

# CMPT 165

## INTRODUCTION TO THE INTERNET AND THE WORLD WIDE WEB



### Unit 2

### Markup and XHTML

# Learning Objectives

In this unit you will learn the following.

- **Create** web pages in HTML with a text editor, following the rules of XHTML syntax and using appropriate HTML tags.
- **Create** a web page that includes links and images.
- **List** some common HTML tags.
- **Use** relative URLs to refer to resources on a website.

# Topics

1. Making Web Pages
  2. Your First HTML Page
- 

**Lecture 1**

3. HTML Tags
  4. Why Do Markup?
- 

**Lecture 2**

5. Attributes
  6. List Tags, Another Page
  7. Images in HTML
  8. Hypertext Links
  9. Using HTML Tables
- 

**Lecture 3**

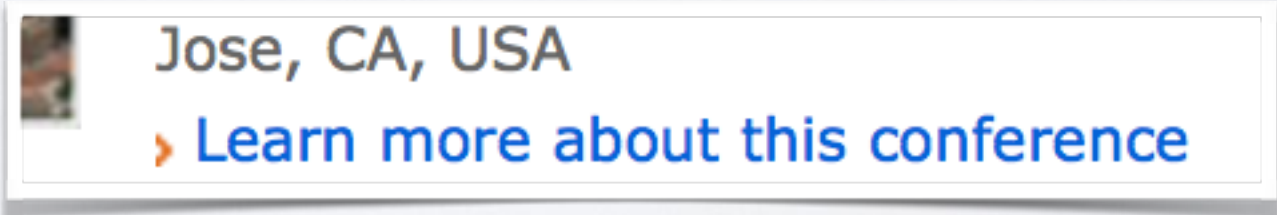
10. Relative URLs
11. Multi-Page Sites

**Lecture 4**



# A Web Page

*def.* an HyperText Markup Language (**HTML**) document connected to the World Wide Web (**WWW**).

- **WWW:** an Internet system of documents that are connected to other documents by **hypertext links**, where information can be searched for by clicking on these links — moving from one document to another.
- **Hypertext Links:** A screenshot of a web interface element. It features a small profile picture on the left. To its right, the text "Jose, CA, USA" is displayed in a dark font. Below this text is a blue hyperlink that reads "› Learn more about this conference". The entire element is enclosed in a thin white border with a subtle drop shadow.
- **Website:** a location on the Internet that contains web page(s) on the WWW.

# HTML

## HyperText Markup Language

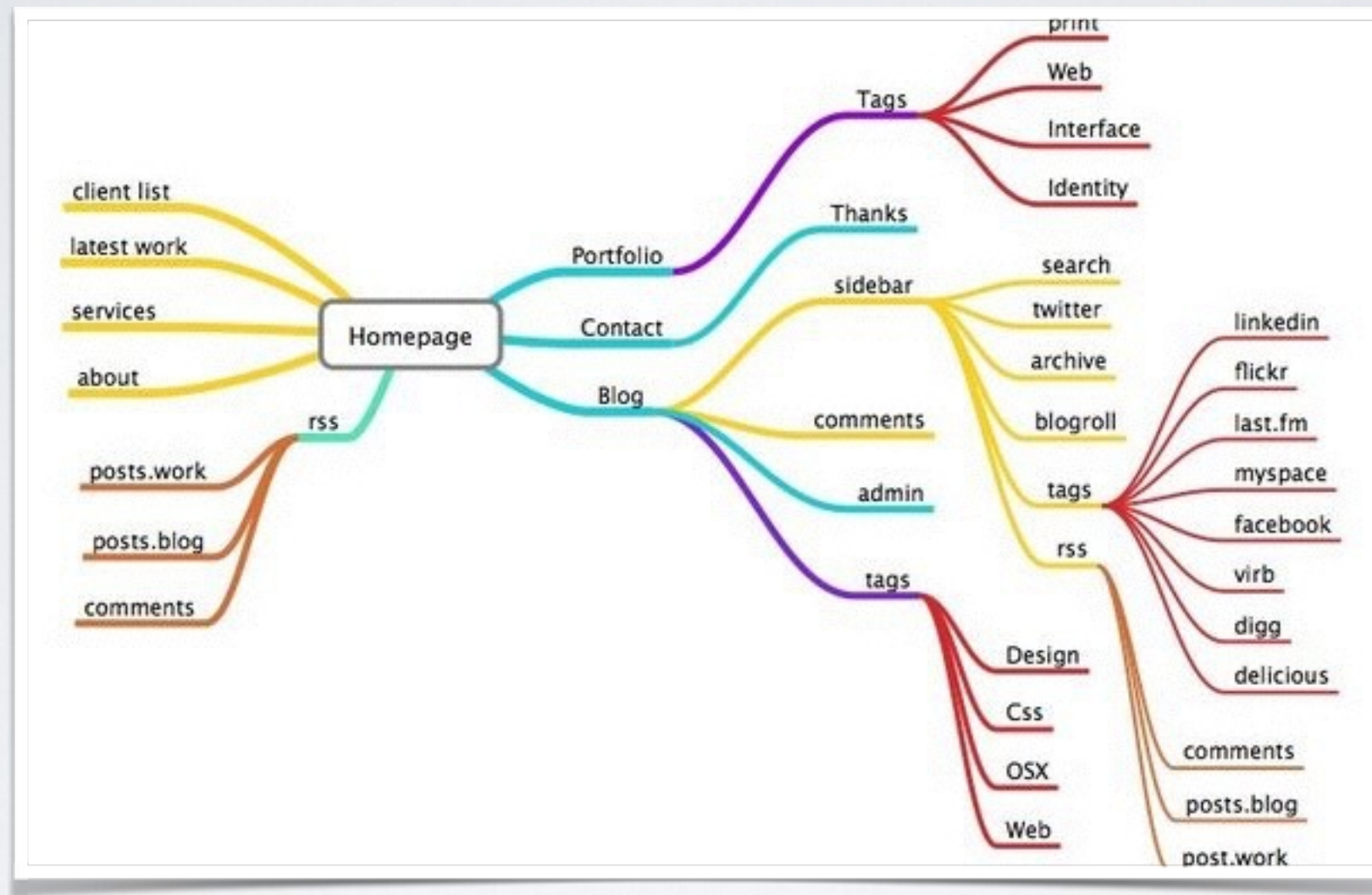
*def.*, a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on web pages.

- **XHTML:** eXtensible HyperText Markup Language
- **HTML, XHTML?** Main different is the enforcement of closing opened tags (more on this next lecture).
- This allowed for easier parsing of web pages.
  - e.g. Googlebot — for information searching
- XHTML is now an old standard and most web developers use HTML 5.



# Hypertext Linking

e.g. Site Maps



How a website is organized by its (hypertext) links.  
However, this does not show to other websites.



# Social Media Linking

e.g. Who are the friends of your friends?

