0-9, (), [], {}, !"#\$%, etc.

Tegnsett forts.

- * Problem: Hva så med æøå? Ä,€,ç ?
- * Lokal løsning: Utvidelser av ASCII
- * Mest utbredt i Europa/USA: ISO 8859-1
 - * En av mange ASCII-varianter
 - * 8 bit pr. tegn
- * Problem: Mange tegnsett = forvirring

Tegnsett: ISO 8859-1

Tekst hentet fra disk eller nettverk:

1110011001110010011011000110100101100111

Hvordan tolker vi dette om til leselig tekst?

Tegnsett: ISO 8859-1

Tekst hentet fra disk eller nettverk:

1110011001110010011011000110100101100111

* Leser 1 byte om gangen

Tegnsett: ISO 8859-1

Tekst hentet fra disk eller nettverk:

1110011001110010011011000110100101100111

- * Leser 1 byte om gangen
- * Byte 1 : $2+4+32+64+128 = 230$

Tegnsett: ISO 8859-1

Tekst hentet fra disk eller nettverk:

1110011001110010011011000110100101100111

230 = Å

- * Leser 1 byte om gangen
- * Byte 1 : $2+4+32+64+128 = 230$
- * Slår opp 230 i ISO 8859 - 1 : "Å"

Tegnsett: ISO 8859-1

Tekst hentet fra disk eller nettverk:

11100110 01110010 01101100 01101001 01100111

230 = Å 114 = r 108 = l 105 = i 103 = g

- * Leser 1 byte om gangen
- * Byte 1 : $2+4+32+64+128 = 230$
- * Slår opp 230 i ISO 8859 - 1 : "Å"

Tegnsett: ISO 8859-1

Tekst hentet fra disk eller nettverk:

11100110 01110010 01101100 01101001 01100111

230 = Å

114 = r

108 = l

105 = i

103 = g

- * Når siste bit er 1, avviker vi fra ASCII
- * Resultat: “Ærlig”
- * OK, men da må “alle” støtte mange tegnsett

Tegnsett forts.

- * Global løsning: Unicode
 - * Felles tegnsett for alle språk
- * UTF-8 er en implementasjon av Unicode
 - * Bakoverkompatibelt med ASCII
 - * Variabel lengde fra 1-4 byte
 - * Representerer alle tegn i Unicode

Tegnsett: Entiteter

- * Alternativ HTML-løsning: Entiteter
 - * Bruker kun ASCII-tabellen
 - * Flere tegn tolkes om til ett
- * Eksempel:
 - * Æ: “&Aelig;” eller “Æ”
 - * æ: “æ” eller “æ”
- * [http://www.w3schools.com/tags/
ref_entities.asp](http://www.w3schools.com/tags/ref_entities.asp)

Whitespace

- * Hva skjer når vi legger inn ekstra mellomrom og linjeskift i HTML-fila?
 - * Ingen ting.
 - * HTML er whitespace-insensitivt.
 - * “Whitespace” er alle usynlige tegn
- * Hvorfor
 - * Vi bruker tagger, og CSS til å indikere utseende
 - * Da kan vi enkelt gi en stor “site” lik layout

I morgen og neste uke:

- * I morgen: Flere oppgaver på lab
- * I neste uke:
 - * Publisering på nett
 - * Oblig1 ut!
 - * Mer HTML og intro til CSS

Mere nytt i HTML5

* <http://html5demos.com/>