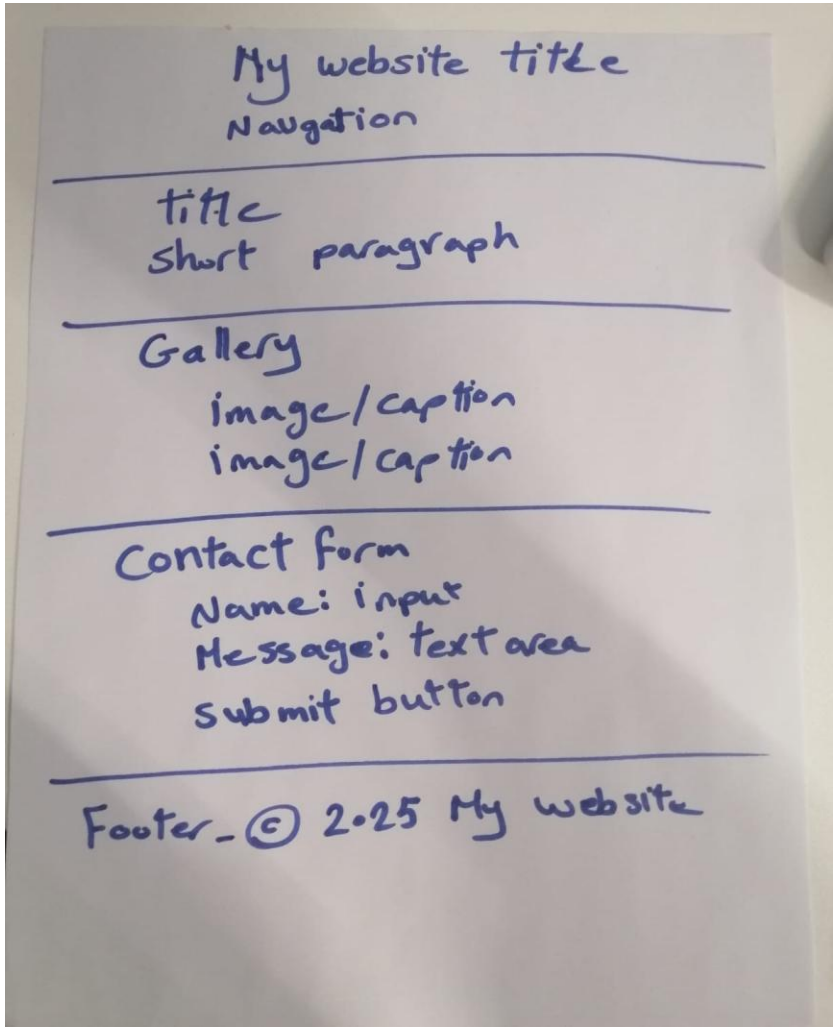
CMPU 1031 - Web Development 1

Lecturer: Dr Fatmaelzahraa Eltaher

Week 3: Introduction to CSS

Practice

```
-------------------------------------------------------
| My Website title                                    |
| Navigation:  Home | Images | Contact                |
|-----------------------------------------------------|
|  Title                                              |
|   A short paragraph of text.                        |
|-----------------------------------------------------|
| Gallery                                             |
|   [ image ]   [ image ]                             |
|-----------------------------------------------------|
| Contact form                                        |
|   Name: [input]                                     |
|   Message: [textarea]                               |
|   Submit button                                     |
|-----------------------------------------------------|
| Footer - © 2025 My Website                          |
-------------------------------------------------------
```



Overview

- Introduction to CSS
- CSS structure and syntax
- Advantages of using CSS
- Styling elements using CSS
- Different ways to style using CSS
- CSS Selectors

CSS

- **C**ascading **S**tyle **S**heets (CSS)
- CSS is **not** a programming language as there is not logic involved.
- It is also **not** a mark up language like HTML.
- CSS is a **styling** language
- Used to describe how the HTML will display on the web page.
- By using external CSS files, multiple pages can be styled together.

CSS

- CSS has properties for controlling basic formatting such as **font size** and **colour**.
- CSS can be used for layout properties such as **positioning**.
- CSS has a number dynamic properties allowing items to **appear** and **disappear** as desired.
- The dynamic properties can be used to create menu structures and drop-down menus.