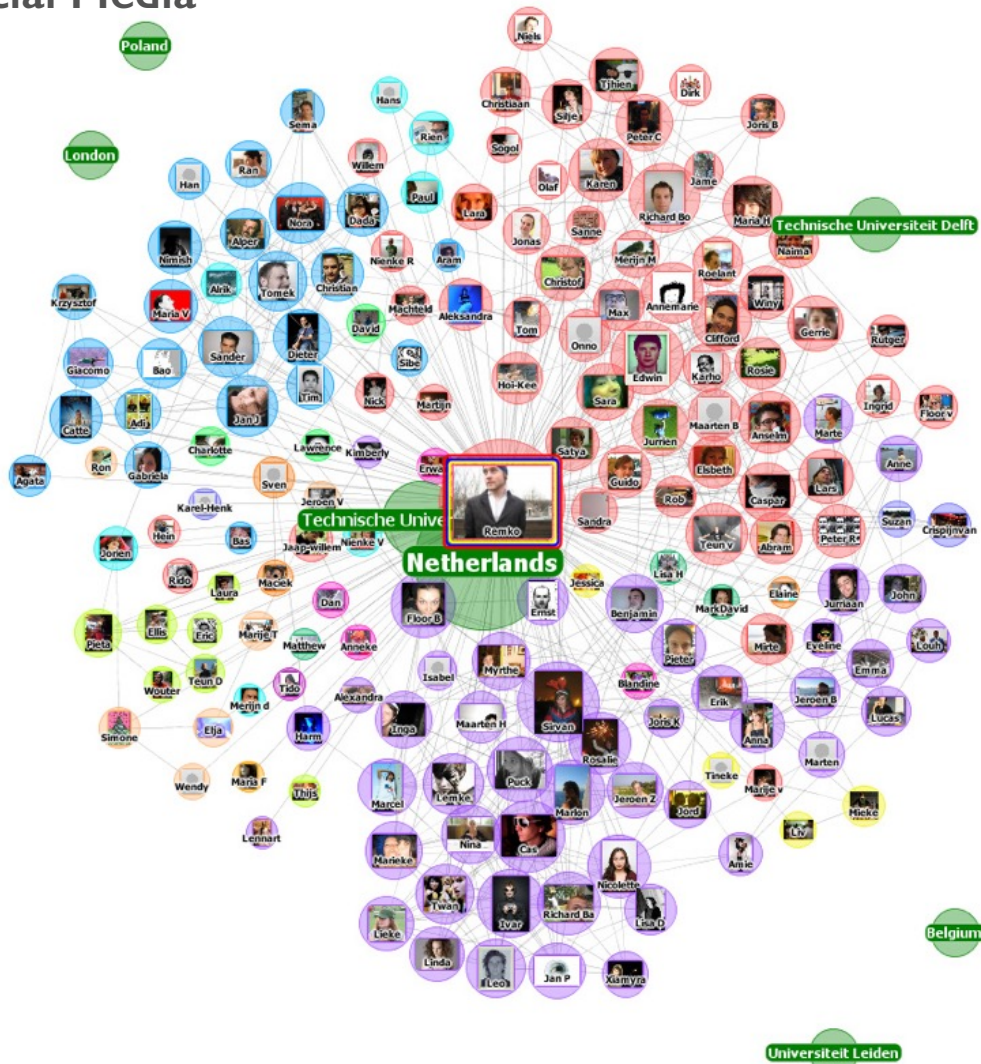




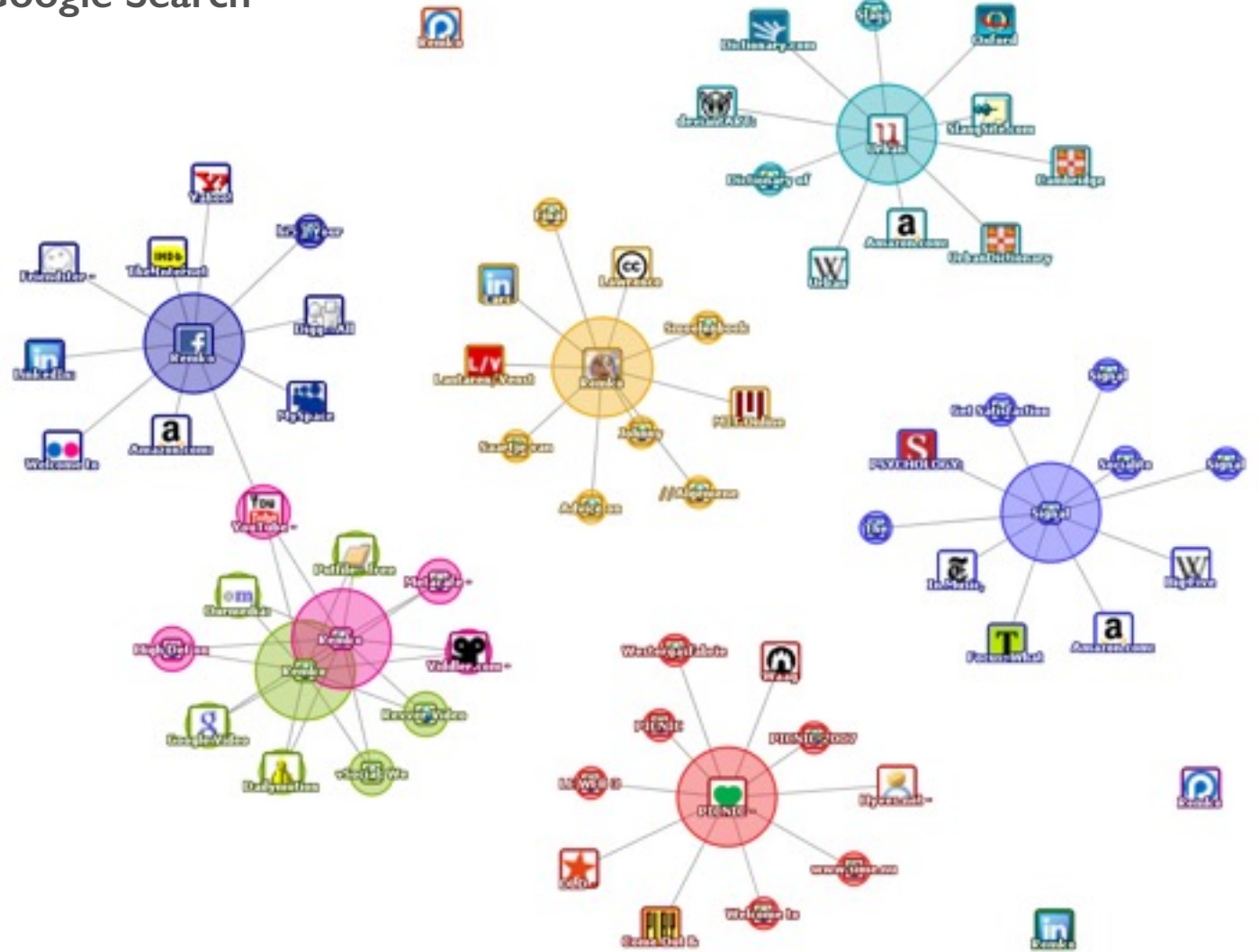
# Social Media vs Hypertext

# Comparing Different Ways of Linking

## Social Media



# Google Search



# IS IT POSSIBLE FOR MARKETING COMPANIES TO BUILD PROFILES OF PEOPLE THROUGH LINK MINING?



# Other Ways of Linking

## (to Create a Profile of **You**)

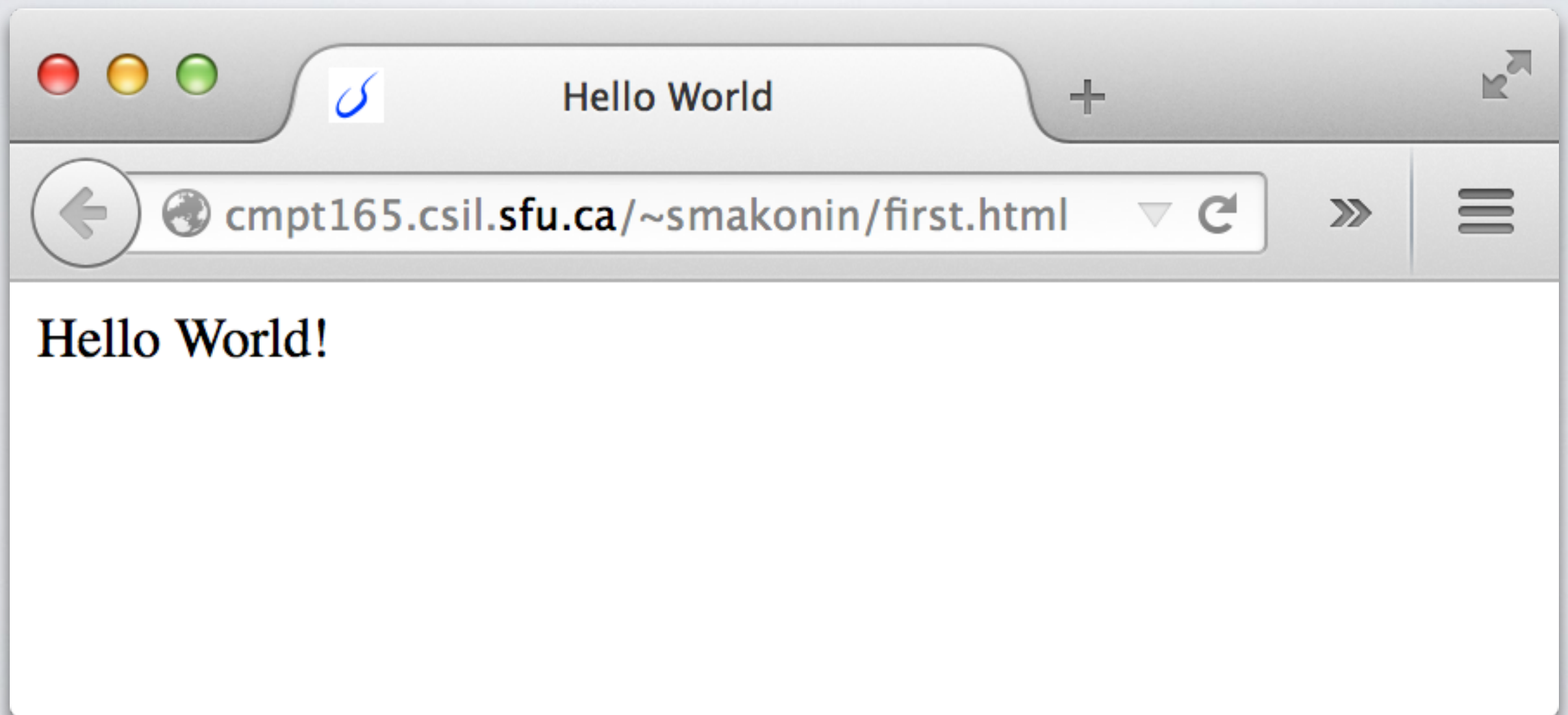
- Friends on Social Media
- Business Connections
