# CMPT 165
# INTRODUCTION TO THE INTERNET AND THE WORLD WIDE WEB

## Unit 3
## Cascading Style Sheets (CSS)



SFU

# Learning Objectives

In this unit you will learn the following.

- **Use** CSS to style HTML content.
- **Describe** commonly used CSS properties.
- **Use** CSS to implement a simple visual design.

# Topics

1. Content *vs* Style
2. What is CSS?

**Lecture 1**

3. Why CSS?

4. Define Levels and Rules
5. Selectors and Properties

**Lecture 2**

6. Applying Basic Style
7. Adding Style to Report

**Lecture 3**

# Nice Webpage

| **Content** | + | **Structure** | + | **Style** |
|---|---|---|---|---|
| • Text | | • *Give Meaning* | | • *Enhance Message* |
| • Images | | • *HTML:* | | • *CSS:* |
| • Music | | • tags, e.g. **`h1`**, **`p`** | | • select tags |
| • Videos | | • attributes | | • style rules |
| • PDFs | | • hyper links | | • change visual properties, e.g. resize, colour |
| • *Other Resources* | | • meta data | | |

# Style

**HTML/XHTML** is the *structured content*.

Cascading Style Sheets (**CSS**) is the *visual appearance* of that content.

- Without defining style we rely on the browser to choose the appearance, e.g. report.
- Separating structured content from visual appearance is deliberate.
- Especially true when we use external CSS files.

# 3 Places to Define Style

**Mixing style with content!**

1. The `style=""` attribute in a tag.

```
style="width:640px;text-align:justify;"
```

2. The `<style>` tag is the `<head>`.

```
<head>
<title>THIS IS THE TITLE OF MY REP
<style type="text/css">
    h2 {
        text-align: center;
    }

    blockquote {
        font-size: smaller;
        font-style: italic;
    }
</style>
</head>
```

3. An external CSS file.

**the best way! Why?**

```
<head>
<title>Lab Instructions</title>
<link rel="stylesheet" href="mystyle.css" type="text/css" />
</head>
```

# New Product Example

Say we want to launch of a **new product**

- **new website content** needs to be generated

This results in:

- existing webpage content needs to be modified
- new webpages and resources need to be added

**BUT**

- the website style is need not be changed

**WHAT HAPPENS WHEN STYLE AND CONTENT ARE MIXED?**

see SparkFun

# Style in Content

```
<body style="width:640px;text-align:justify;">
<hr/>
<center>
```

If style and content are mixed then

- every time a new webpage is added or modified,
- every time there is a structural change to content,
- we need to add/modify the affected tag's `style`.

If we decide to change the appearance of (say) `<h1>`:

- for every `<h1>` in every HTML file we
- need to change the value of the `style=""` attribute.
- Same if style is define in the `<head>` but not as bad.

