CMPT 165 INTRODUCTION TO THE INTERNET AND THE WORLD WIDE WEB

Unit 4
Advanced XHTML and CSS

Learning Objectives

In this unit you will learn the following.

- Use XHTML to create valid web pages.
- Design HTML so it can be easily styled with CSS.
- Develop CSS rules to create particular appearances.
- Understand CSS colour codes for a given colour.
- Construct a CSS that implements a visual design.
- Justify the separation of content and structure from visual appearance.
- Select appropriate HTML tags to correctly describe the different parts of the page.

Topics

1. Validating XHTML
2. Common Mistakes
3. Block vs. Inline Elements

Lecture 1

---

4. Character Entities
5. Generic Tags, IDs and Classes
6. Style Selectors Revisited

Lecture 2

---

7. Positioning Elements
8. Steps in Webpage Creation

Lecture 3

Valid XHTML

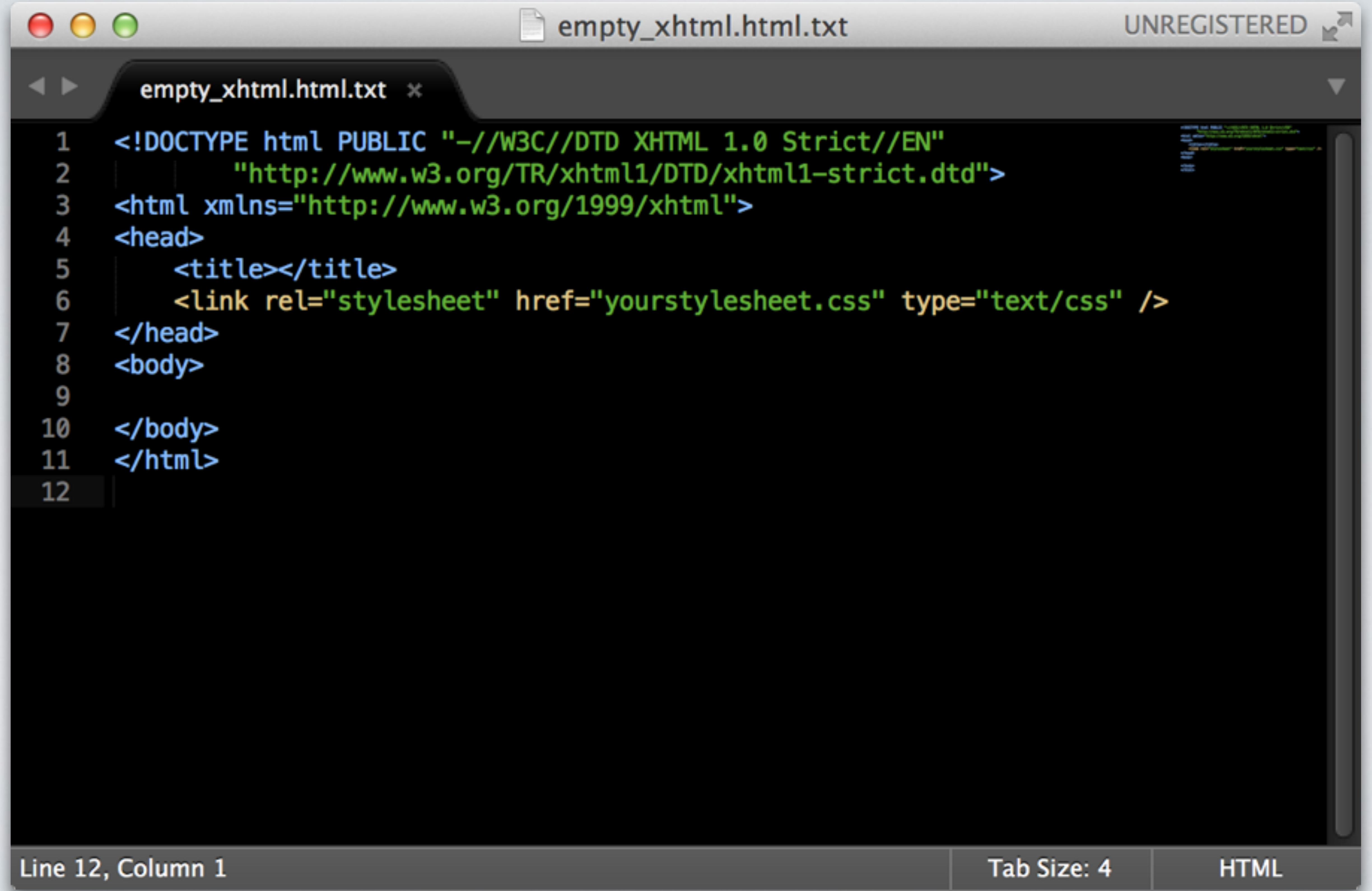
Valid XHTML means your markup follows a set of rules:

- Have a *document type* (`DOCTYPE`) at the top of the.
- Specific the *namespace* in `<html>`.
- Open tags must close in order.
- Inline tags must be inside block tags.
- Some tags such as `<li>` can only be in `<ol>` or `<ul>`.
- Special characters (e.g. `<`) in content must be encoded.
- Markup tags and attributes name are lowercase.