- Faces in Pictures
- Purchase History
- Recommendations
- Tweets
- Blog Posts Authorship
- Rating a Movie or Book
- Comments in a Forum
- Profile Page on a Site
- Search History
- IP Address / Web Logs / Page History
- 3<sup>rd</sup> Party Cookies (the ads that follow you)



# Your 1<sup>st</sup> HTML Page

first.html

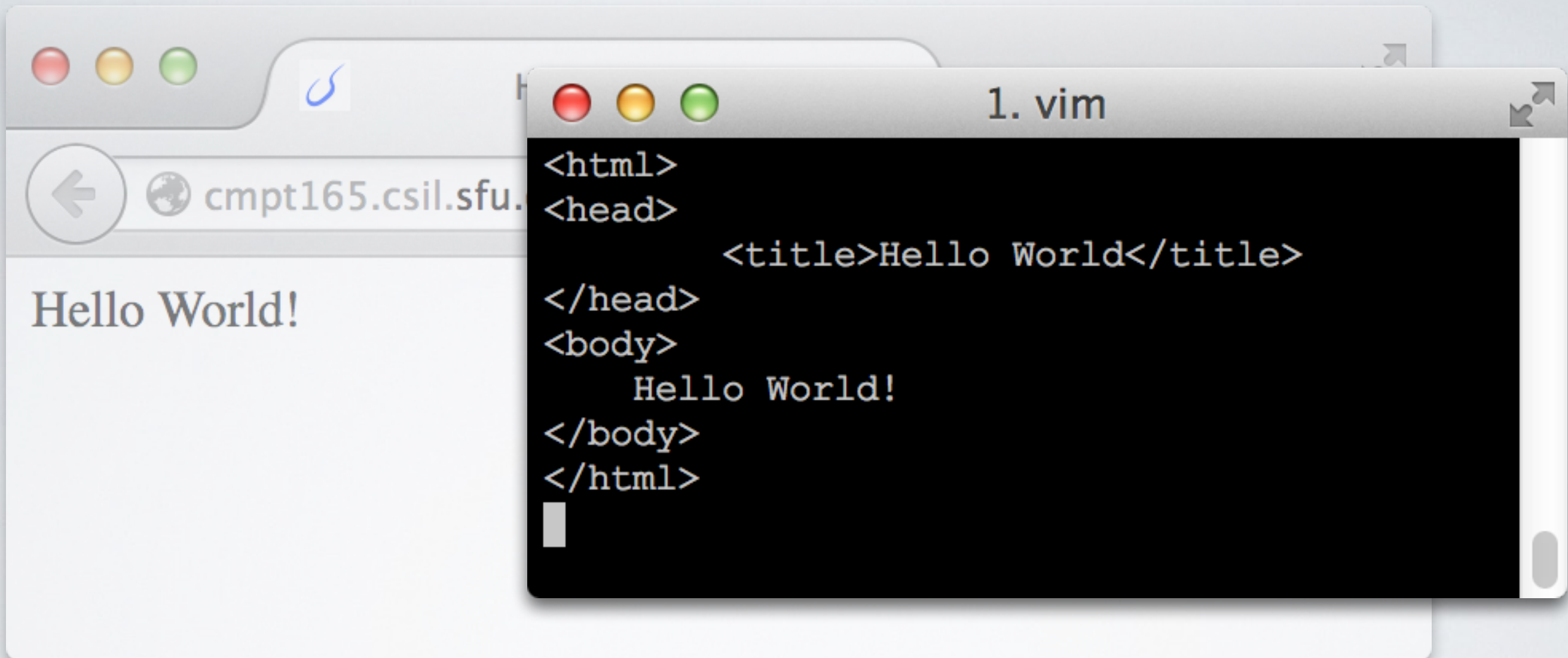
URL: [examples/first.html](http://examples/first.html)