# Separating Style

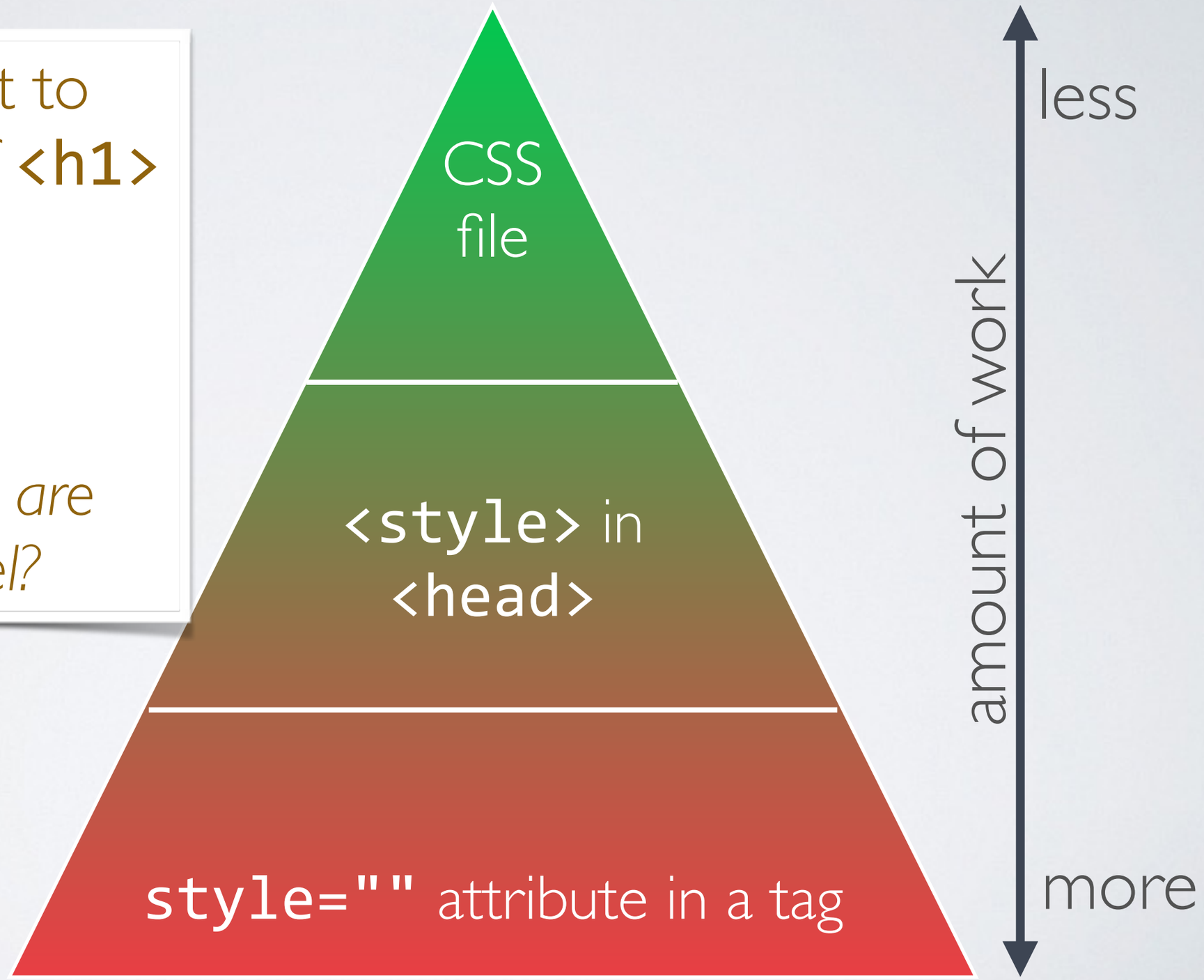
- Makes designing and later modifying a design easier.
- Content and its structure can also be change easily.
- Technical writing (content) and designing (style) are:
  - two separate functions of creating a website,
  - involve two different type of jobs — specialization,
  - involve two people, one for each task,
  - content and design and be created in parallel.

# Amount of Work to Make Style Changes

**Example:** we want to change the style of `<h1>`

- 40 webpages
- 100 `<h1>`'s

*How many changes are needed at each level?*



# Deprecated Tags

*def.* no longer used

Because of CSS:

- Some tags have been deprecation:
  - e.g. `<center>` can be replaced with

```
<div style="text-align:center;">content</div>
```

    - `<div>` is for division or a section (more than `<p>`)
  - Use `<strong>` instead of `<b>` (bold)
    - Using bold fonts is a way to look strong.
  - Use `<em>` (emphasis) instead of `<i>` (italic).
    - Using italic fonts is a way to emphasize text.

# Deprecated Attributes

Because of CSS:

- Many tag attributes have been **deprecated**, too!
  - e.g. `<hr/>` (horizontal rule):
    - attributes ~~`align`, `noshade`, `size`, `width`~~ ***GONE!***
    - `align = "left|center|right "` on the page.
    - `noshade = "noshade "` render in one solid color.
    - `size = "pixels "` height of the rule.
    - `width = "pixels|%"` width of the rule.