If these rules are followed the a **validator** says: 😄👍

Otherwise: 😞👎

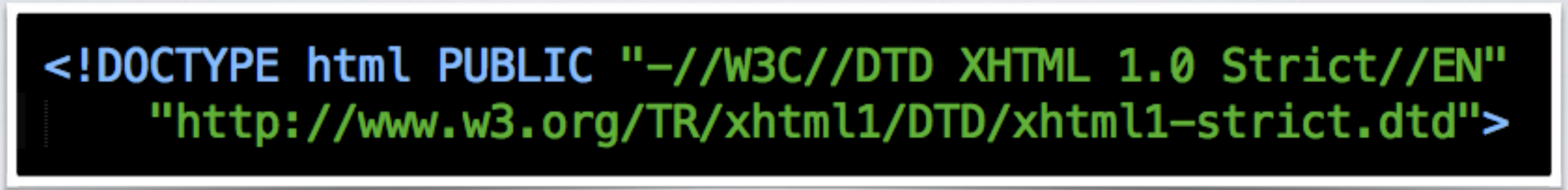
Empty Valid XHTML



```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <link rel="stylesheet" href="yourstylesheet.css" type="text/css" />
</head>
<body>

</body>
</html>
```

Document Type



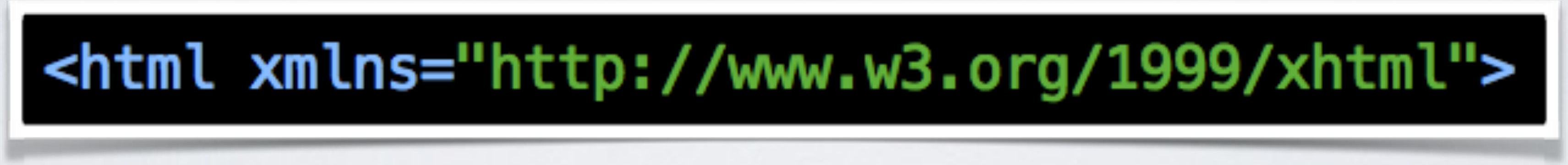
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

You **MUST** declare a document type as the 1st line in your XHTML document.

- So the browser knows what version of HTML/XHTML you are using.
- There is no need to memorize this, copy it from somewhere.
- This can be slit into 2 lines (as above) or on 1 line.
- Above says HTML document is written in XHTML version 1.0 as defined by W3C.

DTD (Document Type Definition)

Namespace



```
<html xmlns="http://www.w3.org/1999/xhtml">
```

def. is a container for a set of identifiers/names.

- Distinguish between identifiers with the same exact name.
- **e.g.** a surname to distinguish people who have the same given name.
- So, we are saying treat the tags as those from XHTML

You **SHOULD** specify the namespace for your XHTML document.

see http://en.wikipedia.org/wiki/Namespace

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!

Block vs. Inline Elements

Elements that go within the `<body>` of an HTML condiment are either **block** (a.k.a. block-level) or **inline** elements.

In the figure:

- **Grey** are inline elements.
- **White** are block elements.