# The HTML Behind

first.html



An HTML document  $\equiv$  Structure & content of a Web Page

# <html> tag



- Each document must begin and end with **<html>**
- 2 parts to HTML: **head** and **body** — no footer

**WHAT IS THE DIFFERENCE BETWEEN AN  
OPEN TAG AND A CLOSED TAG?**



# <head> tag

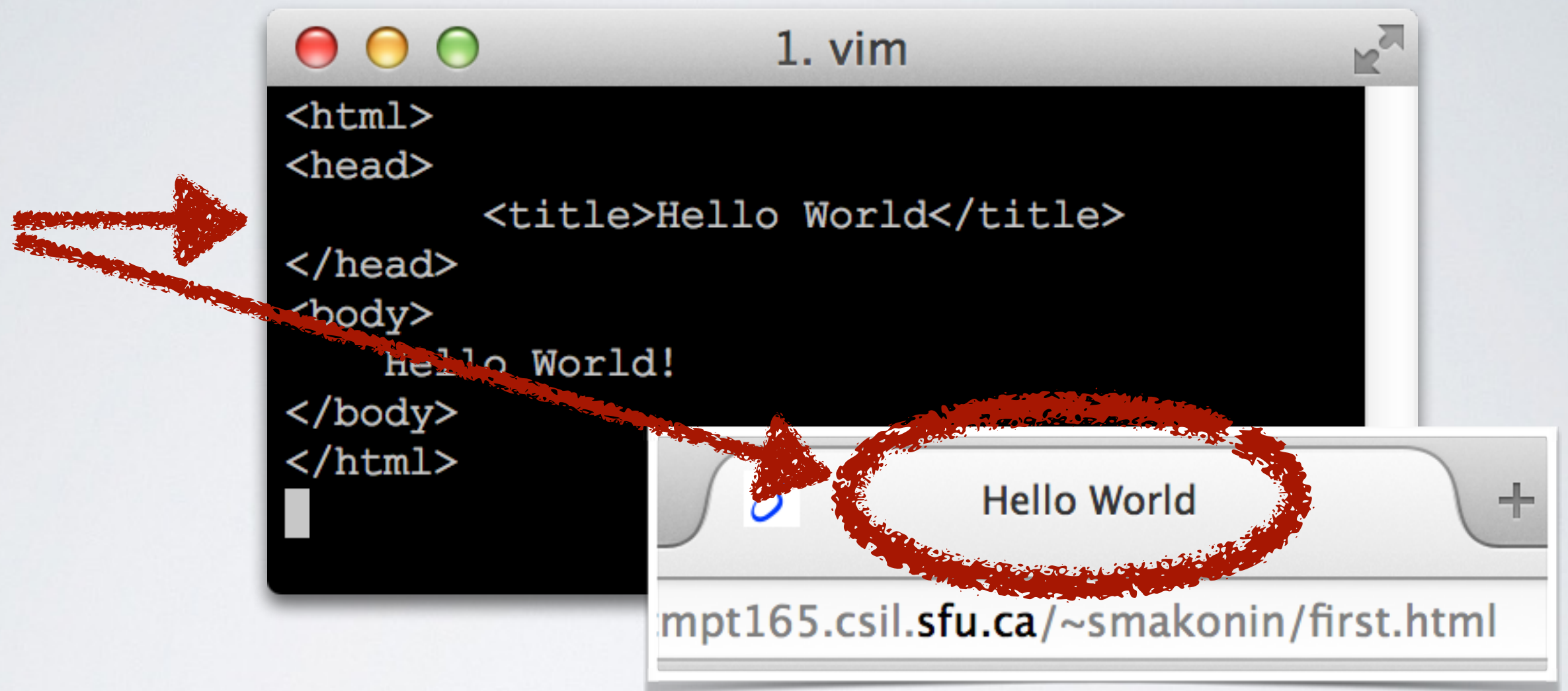


```
1. vim
<html>
<head>
    <title>Hello World</title>
</head>
<body>
    Hello World!
</body>
</html>
```

- defines the title of the document
- can include scripts, instruct the browser where to find style sheets, provide meta information, and more.

## WHAT IS THE DIFFERENCE BETWEEN A HEADER IN WORD VS. HTML?

# <title> tag



- in all HTML documents and it defines the document title
- defines a title in the browser toolbar and in search results
- used when it is added to favourites



# <body> tag



- contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.

# Editing HTML

- Any old text editor will do! Free editors:
  - **Windows:** Notepad, [CoffeCup](#), [Sublime](#)
  - **Linux:** GEdit, VIM, [Sublime](#), many others
  - **Mac OSX:** TextEdit, VIM, Xcode, [Sublime](#)
- Graphical editors such as Dreamweaver or Word are **not allowed!** They create very messy, verbose HTML documents:
  - e.g. [Simple example using Word](#)





**QUESTIONS?**

# Why Markup

- Markup is used to define the structure of a document.
- Word processors (e.g. Word, Pages) do this too.
- Along with CSS, markup instructs the browser how to render the HTML for your webpage.
- e.g., in HTML you could have a heading and CSS would specify the look of the heading font:

*type*, *SIZE*, **colour**, margins, etc.



# Closing Tags

In XHTML open tags must be closed to have a valid XHTML document / web page.

```
<title> My Website!!! </title>
```

# Closing Tags

In XHTML open tags must be closed to have a valid XHTML document / web page.



Some tags contain no content (or text) between the open and closed tag. These are called **empty tags**. So they can self-close:

`<br/>`



# Closing Order

- If you have multiple open tags you must close them in reverse order, to have valid XHTML, e.g.

`<em><a></a></em>`

`<a><em></em></a>`

- If not it is incorrect, e.g.