# Reasons to Use CSS

- More options for controlling the appearance of pages compared to old HTML designs.
- Defining style once in an external file produces smaller files for faster transfer times.
- Forces authors to separate the meaning of the content from its appearance for better design.
- Easy to update the appearance of an entire site with external CSS files.
- Allows you to apply different styles at different times
  - e.g. having a halloween theme in October.
- Easier for searcher engine to read and parse content.

# QUESTIONS?

# Having No Style

In many browsers we can **turn off styles**. What do sites look like without style?

CMPT 165 Course Website

Microsoft

Google Search

# Levels of CSS?

- A level in CSS is equivalent to a version.
- CMPT uses CSS **Level 1** or Version 1 (Dec 1996).
- There many other levels:
  - **Level 2** was released in May 1989.
  - **Level 2.1** was released in June 2011.
  - **Level 3** is being worked on (split into 50 modules).
  - **Level 4** is planned for the future.

# CSS 1

- The first official CSS specification by W3C
- Style support for:
  - **Font** properties, e.g. typeface and emphasis.
  - **Colour** of element text, backgrounds.
  - **Text** attributes, e.g. spacing of words, letters, and lines.
  - **Alignment** of elements (text, images, etc.)
  - **Boxing**: margin, border, padding, and positioning.
  - **Unique identification** of elements.
  - **Generic classification** of by tag name.
- The W3C no longer maintains the CSS 1.

# Style Rules

```
selector { property: value; ...}
```

Each `property:value;` is a *declaration*. Each rule can have 1 or more declarations.

```
h1 {
   text-align: center;
   color: red;
}
```

For every `<h1>` have a horizontal alignment of centre with red coloured text.

# Style Grouping

Group style rules by selector:

```
h1, h2, h3 { font-family: helvetica }
```

Declarations can be grouped:

```
h1 {
  font-weight: bold;
  font-size: 12pt;
  line-height: 14pt;
  font-family: helvetica;
}
```

Grouping of some properties:

```
h1 { font: bold 12pt/14pt helvetica }
```

# Style Inheritance

In the following HTML:

```
<h1>
    THE <em> REPORT</em> TITLE
</h1>
```

The **** element would be styled similar to **<h1>** with red coloured text if the following style rule was declared:

```
h1 { color: red; }
```

# Class as Selector

When a using the `class` attribute:

```
<h1 class="title">THE REPORT TITLE</h1>
```

We can define a style for that class of element:

```
.title { color: red; }
```

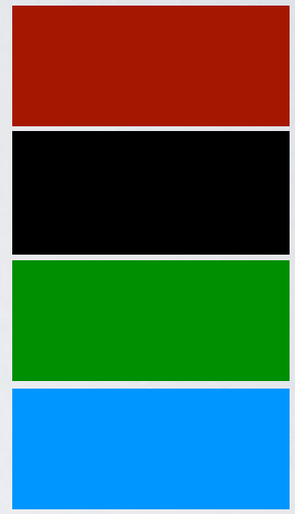
Resulting in:

**THE REPORT TITLE**

# ID as Selector

When a using the `id` attribute:

```
<h1 id="main">THE REPORT TITLE</h1>
<h1>1. INTRODUCTION</h1>
<p id="main">In summary ...</p>
<h2 id="main">1.1 ORGANIZATION</p>
```



We can define a style for that class of element:

```
#main    { color: blue; }
h1#main { color: red; }
p#main  { color: green; }
```

# Contextual Selectors

```
<h1>THE <em> REPORT</em> TITLE</h1>
```

If we want all all emphasized on our web page to be red:

```
h1 { color: blue; }
em { color: green; }
```

If we only what red emphasized text in **<h1>** then we:

```
h1 em { color: red; }
```

The emphasized text outside **<h1>** would be black.

**THE *REPORT* TITLE**

# Anchor Pseudo-Classes

Some elements have like `<a>` have specific sub:

```
a:link    { color: red;
          text-decoration: none;}
a:visited { color: blue; }
a:active, a:hover
          { text-decoration: underline; }
```

For `<a href="#">Go To Top</a>` this would result in:

Go To Top

Go To Top

Go To Top or Go To Top

# Typographical Pseudo-Elements

Some elements have like `<p>` have specific sub elements:

```
p:first-line { font-variant: small-caps }
p:first-letter { font-size: 200%; float: left }
```

This would result in:

LOREM IPSUM DOLOR SIT AMET, MUTAT VIRTUTE INTELLEGAT EU MEL, LUDUS dolore nec ad. Agam hendrerit in cum. Atqui ubique scripserit duo id. Vel reque iracundia interesset ea. Postea verear posidonium ei vel, in ius tation accommodare.