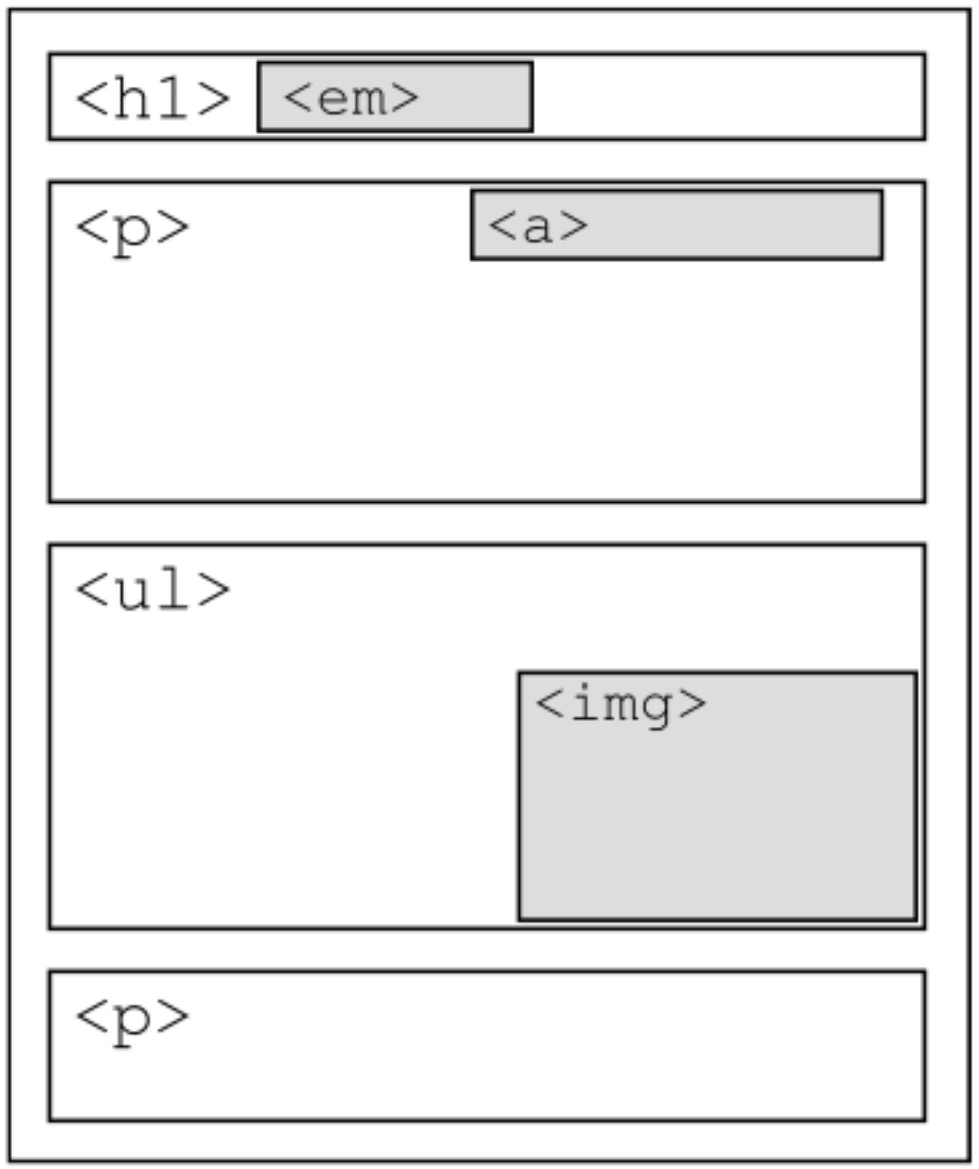


Figure 4.3: Block and inline tags

Source: CMPT 165 Course Study Guide, p. 69

Block Elements

- Occupy the entire space of its parent element
  - (e.g. `<body>`, `<p>`) creating a *block*.
- They begin on a new line and end with a new line.
- May contain inline and other block elements.

`<address>` `<blockquote>` `<dd>` `<div>` `<dl>` `<fieldset>` `<form>` `<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>` `<hr>` `<noscript>` `<ol>` `<p>` `<pre>` `<table>` `<tfoot>` `<ul>`

Inline Elements

- Occupy only the space bounded by by the tags that define the inline element.
- They do not begin with new line.
- Contains only data and other inline elements

```
<a> <abbr> <acronym> <b> <bdo> <big> <br>
<cite> <code> <dfn> <em> <i> <img> <input>
<kdb> <label> <q> <samp> <select> <small>
<span> <strong> <sub> <sup> <textarea>
<td> <th> <tr> <tt> <var>
```

Both Block & Inline

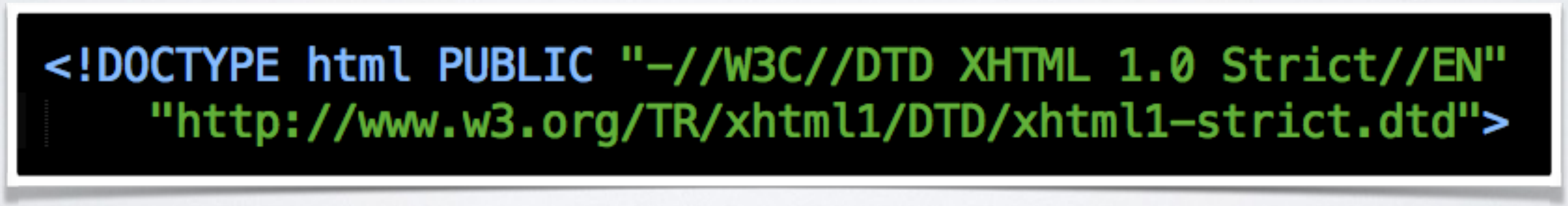
- Some elements can be both block and inline
- If used as inline then
  - They should not contain any block elements
- Only need to remember this exists — not tag names.

```
<button> <del> <ins> <map> <object>
<script>
```

Common Mistakes 1/2

Other things to avoid:

- **DO NOT** use the `name=""` attribute in tags, use the `id=""` instead.
- The quoted string that appears after the `public` keyword in the `doctype` declaration is case sensitive:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Common Mistakes 2/2

Other things to avoid:

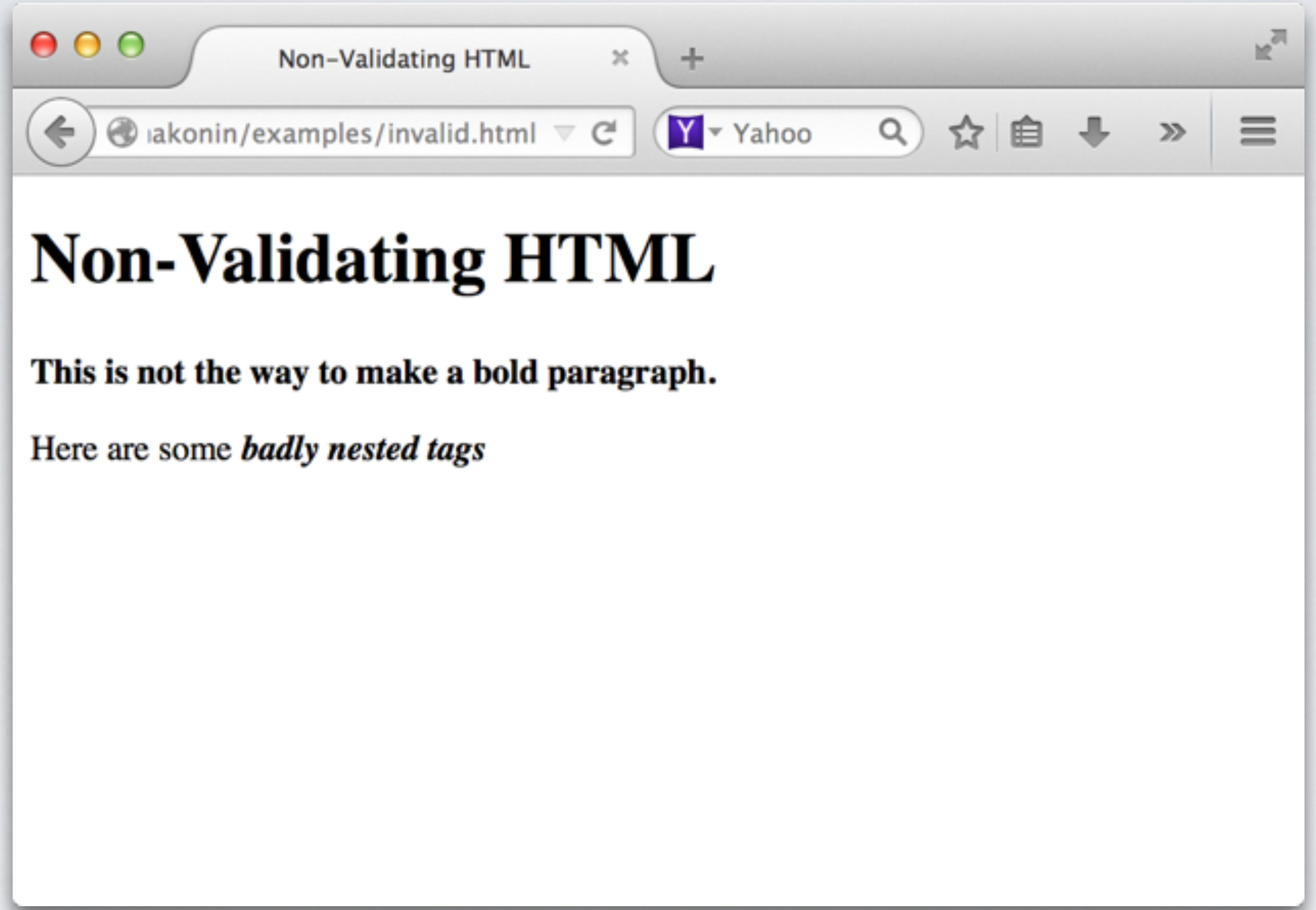
- The path part of a URL is also case sensitive.
- Missing `<title>` in the `<head>` element.
    - In `<head>`, `<mega>` and `<link>` are also OK.
    - Not other tags, e.g. `<h1>` should be in `<body>`
- Tag names and attribute names **MUST** be all lowercase:

```
<IMG Src="images/logo.png" Alt="logo"/>
```
❎

```
<img src="images/logo.png" alt="logo"/>
```
✅

Class Demo

Looking at invalid XHTML and using a validator:



Non-Validating HTML

This is not the way to make a bold paragraph.

Here are some ***badly nested tags***

URL: Invalid XHTML ⇒ Valid XHTML

QUESTIONS?

Character Entities

Character entities are used to display reserved or special characters in HTML.

- Display characters in our HTML not on the keyboard
- Some characters are reserved in HTML.
- Using the `<` or `>` signs will cause the browser to use your text content as tags

e.g. `&nbsp;` is non-breaking space

The HTML Entity

`&entity_name;` or `&#entity_number;`

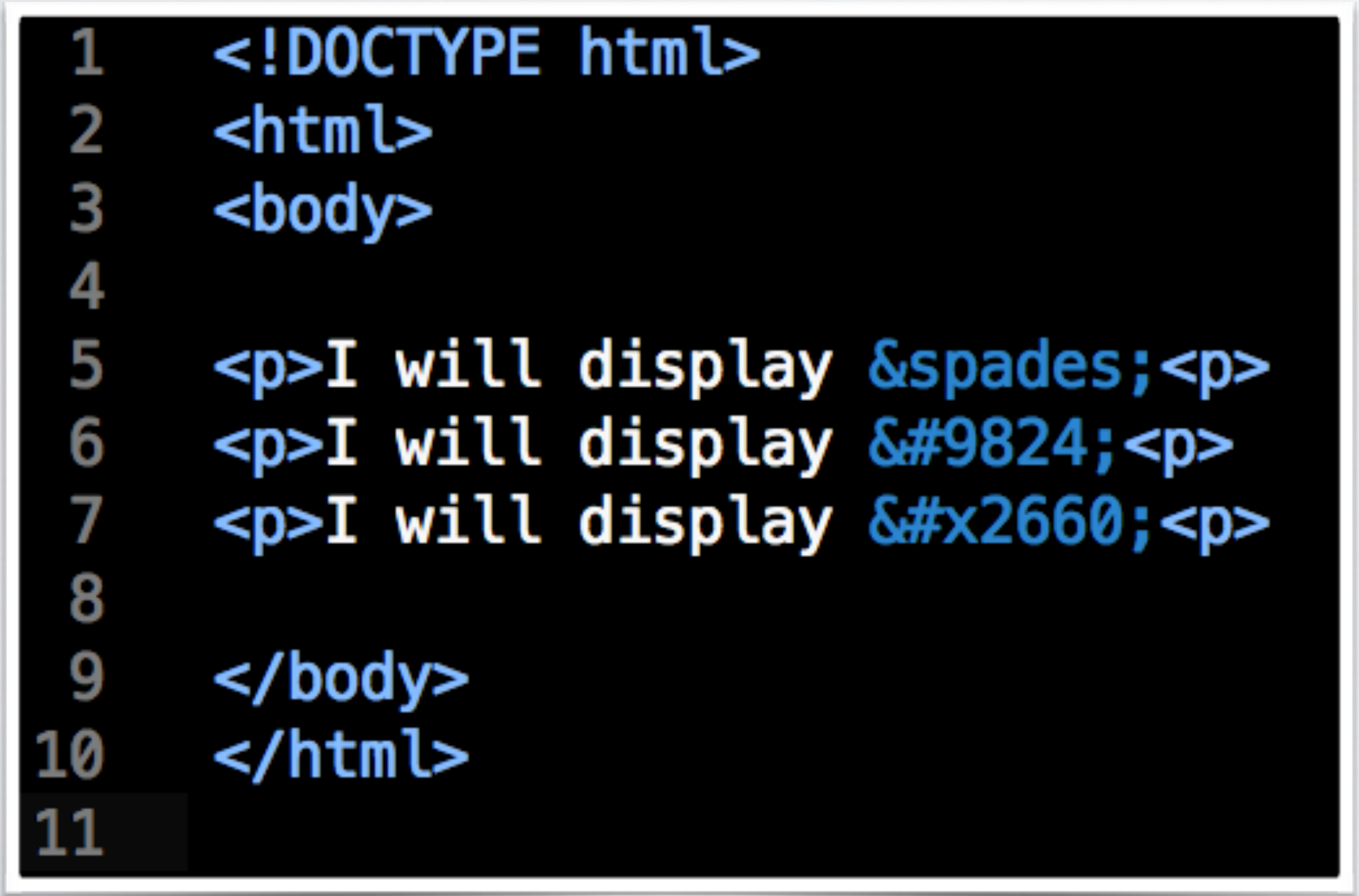
- Starts with either:
    - `&` for name
    - `&#` for decimal (dec)
    - `&#x` for hexadecimal (hex)
- Specify the entity and or number
- Specify the end with a semi-colon `;`