`<em><a></em></a>`

`<a><em></a></em>`

- Remember

**LOFC** *|lōfs|* — **L**ast tag **O**pened, **F**irst tag **C**losed!

# Whitespace

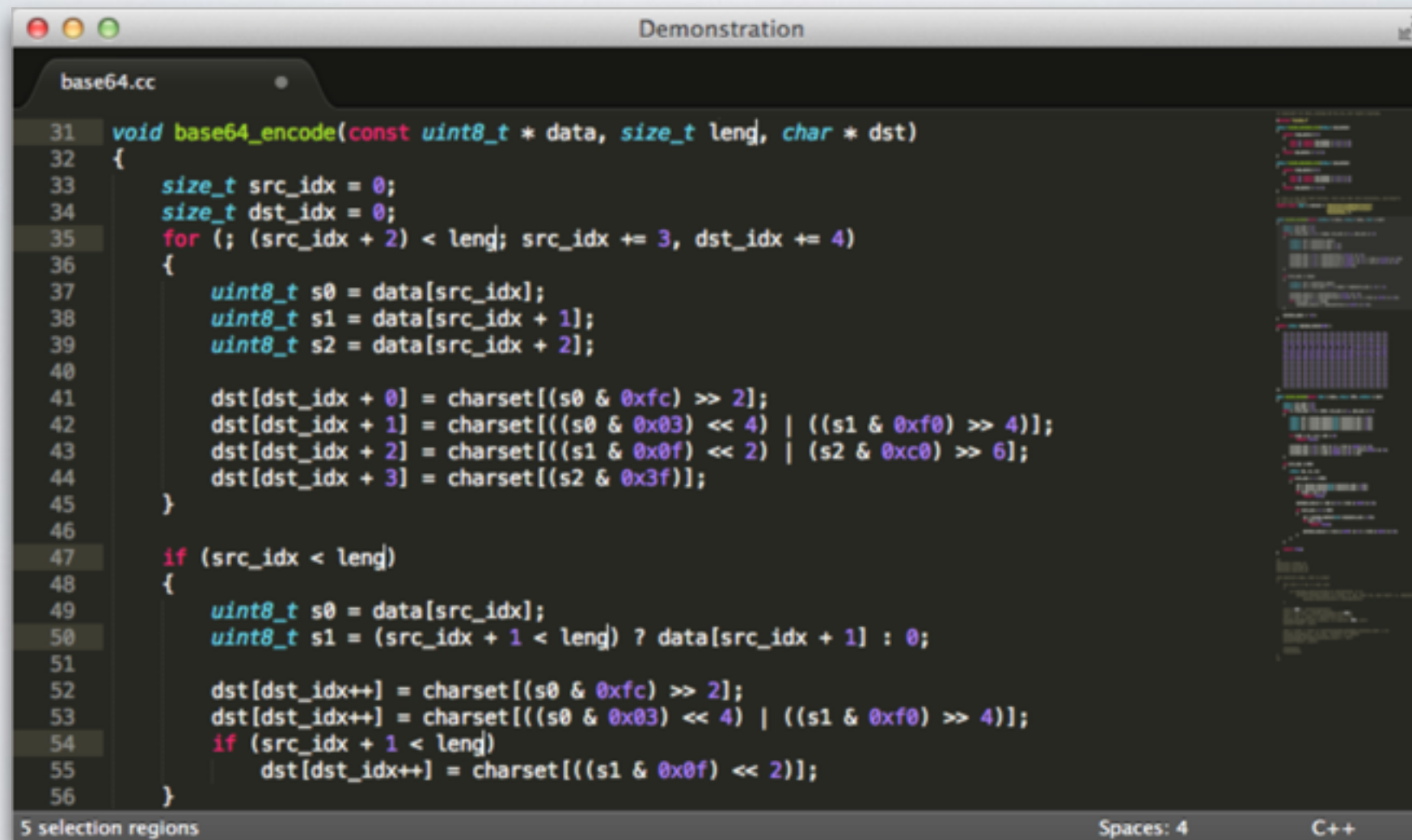
- The browser when rendering the HTML to the screen ignores the *white space* contained in the HTML file.

```
<h1>Page Heading</h1>
<h1>Page      Heading</h1>
<h1>Page
      Heading</h1>
<h1>
Page Heading
</h1>
```

- If you want white space in your on your web page then you need to use the right HTML tag of character entity to do this.



# Sublime Text Editor



```
31 void base64_encode(const uint8_t * data, size_t leng, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < leng; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < leng)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < leng) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < leng)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
```

- URL: <http://www.sublimetext.com>
- Same editor for Mac OSX, Linux, and Windows
- Free to use, with occasional *Buy Me* popup

# Basic Tags

`<b>` is for bold which also looks `<strong>`

`<blockquote>` is of indented quoting

`<br/>` is for line break

`<center>` is for centring on the page

`<h#>` is for heading, from 1 to 6, e.g. h1, h2, h3...

`<hr/>` is for horizontal rule

`<i>` is for italic (same as `<em>` is for emphasis)

`<p>` is for paragraph

`<pre>` is for pre formatted text

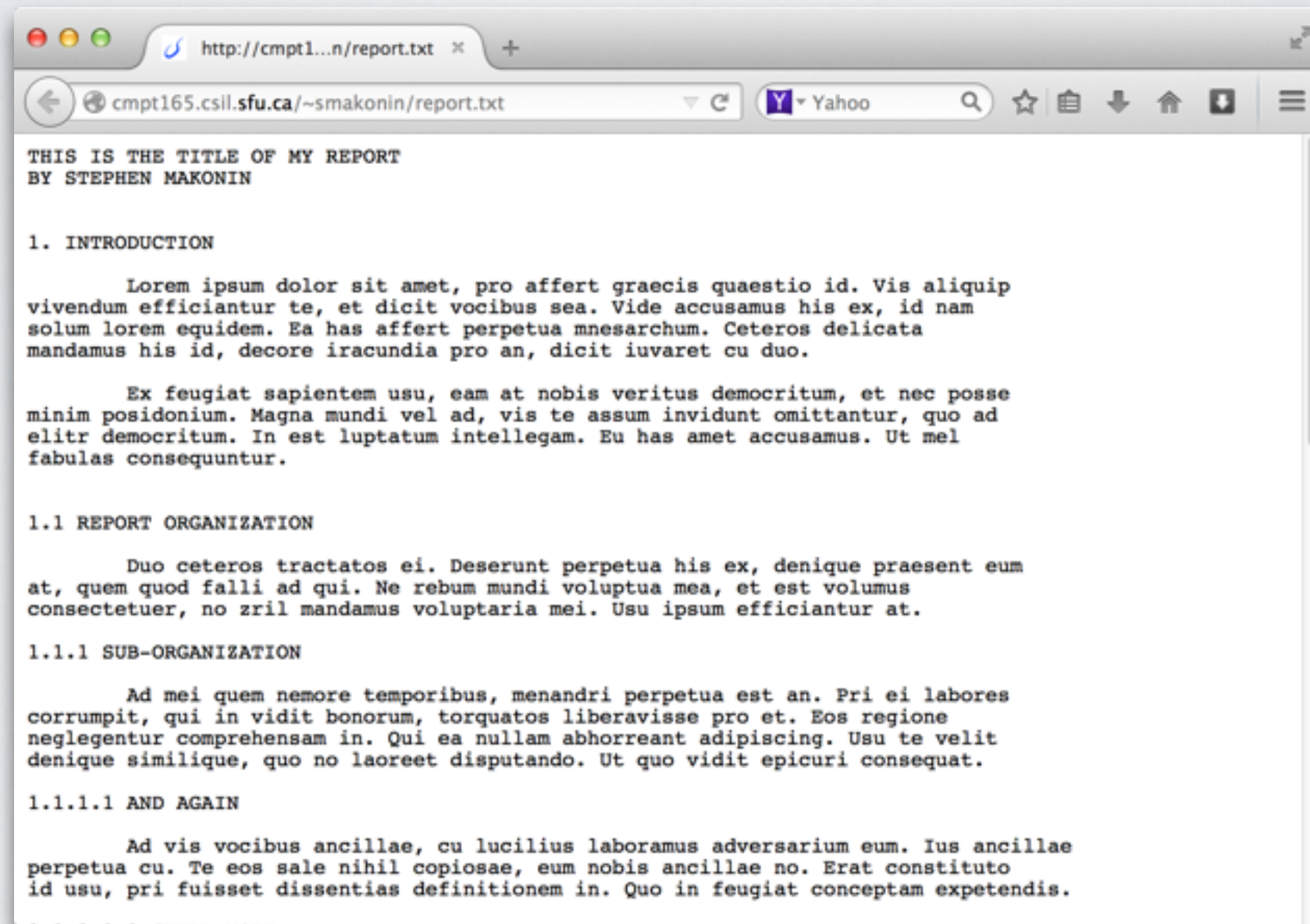
`<q>` is for quoting texted (for citations)