# Cascading Order

**How does the browser decide what style rule to use?**

Styles will *cascade* into a new *virtual* style sheet by inheritance and by ordered the replacement of conflicting rules or multiple rules with the same selector.

1. Browser's default
2. External CSS file
3. Internal style sheet (in the `<head>` section)
4. Inline style (using the `style` attribute)

So, (4) would override (3), (3) overrides (2) and so on.

# Font Properties

**font-family** - what type of font you want to use

e.g. `font-family: gill, helvetica, arial;`

**font-style** - can be `"normal | italic | oblique"`

**font-variant** - can be `"normal | small-caps"`

**font-weight** - can be `normal | bold | bolder | lighter | 100...900` (e.g. 700 = bold)

**font-size** - can be a absolute/relative-size, length, or %

**font** - short hand to combine all in one line, e.g.

```
font: normal small-caps 120%/120% arial
```

# Colour & Background Properties

**color** - the text colour of an element (more in Unit 4)

**background-color** - background colour of an element

**background-image** - have an image as the background

**background-repeat** - have the image repeat, fi small.

**background-attachment** - fixed to canvas or element.

**background-position** - position within the element, e.g.

```
background-position: top right;
```

**background** - short hand to combine all in one line, e.g.

```
body { color: black; background: white; }
```

# Text Properties

**word-spacing** - the space between words, e.g. `1em`

**letter-spacing** - the space between characters

**text-decoration** - can be

`"underline|overline|line-through|blink"`

**vertical-align** - e.g. `"sub|super|top|middle|bottom"`

**text-transform** - can be

`"capitalize|uppercase|lowercase|none"`

**text-align** - can be `"left|right|center|justify"`

**text-indent** - indentation that appears before the first line

**line-height** - line height in number, length, or %

# Everything Boxed


**Content** - The content of the box, where text and images appear

**Padding** - Clears an area around the content. The padding is transparent

**Border** - A border that goes around the padding and content

**Margin** - Clears an area outside the border. The margin is transparent

See http://www.w3schools.com/css/css_boxmodel.asp

# Box Properties

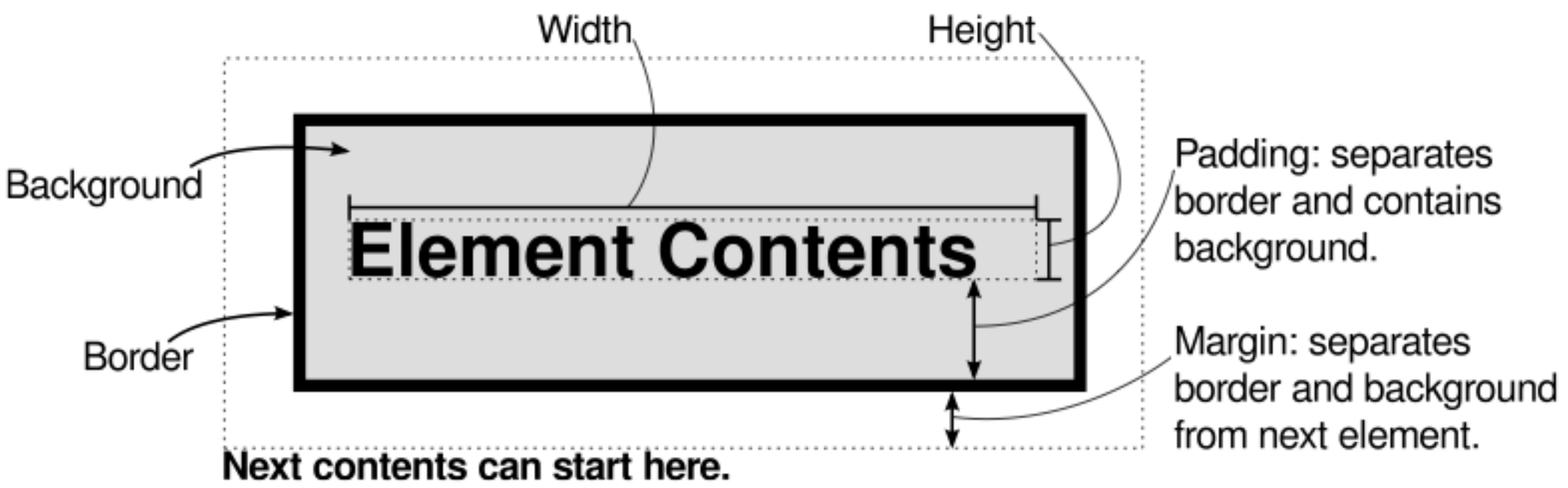



**Figure 3.6**: The CSS "Box Model"

Source: CMPT 165 Course Study Guide, p. 58