CSS

- CSS is flexible, powerful and efficient.
- CSS can provide performance gains for a website.
- Keeping CSS is a separate file will only be loaded once by a browser and then cached. If this CSS file is required again it can be taken from the browser cache, it does not need to be downloaded each time.

Separation of Concerns (SoC)

- For **very** simple examples, it is not uncommon to see HTML, CSS, and JavaScript positioned in the same .html file.
- However, in a real-life scenario, the code for website building will be definitely complex
- **Separation of concerns** is the idea that each layer in an application should only be responsible for one task, and should not contain code that deals with other tasks.
- In that case, it is **highly recommended** that you separate HTML, CSS, and JavaScript in different files and link them together using the tag <link>

Separation of Concerns (SoC)

- If your website has multiple pages (such as Home, News, Contact), you can use the same CSS file to style all of them
- Adopting the CSS makes it easier to change the website style – if you change the colour of the background in one page, all pages will change (making the website **maintainable**)
- Adopting the CSS saves time and space – there is no need to write a new CSS code for each new page you add to your website (making the website **extendible**)
- Adopting the CSS allows the browser to keep information about the style of that page in the browser memory. If the user accesses the page again, the CSS does not need to be downloaded each time (making the website **faster**).

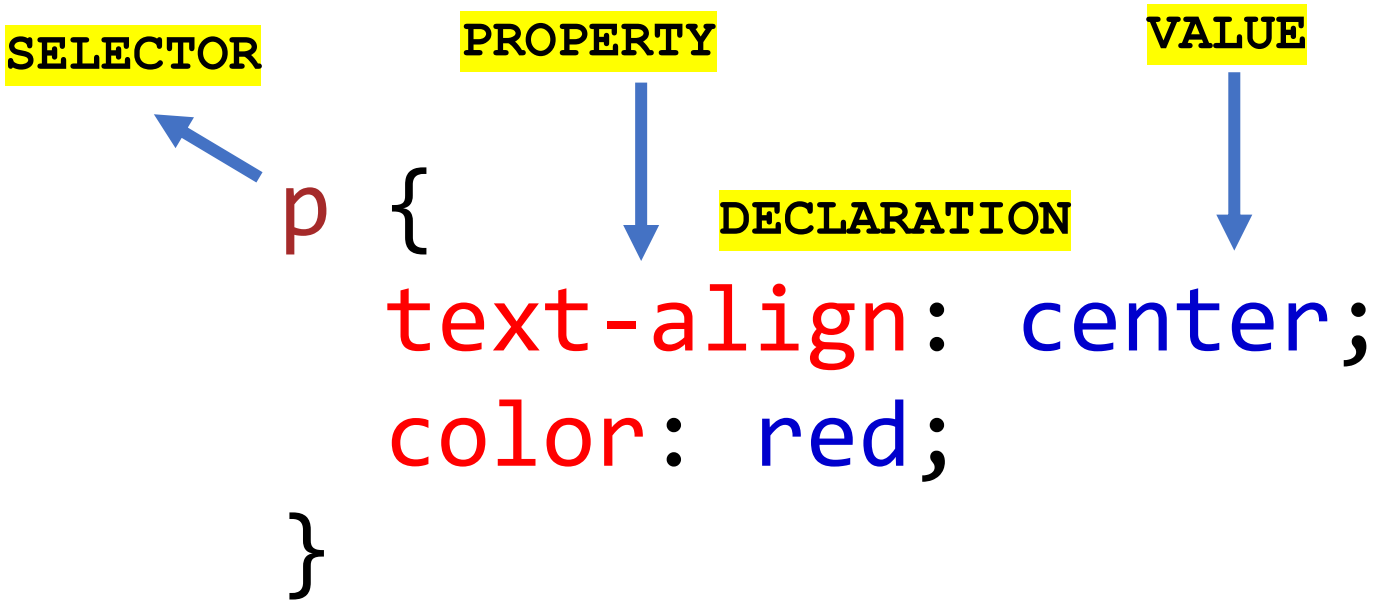
CSS Syntax

- CSS is a rule-based language. You need to list the styling rules you want to apply to one or more HTML elements.
- In CSS the **selector** is used to identify the element to be styled.
- For each selector, there will be a block that will contain one or more declarations.
- Declarations are the **CSS rules** used to style the given element.
- Each declaration must be followed with a **semi-colon** (;)

CSS Syntax – rules

- Each CSS style declaration is made up of a property and a value.
- The property and the value are separated by a colon (:).
- Each declaration must be followed with a semi-colon (;).
- This follows the format of a **name: value** pair.
- The property (name) identifies the **feature** of the element that requires styling, the **value** describes how this will display on the page.