e.g. `&nbsp;` is non-breaking space

- Entity names are case sensitive (e.g. greek characters).

Entity: Name, Dec, Hex

- You can specify some entities 3 different ways: HTML **name**, decimal (**dec**), or hexadecimal (**hex**).

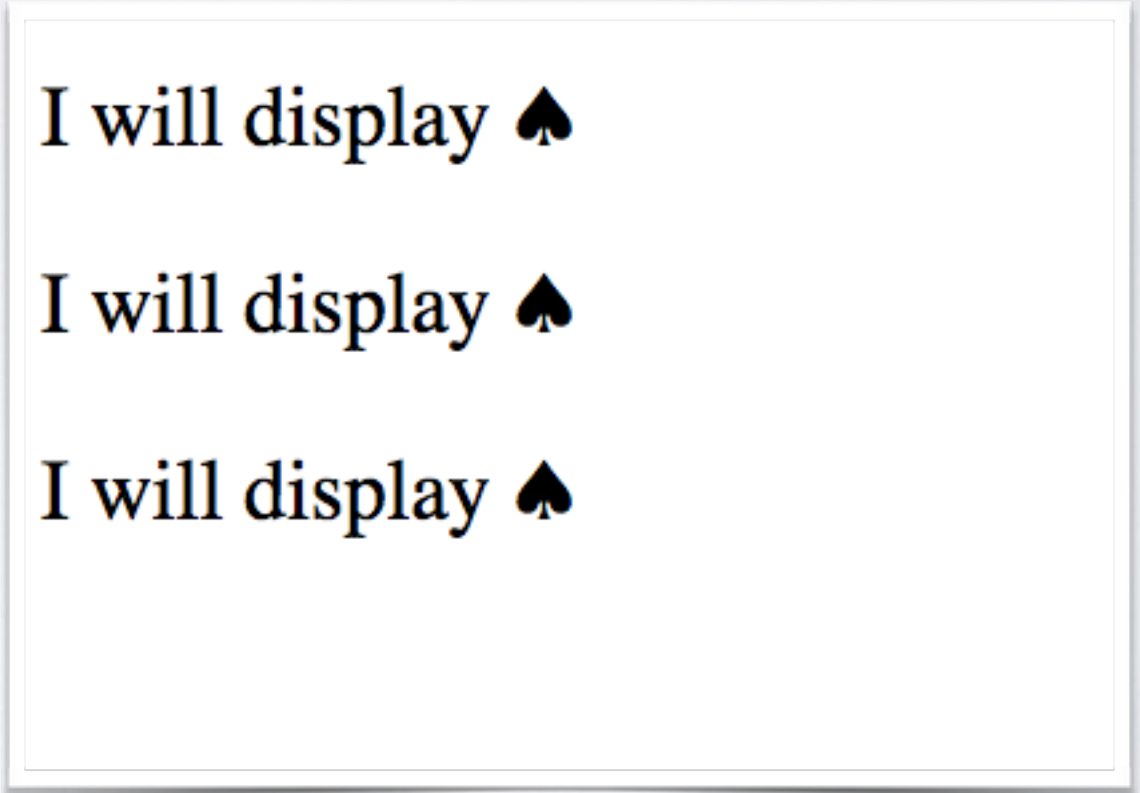


```html
<!DOCTYPE html>
<html>
<body>

<p>I will display &spades;<p>
<p>I will display &#9824;<p>
<p>I will display &#x2660;<p>

</body>
</html>