`<tt>` is for typewriter spacing (monospace), not = `<pre>`

`<u>` is for underline

# Class Demo

Convert a plain text document into HTML



URL: [examples/report.txt](http://examples/report.txt)  $\Rightarrow$  [examples/report1.html](http://examples/report1.html)



# Element vs Tag

- Elements are a semantic structure that usually consists of a start tag, content, and an end tag.

e.g. `<h1>My Report</h1>`

- Special cases are self-closing tags, e.g. `<hr/>`
- Tags are used to specify the type of element and to mark the start and end of an element, e.g. `<h1>`, as above.
- For a good article see: [http://www.456bereastreet.com/archive/200508/html\\_tags\\_vs\\_elements\\_vs\\_attributes/](http://www.456bereastreet.com/archive/200508/html_tags_vs_elements_vs_attributes/)



# QUESTIONS?

# Tag Attributes

- Attributes modify tags and give context and meaning.
- Also, to specify different options available for that tag.

*attribute\_name* = "*value*"

e.g. `<h1 id="title">My Report</h1>`

- **Always** enclose the *value* with quotes ""
- Some attribute can have a specific set of values.
  - e.g. in a order list we can specify the type of numbering used: **1, 2, 3...** or **A, B, C...**



# HTML Global Attributes

- `accesskey=""` Specifies shortcut key to set focus
- `class=""` Specifies the classname defined in CSS
- `dir=""` Specifies text direction, e.g. `rtl|ltr|auto`
- `id=""` Specifies a unique id for an element
- `lang=""` Specifies language of the element's content
- `style=""` Specifies an inline CSS style for an element
- `tabindex=""` Specifies the tabling order
- See: [http://www.w3schools.com/tags/ref\\_standardattributes.asp](http://www.w3schools.com/tags/ref_standardattributes.asp)

# HTML Lists

- Why are lists useful?
  - Create outlines for large webpages
  - Summarizes points, e.g. ...*this includes*:
  - Describe steps in a process or in directions
  - Have nomenclature or a set of definitions
  - Use the same way as in a document/report
- You can have lists within lists
- See: [http://www.w3schools.com/html/html\\_lists.asp](http://www.w3schools.com/html/html_lists.asp)

# 3 Types of Lists

**<ul>** is for unordered list

attribute **type** = "disc|circle|square"

has 1 or more **<li>** is for list item

**<ol>** is for ordered list

attribute **type** = "1|A|a|I|i"

attribute **start** = "1,2,3..."

has 1 or more **<li>** is for list item

**<dl>** is for definition list

has 1 or more **<dt>** is for definition term and

with **<dd>** is for definition description

See: [examples/nolist.html](http://examples/nolist.html) ⇒ [examples/lists.html](http://examples/lists.html)



# <img> tag

```

```

- self-closing tag
- attribute **src** is a relative URL to the image file
- attribute **alt** ha textually describes image
  - search engines like this 😊👍
- attribute **width** in pixels or percent % or actual size
  - "400" resize to a width of 400 pixels
  - "50%" resize to half of the actual size
- attribute **height** in pixels or percent % or actual size

# <a> tag

<a href="examples/report.txt">report</a>

- <a> is for anchor or hyper link, each has two ends:
  - a source that points to a destination
  - **source:** content can be text, image, etc.
  - **destination:** a resource/file that can be an image, a video, audio, a program, an HTML document
- attribute **href** is a relative URL to the image file
- attribute **target** specifies where to open the has is a relative URL to the image file
- Link to tag IDs (or locations) in a webpage:

e.g. "examples/report.html#intro" ⇒

<h1 id="intro">1. INTRODUCTION</h1>

#

hash tag

# HTML Tables

Month	Savings
January	\$100
February	\$80

```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

`<table>` is for table  
`<tr>` is for table row  
`<th>` is for table heading  
`<td>` is for table data

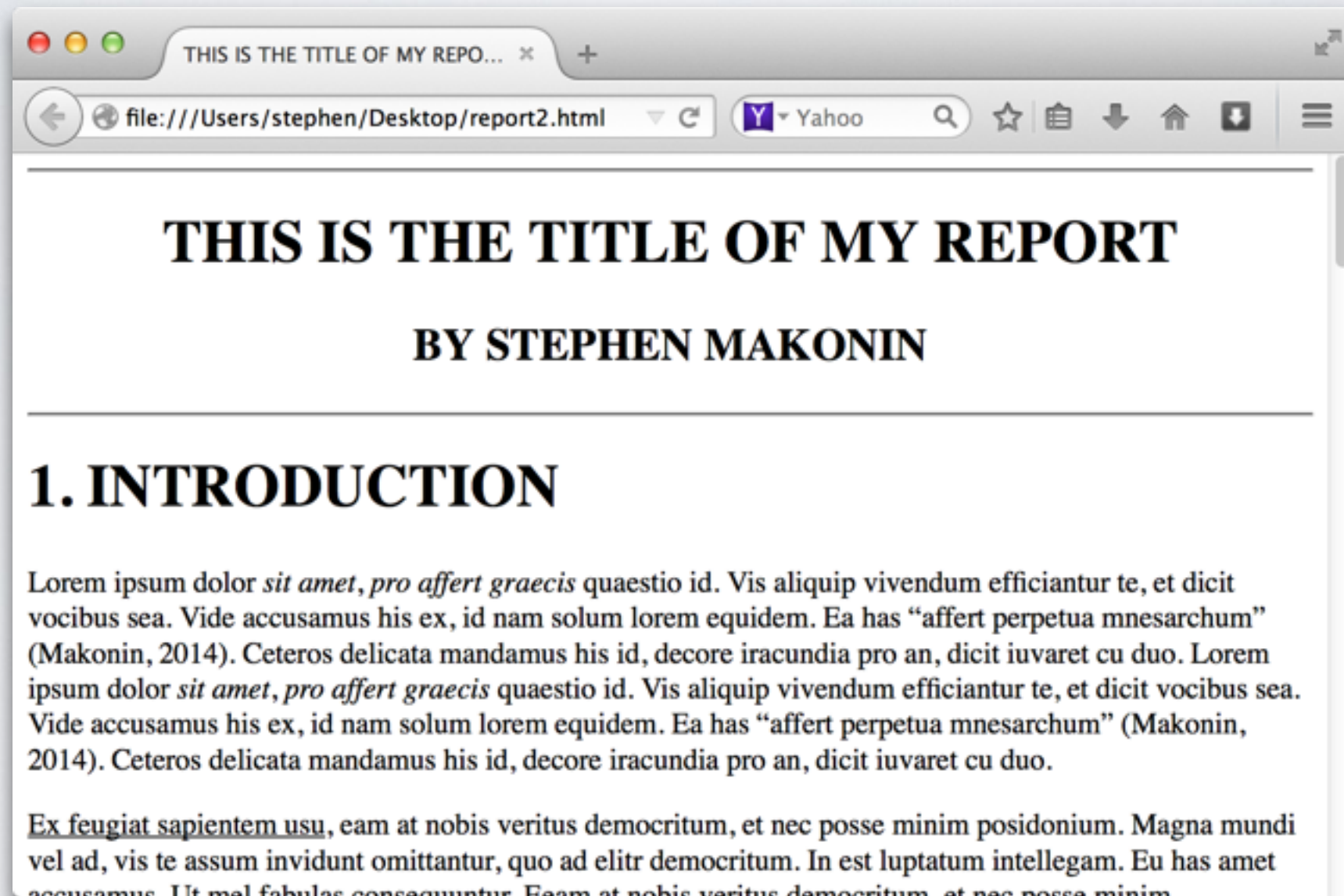
- attribute **colspan**="1,2,..." to merge columns
- attribute **rowspan**="1,2,..." to merge rows

We will **not** learn any other attributes, instead will use CSS in Unit 3 to style and formatting tables.



# Class Demo

Adding lists, images, and links



URL: Report v1  $\Rightarrow$  Report v2



**QUESTIONS?**

# Default Files

`http://www.sfu.ca/`

How does SFU.ca show a webpage without a path?

1. The slash / is the path & if not, then / is assumed.
2. The web server is setup to have a default file if only the directory is specified.
  - Web servers are configured to check for defaults  
e.g. "index.php", "index.html", "default.htm"
  - So / would be equivalent to /**index.html**
  - **Security:** Prevents directory browsing, [see here](#).
    - may have old files, time sensitive files



# Relative URLs

URL	Destination
img.png	http://www.sfu.ca/~somebody/pics/img.png
../file.html	http://www.sfu.ca/~somebody/file.html
../.. /test.html	http://www.sfu.ca/test.html
dir/img.png	http://www.sfu.ca/~somebody/pics/dir/img.png
http://www.cs.sfu.ca/	http://www.cs.sfu.ca/ (absolute URL)

**Figure 2.8:** URLs starting at `http://www.sfu.ca/~somebody/pics/index.html`

- **Absolute URLs** contain: scheme + server + path  
e.g. `http://www.sfu.ca/~somebody/page.html`
- **Relative URLs** contain only the path  
e.g. `~somebody/page.html`  
e.g. `images/logo.png`  
e.g. `../index.html`

# When to Use

- Relative URLs are used:
  - when adding images, audio, video to the web page
  - to linking to other pages on your website
  - to load CSS style files (more in Unit 3)

**General Rule:** Use relative URLs when you are referring to some resource/file on your website. When that resource is on another website then you must use an absolute URL.

# Being Relative

URL: <http://www.cs.sfu.ca/CourseCentral/165/smakonin/examples/>

- Filename only