CSS Syntax


SELECTOR
PROPERTY
VALUE
DECLARATION
p {
text-align: center;
color: red;
}

CSS Syntax – selecting multiple elements

To apply the same rules to multiple elements, declare them at the selection block, separated by a comma.

Styling a paragraph

HTML

```
<p> This class is great! </p>
```

CSS

```
p {
  color: green;
  text-align: center;
}
```

This class is great!

Click on the code to see the example in CodePen

Styling a heading

HTML

```
<h1>This should be a large text but I will change it with CSS. I will also make it red.</h1>
```

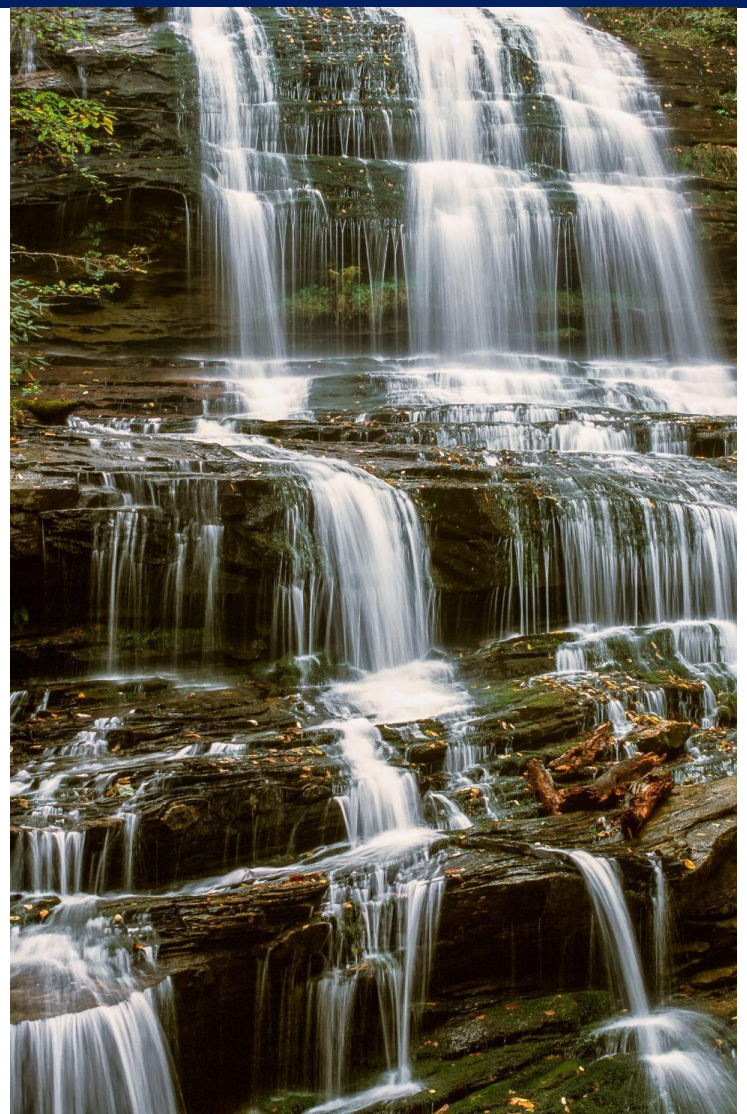
CSS

```
h1 {
  color: red;
  font-size: 50%;
}
```

This should be a large text but I will change it with CSS. I will also make it red.

Click on the code to see the example in CodePen

The concept of cascade



- CSS is considered a cascading language because there could be more than one stylesheet declaration being applied.
- The way the algorithm behind it selects what it is going to present takes place as a cascade (a waterfall with steps), by cascading down from general declarations to specific ones.
- https://codepen.io/fatma_eltaher/pen/wBMKxNB

Inline CSS

- Inline CSS is used to style an individual element in a HTML page.
- The CSS rule is added to the HTML page using the style attribute.
- The rule can contain any CSS property.

This is an example of inline CSS

Click on the code to see the example in CodePen

Internal CSS

- An internal style sheet can be used if one single page has a unique style.
- The internal style rules are contained in the head section of the webpage within the <style> element.

