

IT 105 – Principles of Programming

Day 09

1. Just a reminder, when you save your pages make sure they are text, but that they have the .html extension and not .txt. As you modify your pages to add functionality, be sure to save separate copies.
2. A reminder regarding collaboration:
 - a. Discussing high-level concepts with others is fine.
 - b. If there is any collaboration, the students involved should all be doing their own typing/coding.
 - c. Any of these collaborations should be noted in a comment in the code.
 - d. ***Under no circumstances should code/files be shared/mailed between students. Taking someone else's file and "changing a few things around" is not acceptable.***
3. Cascading Style Sheets (CSS)
 - a. inline
 - b. style section in <head> element
 - c. separate .css file
4. Add style to sheets, with additional attributes in tags. The goal is to separate the content of the webpages from their style.
5. Text color (here in a heading), <h1 style="color:blue;">
 - a. style is the **attribute**, color is a **property**, blue is the **value**
6. Text color (here in a paragraph), <p style="color:blue;">
7. Rather than adding this to each element, could make global, to everything in the body:
 - a. <body style="color:blue;">, everything nested under body
 - b. Note that you may define colors using hexadecimal R/G/B, such as blue would be "color: #0000ff;"
8. Other properties
 - a. Can apply multiple css properties to an HTML element.
 - b. Alignment (default is left),
<h1 style="text-align:center;">
 - c. Font size,
<p style="text-align: center; font-size: large;">
 - d. <body style="color: red; background-color: blue;">

9. Same properties on elements:

```
<h1>Hello, world!</h1>
<h2 style="text-align: center; color: blue;"> Sec 1</h2>
<p>Some text.</p>
<h2 style="text-align: center; color: blue;"> Sec 2</h2>
<p>Some more text.</p>
```

This could be sub-optimal design, repeating the same code. The above is called **inline styling**.

10. Can include **style** information in a **style** tag inside the **head** element, factoring out the inline styling, so it can be shared:

```
<head>
  <title>Hello!</title>
  <style>
    h2
    {
      text-align: center;
      color: blue;
    }
  </style>
</head>
```

11. This can clean up HTML code, focus on structure in **<body>**, separate from style. Can make universal changes in one place, rather than a bunch of element.

12. Can define multiple style elements, and re-use code, with a **class** element:

```
<head>
  <title>Hello!</title>
  <style>
    .title
    {
      text-align: center;
      color: blue;
    }
  </style>
</head>
<h1 class="title">Hello, world!</h1>
<h2 class="title"> Sec 1</h2>
<p>Some text.</p>
<h2 class="title"> Sec 2</h2>
<p>Some more text.</p>
```

13. More properties, like **font-family**:

```
<head>
  <title>Hello!</title>
  <style>
```

```

.title
{
    text-align: center;
    color: blue;
    font-family: sans-serif;
}
</style>
</head>
<h1 class="title">Hello, world!</h1>
<h2 class="title"> Sec 1</h2>
<p>Some text.</p>
<h2 class="title"> Sec 2</h2>
<p>Some more text.</p>

```

14. Can create multiple style classes:

```

<head>
    <title>Hello!</title>
    <style>
        .title
        {
            text-align: center;
        }
        .green
        {
            color: green;
        }
    </style>
</head>
<h1 class="title">Hello, world!</h1>
<h2 class="title"> Sec 1</h2>
<p class="green"class="green"

```

15. HTML elements can have multiple classes as well, separated by a space:

```

<head>
    <title>Hello!</title>
    <style>
        .title
        {
            text-align: center;
        }
        .green
        {
            color: green;
        }

```

```

        }
    </style>
</head>
<h1 class="title green">Hello, world!</h1>
<h2 class="title"> Sec 1</h2>
<p class="green">Some text.</p>
<h2 class="title"> Sec 2</h2>
<p class="green">Some more text.</p>

```

16. Next, let's move all the class definitions from the `<style>` element nested in the `<head>` element will be separated out into `styles.css`:

```

.title
{
    text-align: center;
}
.green
{
    color: green;
}

```

17. The head element in the web-page will now look like:

```

<head>
    <title>Hello!</title>
    <link rel="stylesheet" href="styles.css">
</head>

```

- a. Makes it much less repetitive!
- b. The relation is `rel="stylesheet"`
- c. The location (assuming same folder as web-page) is `href="styles.css"`

18. If you would like a background picture here is some code (thanks to Nate B) you could adapt. This assumes that a file named `illini.jpg` is in the same folder/directory as the `.css` file. Note that the code below (won't repeat the image and will have it fill the entire page) uses the `body` class name, so it applies to the entire `body` element. If you already have a `body` class definition in your `.css` file, you can add the code to it.

```

body
{
    background-image: url('illini.jpg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: 99.5% 100%;
}

```