e.g. `report.txt`

- Directory + Filename

e.g. `DirList/file1.html`

- Parent Directory

e.g. `../`

- Parent Directory + Filename

e.g. `../media/`

- Parent Directory + Directory + Filename

e.g. `../media/sfu-logo.png`

- Going back multiple Parent Directories

e.g. `../../`



# Using Relative

LIBRARY

[courseCentral/165/smakonin/examples/](http://courseCentral/165/smakonin/examples/)

example.txt

example/report.html

- Filename + Extension  
e.g. ../media/
- Parent Directory + Directory + Filename  
e.g. ../media/sfu-logo.png
- Going back multiple Parent Directories  
e.g. ../../

Think of it  
as having the  
browser  
remember the  
absolute  
location!

# Relative

L I D I

[courseCentral/165/smakonin/examples/](http://courseCentral/165/smakonin/examples/)

c.txt

Think of it  
as having the  
browser  
remember the  
absolute  
location!

The  
location of  
other files is  
relative to the  
webpage you  
are on!

- Parent Directory + Directory  
e.g. .../med
- Going back multiple Parent Directories  
e.g. .../.../

# Multi-Page Websites

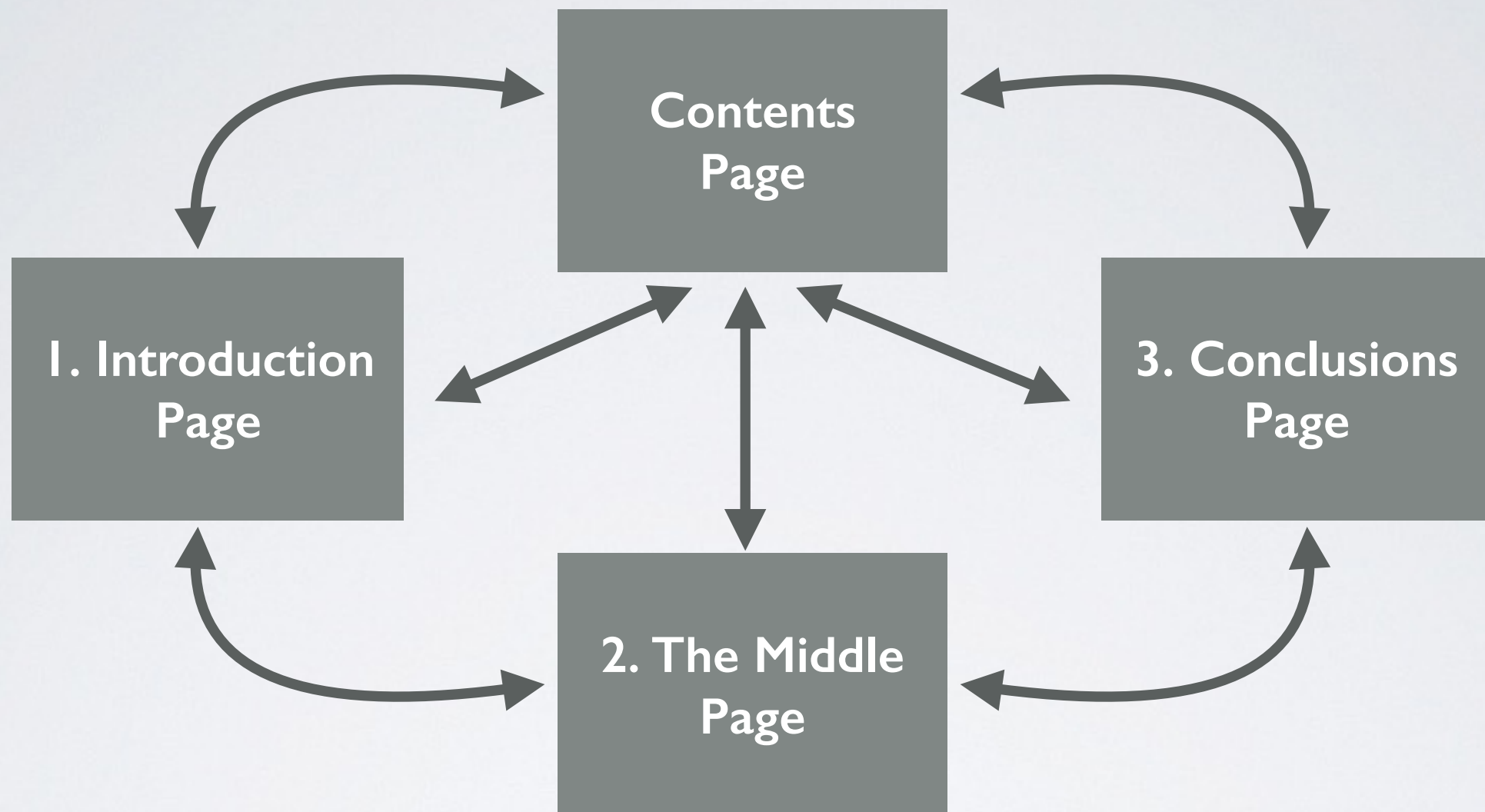
## We need to ask some questions:

- How should site navigation occur?
  - Is there a logical flow the the information?
  - What is the purpose of the information/site?
- What is the best way to organize the files?
  - Is there lots of images? ... product brochures?
    - Create an image folder? ...a brochures folder?
  - Are there different, separate sections?
    - Create a folder for each section?

Look at [Report 2 single page](#) to see what we can do...



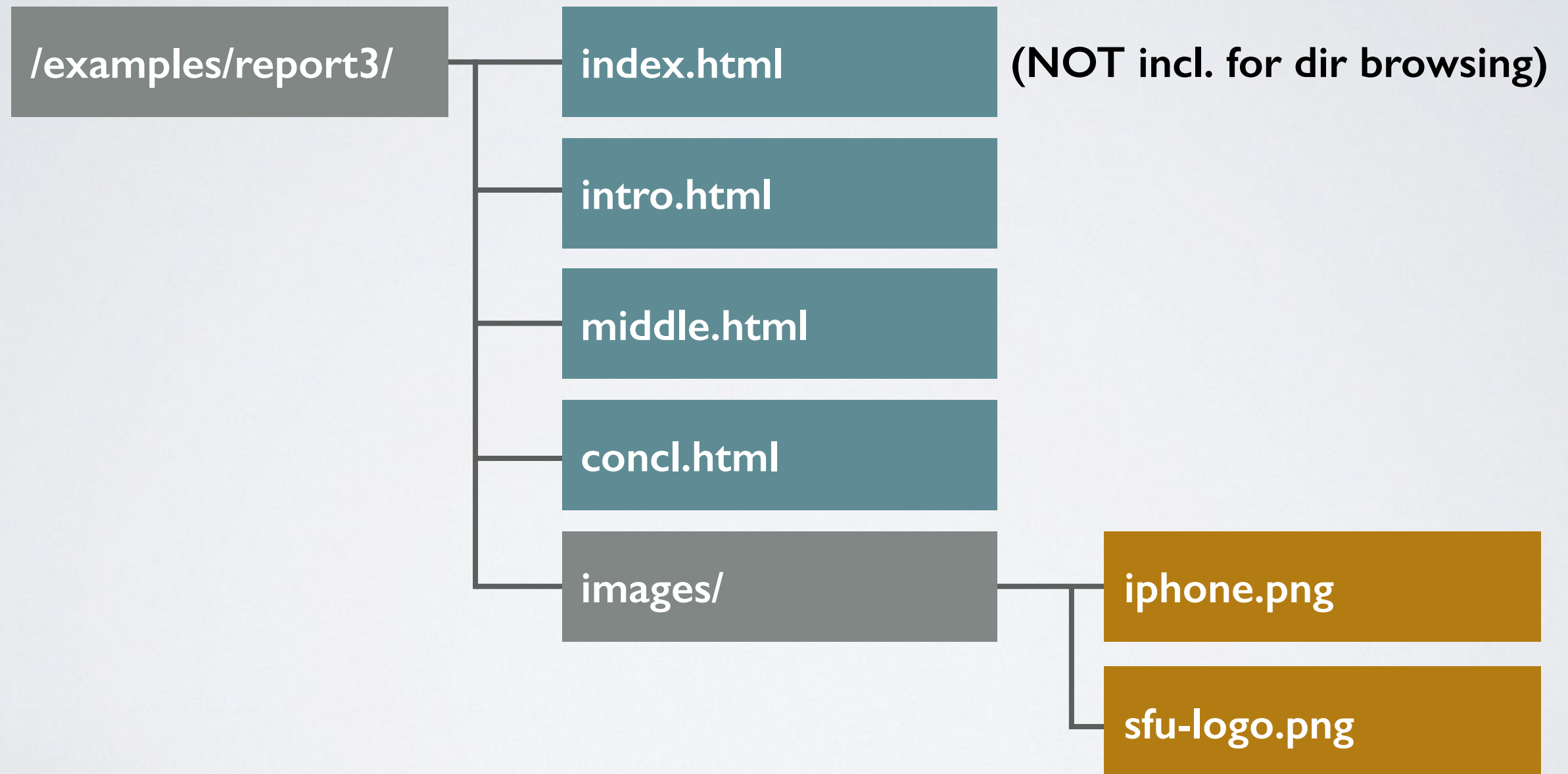
# Report 2 Navigation



**CONTENTS:** [Introduction](#) - [The Middle](#) - [Conclusions](#)

[< Previous](#) | [Go To Top](#) | [Next >](#)

# Report 2 Site Organization



# Class Demo

Multi-Page Report Version



URL: [Report v2](#)  $\Rightarrow$  [Report v3](#)



# Summary

- Looked at markup and how to use it to structure an HTML document.
- Learnt about elements, tags, and attributes.
- Used different HTML tags to create a web page.
- Added images and hyper links to theses pages.
- Created a multi-page website.

**Next Unit:** we look at adding style to HTML with style sheets.



**QUESTIONS?**