# Box Properties

**margin:** margin-top, margin-right, margin-bottom, margin-left

**padding:** padding-top, padding-right, padding-bottom, padding-left

**border:** border-color, border-style, border-top, border-right, border-bottom, border-left

**border-width:** border-top-width, border-right-width, border-bottom-width, border-left-width

**width, height:** of text or image, images are scaled

**float:** where it will appear in the text `"left|right|none"`

**clear:** allows floating elements are not allowed, can be `"none|left|right|both"`

# Boxed Example

```
<div>
In this course, we shall
examine the structure of
the Internet and the
World Wide Web as well as
design and create web
sites.
</div>
```

**Style Rule:**

```
div {
    width: 300px;
    padding: 25px;
    border: 25px solid navy;
    margin: 25px;
}
```

**Results in ⇒**

In this course, we shall examine the structure of the Internet and the World Wide Web as well as design and create web sites.

# Classification Properties

Can specify how and where elements are displayed:

**display:** how an element is displayed,
can be `"block|inline|list-item|none"`

**white-space:** how whitespace inside the element is handled, can be `"normal|pre|nowrap"`

**list-style-type:** was the `type=""` attribute for lists

**list-style-image:** use an image as the bullet point

**list-style-position:** how the bullet point is drawn `"inside|outside"` of content

**list-style:** the one liner for above 3 properties

# Units: Length

Specify the length of an element in absolute units:

`h1 { margin: 0.5in; }` - inches, 1in = 2.54cm

`h2 { line-height: 3cm; }` - centimeters

`h3 { word-spacing: 4mm; }` - millimeters

`h4 { font-size: 12pt; }` - points, 1pt = 1/72 in

`h4 { font-size: 1pc; }` - picas, 1pc = 12pt

Specify the length of an element in relative units:

`h1 { margin: 0.5em; }` - ems, height of the font

`h1 { margin: 1ex; }` - the height of the letter 'x'

`h1 { font-size: 9px; }` - pixels, relative to canvas

`img { width: 120%; }` - 120% of the body width

# Units: Colour

**Specify the colour by name:** aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow, e.g. `h1 { color: maroon; }`

**Specify RGB value:** stands for red, blue, green

`h1 { color: #f00; }` - #rgb

`h1 { color: #ff0000; }` - #rrggbb

`h1 { color: rgb(255,0,0); }` -
integer range 0 - 255

`h1 { color: rgb(100%, 0%, 0%); }` -
float range 0.0% - 100.0%

# Hex Numbering

```
Dec: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Hex: 0 1 2 3 4 5 6 7 8 9  A  B  C  D  E  F
```

Hexadecimal number **FF** = decimal number **255**, hence the integer range 0 - 255 in the example:

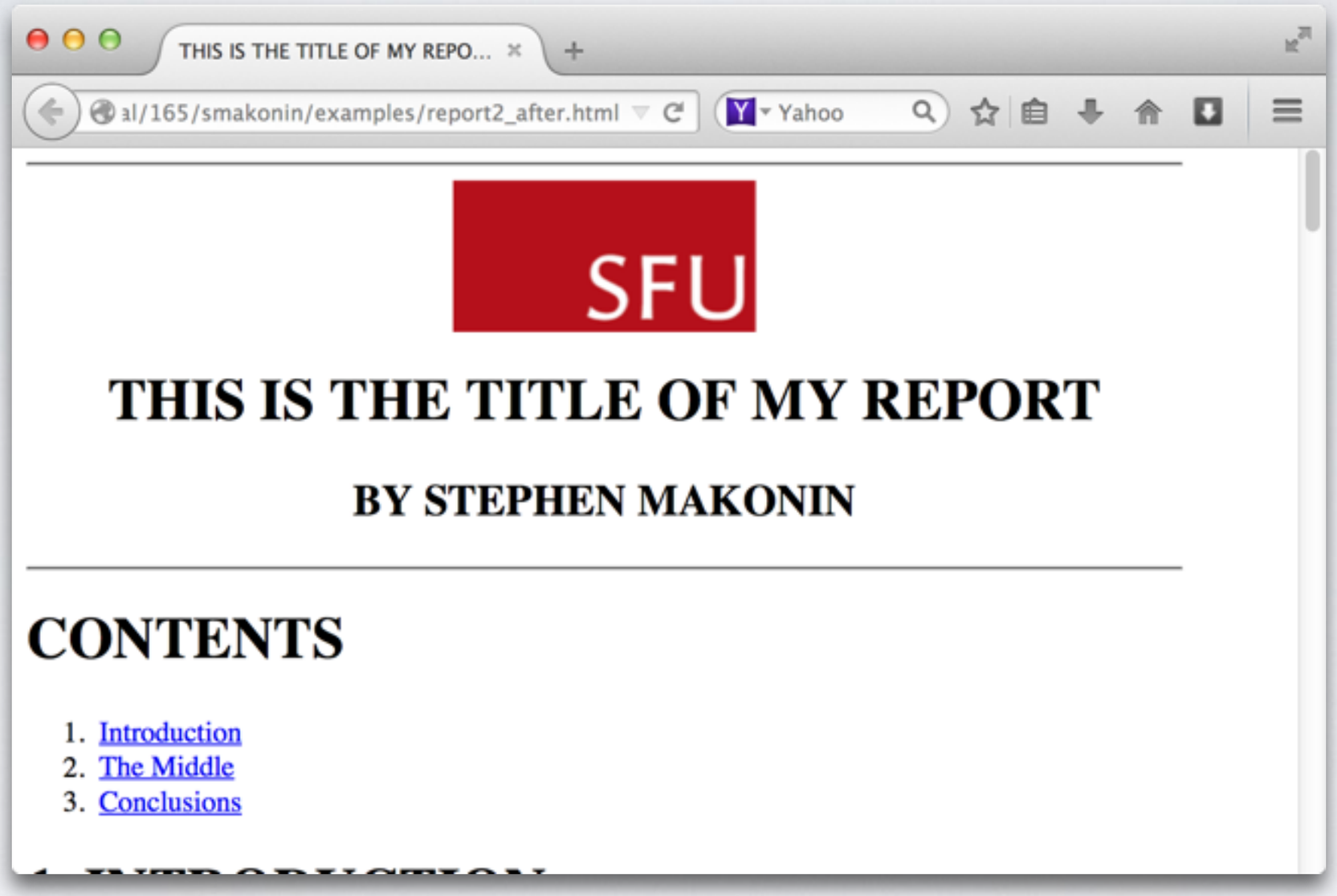
```
color: rgb(255,0,0);
```

**THIS IS IMPORTANT TO REMEMBER!!!**

# QUESTIONS?

# Class Demo

Adding some style to Report 2




URL: Report2 ⇒ Report4

# Summary

- Looked at the reasons why CSS and style are important.
- Apply cascading style sheets to an XHTML page/site.
- Learnt about style selectors, properties, and values.
- Create CSS files that make visual changes to a page.
- Experiment with the CSS selectors and properties.

**Next Unit:** brining XHTML and CSS together, do more advanced web design.

# QUESTIONS?