```

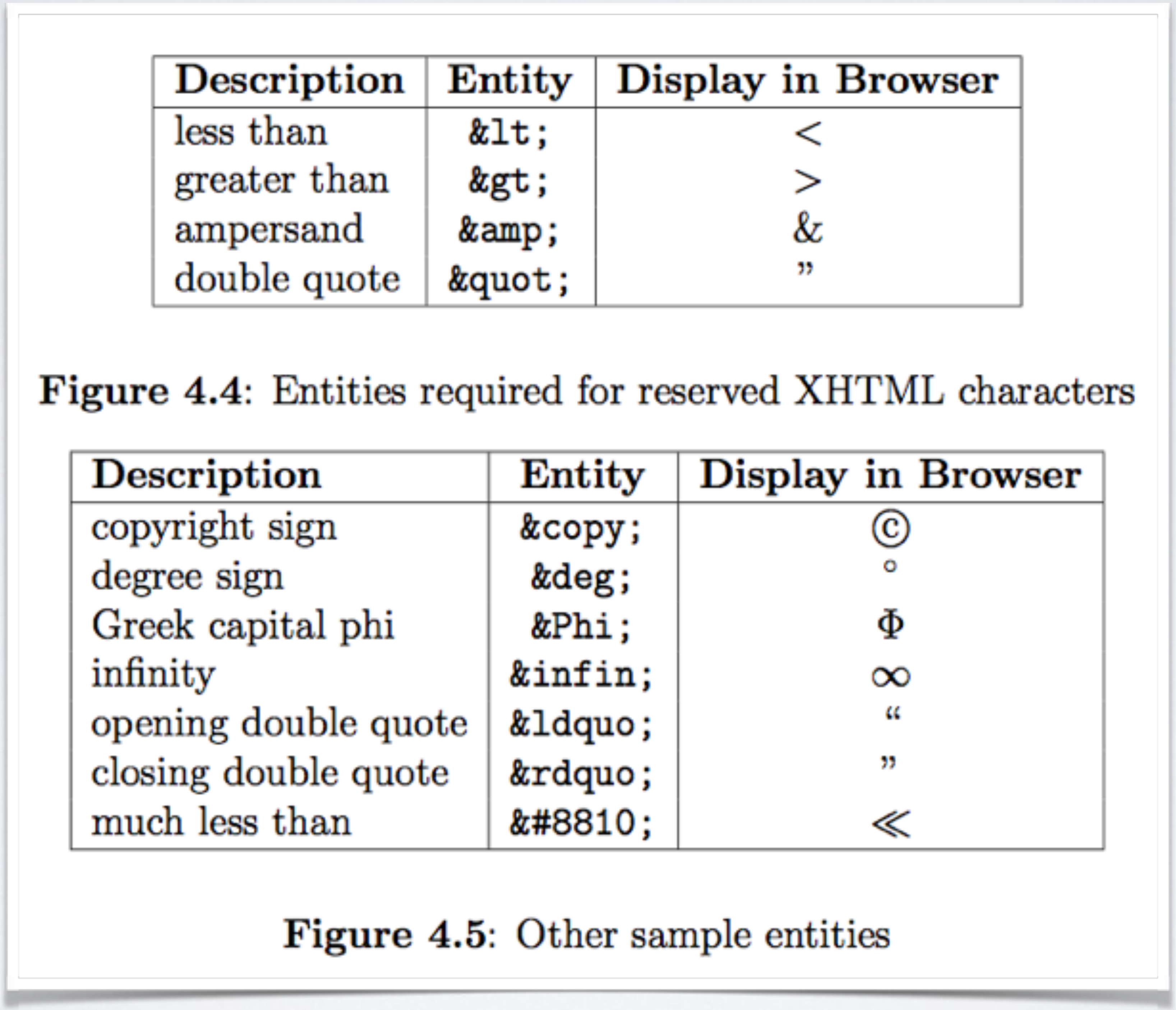


```
I will display ♠

I will display ♠

I will display ♠
```

Character Entities



Description	Entity	Display in Browser
less than	`&lt;`	<
greater than	`&gt;`	>
ampersand	`&amp;`	&
double quote	`&quot;`	”

Figure 4.4: Entities required for reserved XHTML characters

Description	Entity	Display in Browser
copyright sign	`&copy;`	©
degree sign	`&deg;`	°
Greek capital phi	`&Phi;`	Φ
infinity	`&infin;`	∞
opening double quote	`&ldquo;`	“
closing double quote	`&rdquo;`	”
much less than	`&#8810;`	≪