HTML

```
<head>
  <style>
    p {
      color: red;
      text-align: right;
    }
  </style>
</head>
<body>
  <p>This is an example of internal CSS styling</p>
</body>
```

This is an example of internal CSS styling

Click on the code to see the example in CodePen

External CSS

- External style sheets are a great way to style all the pages on a website.
- Using one style sheet for a website can give a common look and feel.
- All of the style rules are defined in a single style sheet and each webpage is told to use this style sheet to style the elements on the page.
- External style sheets are defined using the HTML tag **<link>** inside the HTML tag **<head>**. The attribute **rel** specifies the relationship between the current document (HTML file) and the linked document (CSS file)

```
<head>
    <link rel="stylesheet" href="mystyle.css">
</head>
```

External CSS

Site.css

```
h1 {font-size: 14; color: red;}
h3 {font-size: 12; color: blue;}
```

The external style sheet

CSS3.htm

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<link rel="stylesheet" type="text/css" href="Site.css"  />
    <title>templates</title>

</head>

<body>

<h1>This is heading 1</h1>
<h3>This is heading 3</h3>
</body>
</html>
```

The source for the HTML document

Adding CSS to HTML

- Cascading Style Sheets come in three types:
  - internal, external, and inline.

- The concept of cascading in CSS can be a bit complex, and we want to keep things simple. For now, keep in mind that the basic rules are:
  - First, the Inline CSS using the attribute 'style' of the HTML element is considered
  - Second, an internal style sheet using the <style> HTML element is considered
  - Third, rules on an external CSS file (linked to the HTML file) are considered
  - Fourth, rules coming from the web browser (such as Chrome or Mozilla extension) are considered

Practice

In an External CSS File (style.css), Apply the Following Rules:

- Make the <h1> centered and red.
- Change all <p> text to blue and make font size=18px

```
<!DOCTYPE html>
<html>
<head>
  <title>Portfolio</title>
</head>
<body>
  <h1>My Portfolio</h1>
  <p>Welcome to my personal page.</p>
  <ul>
    <li>Home</li>
    <li>Projects</li>
    <li>Contact</li>
  </ul>
</body>
</html>
```

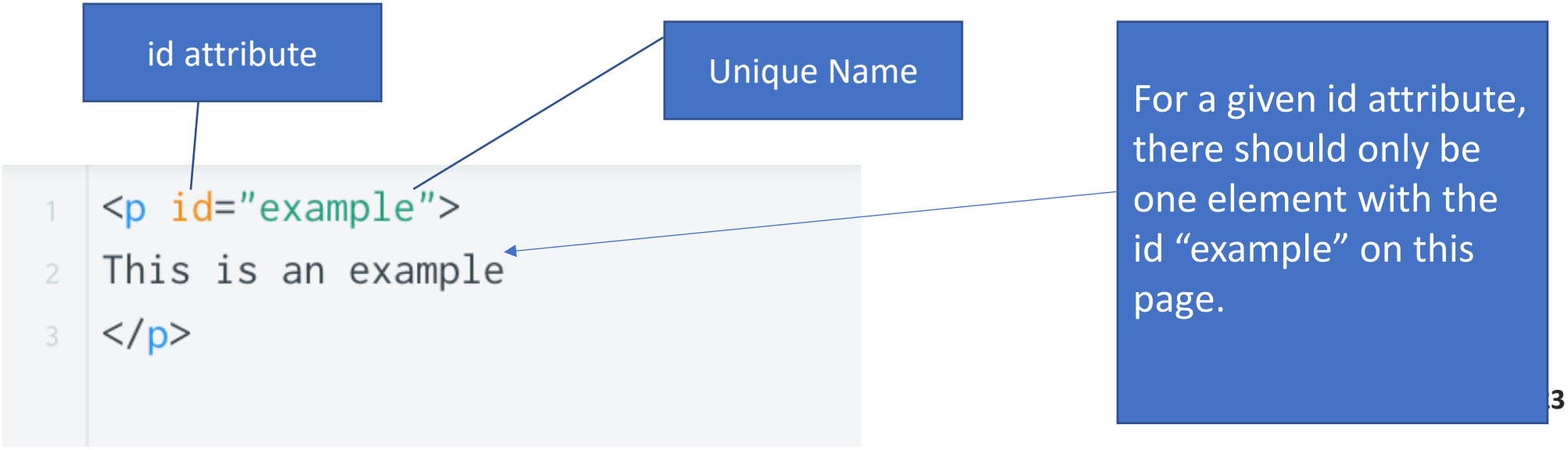
CSS Selectors

- The most basic selector to use is the element name. This will style the content of every occurrence of this element using the one-style rule. The example below will create a rule to style every paragraph on a page.

```
1  p {
2    color: green;
3    font-size: 20px;
4  }
```

ID Selector

- The id selector can be added to an element so that the element can be **uniquely identified** on a page.
- An attribute called **id** is added to the element opening tag and should have a meaningful value.
- ID names have to be unique so one ID can only be associated with one element.



ID Selector

- When an element has been given an id attribute, a CSS rule can be written for this id.
- In the stylesheet, a hash symbol (#) must be added before the ID when you use it as a selector.

HTML

```
<p id="uniqueParagraph">This paragraph has an ID
associated to it.</p>
```

CSS

```
#uniqueParagraph {
  font-family: "Helvetica";
  color: magenta;
  font-size: 20px;
}
```

This paragraph has an ID associated to it.

Click on the code to see the example in CodePen

Class Selector

- When an element has been given a class attribute, a CSS rule can be written for this.
- However, differently from ID, the name given to the class **does not** have to be **unique** to the page.
- There can be multiple elements with the same class attribute on the same page. This can be used to style **multiple occurrences of the same element** on the page
- In the stylesheet, a period symbol (.) must be added before the ID when you use it as a selector.

Class Selector

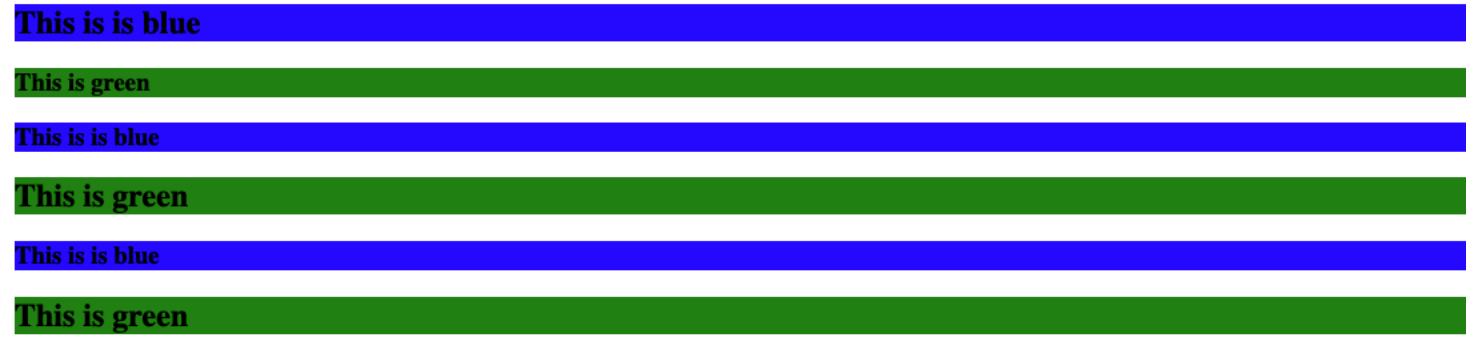
HTML

```
<h2 class="blueHeading">This is is blue</h2>
<h3 class="greenHeading">This is green</h3>
<h3 class="blueHeading">This is is blue</h3>
<h2 class="greenHeading">This is green</h2>
<h3 class="blueHeading">This is is blue</h3>
<h2 class="greenHeading">This is green</h2>
```

CSS

```
.greenHeading{
  background-color: green;
}

.blueHeading{
  background-color: blue;
}
```



Click on the code to see the example in CodePen

```html
<!DOCTYPE html>
<html>
<head>
  <title>My Webpage</title>
  <style>
    p { color:blue; }
    .highlight { background-color:yellow; }
  </style>
  <link rel="stylesheet" href="styles2.css">
</head>
<body>
  <h1 style="color:red;">Welcome to My Page</h1>
  <p class="highlight" id="main-text">Hello, world!</p>
  <p>Hello again!</p>
</body>
</html>
```

Practice

- Which styling method is being used in each part (inline, internal, external)?
- Move all styles into a separate external CSS file (styles2.css).