Figure 4.5: Other sample entities

Source: CMPT 165 Course Study Guide, p. 71

Character Entities

Mathematical Symbols

- http://www.w3schools.com/charsets/ref_utf_math.asp

Greek and Coptic Symbols

- http://www.w3schools.com/charsets/ref_utf_greek.asp

Currency Symbols

- http://www.w3schools.com/charsets/ref_utf_currency.asp

Arrows Symbols

- http://www.w3schools.com/charsets/ref_utf_arrows.asp

Miscellaneous Symbols

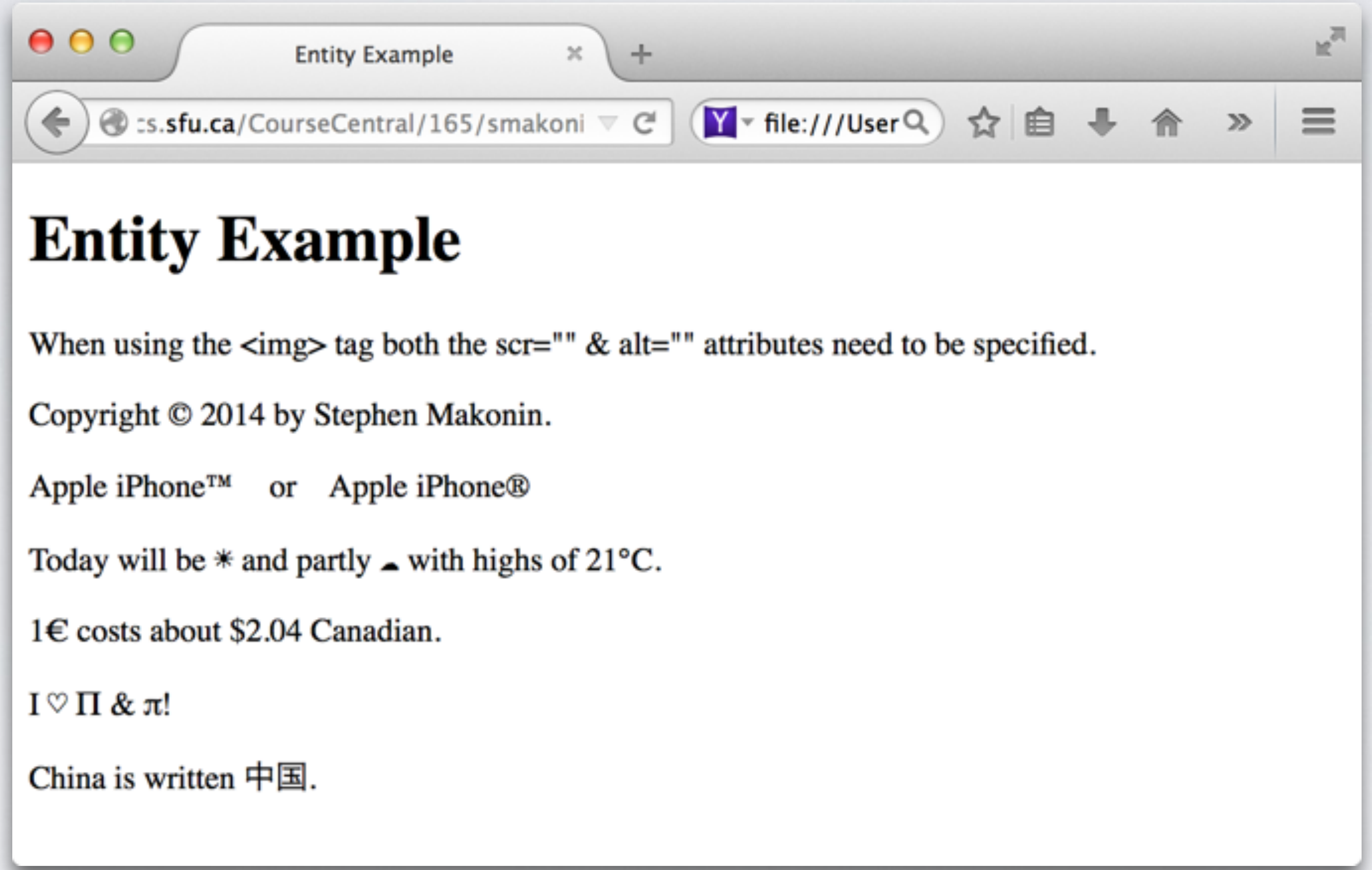
- http://www.w3schools.com/charsets/ref_utf_symbols.asp

Chinese Unicode Converter

- http://pages.ucsd.edu/~dkjordan/resources/unicodemaker.html

Class Demo

Looking at HTML entities using the editor:



Entity Example

When using the <img> tag both the scr="" & alt="" attributes need to be specified.

Copyright © 2014 by Stephen Makonin.

Apple iPhone™ or Apple iPhone®

Today will be ☀ and partly ☁ with highs of 21°C.

1€ costs about $2.04 Canadian.

I ♡ Π & π!

China is written 中国.

URL: entity.txt ⇒ entity.html

Generic Tags

Two generic tags `<div>` and `<span>`

- `<div>` is used for block elements
  - e.g. a list menu, contents.
- `<span>` is used for inline elements.

```
<div>
  This is a block of text and this is a
  <span>phrase in this block</span>.
</div>
```

Tag Identifiers

Uniquely identify and element by specifying the attribute `id=""` within the open tag.

```
<h1 id="title">content</h1>

<p id="abstract">content</p>
```

- ID **must** only be used once per page (for a given tag).
- Style rule examples:

```
#title { text-transform: uppercase; }

p#abstract { color: #F00; }
```

Tag Classes

Uniquely identify and element by specifying the attribute `class=""` within the open tag.

```
<h1 class="discussion">content</h1>

<p class="discussion">content</p>
```

- Class names can be used many times on a page.
- Style rule examples:

```
 .discussion { color: #00F; }

h1.discussion { color: #0F0; }
```

Selectors Revisited

Tag: selects all instances of that element.

```
h1 { color: F00; }
```

ID: selects the element with that ID

```
#title { color: FFF; }
```

Class: selects only element with that class name.

```
.discussion { color: #888; }
```

Contextual: selects elements in other elements.

```
ul ul { color: #0F0; }
```

Pseudo: selects a sub-class or sub-element.

```
a:link { color: #00F; }
```

RGB Colours

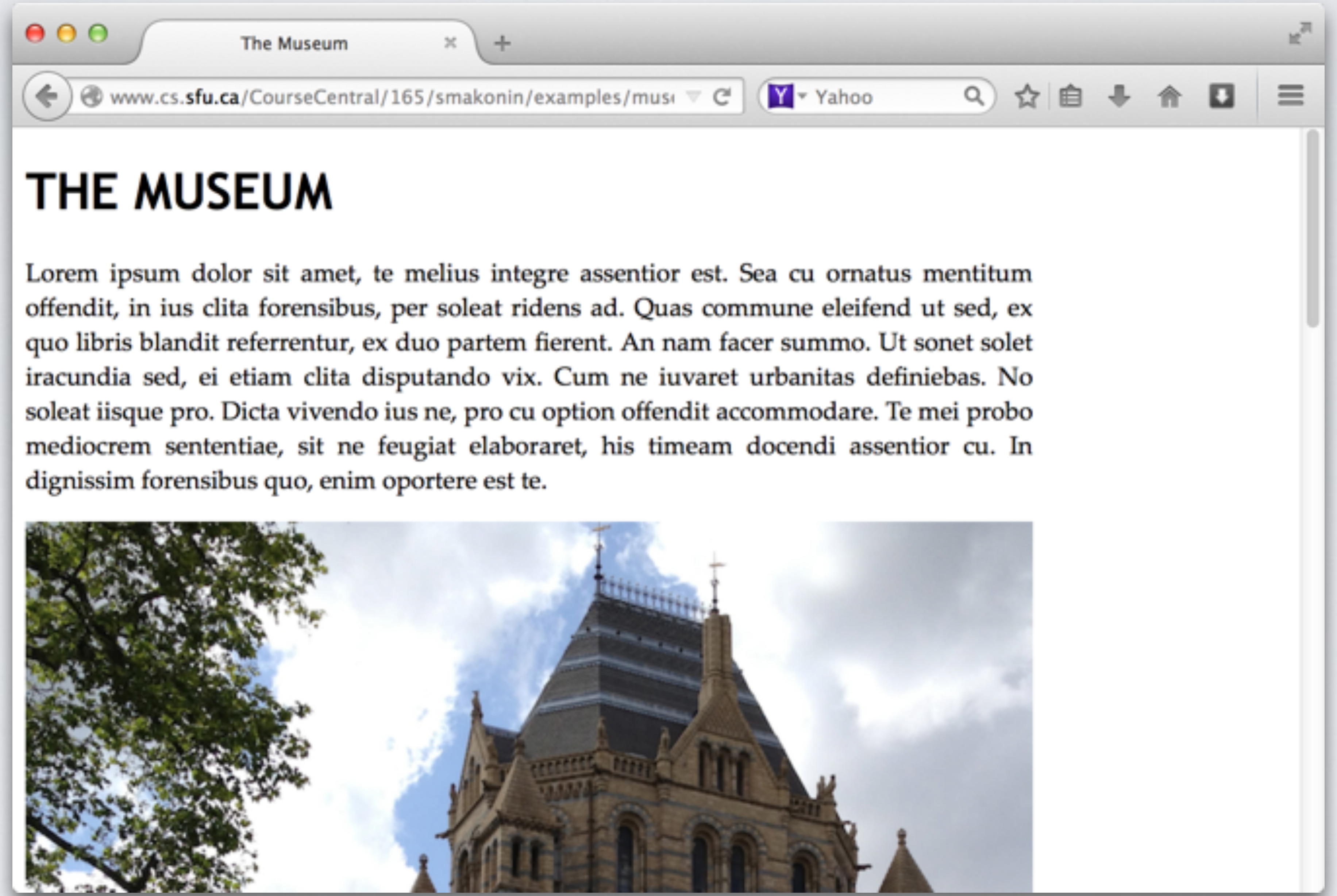
(RED, GREEN, BLUE)

🟥	#F00	🟩	#0F0	🟦	#00F
⬛	#000	■	#888	⬜	#FFF
🟨	#FF0	🟧	#F90	🟫	#963
🟪	#F0F	🟪	#609	🟦	#0FF

QUESTIONS?

Class Demo

Positioning images and text:



THE MUSEUM

Lorem ipsum dolor sit amet, te melius integre assentior est. Sea cu ornatus mentitum offendit, in ius clita forensibus, per soleat ridens ad. Quas commune eleifend ut sed, ex quo libris blandit referrentur, ex duo partem fierent. An nam facer summo. Ut sonet solet iracundia sed, ei etiam clita disputando vix. Cum ne iuvaret urbanitas definiebas. No soleat iisque pro. Dicta vivendo ius ne, pro cu option offendit accommodare. Te mei probo mediocrem sententiae, sit ne feugiat elaboraret, his timeam docendi assentior cu. In dignissim forensibus quo, enim oportere est te.

URL: Original Article ⇒ Final Article

Creating Websites

1. Start with a blank, valid XHTML file.
2. Create and link a blank CSS file.
3. Create new or markup existing content.
4. Add style rules that enhance your content.
5. Repeat steps 3 & 4.

Remember: creating a website is vary much like painting a picture — you iteratively add dabs of colour (in our case tags and style) until you have something that you like.

Summary

- Used a validator to validate XHTML.
- Discussed common mistakes make in HTML markup.
- Learnt about inline/block elements & character entities.
- Reviewed generic tags and style selectors.
- Reviewed how to position elements on a page.

Next Unit: learn more about graphics and images.

QUESTIONS?