

Cascading Style Sheets

- **Web Pages**
 - Content (what's there: text, links, images, forms)
 - Style (how it looks)

- **What is Style?**
 - Objects in an HTML document have “style” attributes
 - Font, font size, font color assigned to headings, paragraphs
 - Background color
 - Size and shape of images
 - Hyperlinks, colors, behaviors
 - Placement of objects on the page

- **Applying style**
 - Tag-level: Style attributes within a tag on a page.
 - Page-level: Style defined within <head> of each page.
 - Site-Level: Within an external file, pulled into each page for an entire website

Cascading Style Sheets

- **Applying style**

- Tag-level: Style attributes within a tag on a page.
 - Very Granular. Difficult to maintain, No Consistency
- Page-level: Style defined within <head> of each page.
 - Difficult to maintain across many pages, No consistency

Cascading Style Sheets

- **Applying style**
 - Site-Level: Within an external file, pulled into each page for an entire website
 - Easy to maintain
 - Saves time for support & maintenance
 - Saves page load time
 - Enables consistency across the website

Cascading Style Sheets

- **What is a Style Sheet?**

- The style sheet is an EXTERNAL document (.css) containing the rules of style to be applied to your document
- You “link” to it to bring it into one or more pages

- **Why do we use them?**

- Consistency from Page to Page
- Faster, Easier Page Construction

Cascading Style Sheets

- **Guidelines for a Style Sheet**
 - Should not contain any HTML tags
 - Saved as file with a type/extension of “.css”

Cascading Style Sheets

- **Style Sheet –**
 - Reference it and bring it into your document:

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


Cascading Style Sheets

- **What does “cascading” mean?**
 - 3 ways to apply “style” to your document
 - External style sheet
 - Internal `<style>` tag in `<head>` section
 - Inline style attributes for a tag
- **Hierarchy of Applying Style**
 - Child tags Inherit Style from Parent tags
 - Detail-Level Overrides High-Level

Style Sheets - Hierarchy

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

```
<style type="text/css">
```

```
Body {color:purple;}
```

```
</style>
```

```
<p style="color:red;">
```

Red Text

```
</p>
```

- **Using the <style> tag**
 - Entered in <head> section
 - Applies to entire document
- **Style Rules are Not HTML**
- **Rule Syntax**
 1. selector – the tag that the rule applies to
 2. { } Braces to contain the rule
 3. Property:value pair
 4. Ends with semicolon

- **Let's Look at Some Examples**
- www.w3schools.com/css

- **Examples**

```
<style type="text/css">
  body  {font-size: 16pt;
         color: blue;
         background-color: pink;}

  h1    {font-size: 24pt;
         color: black; }

  p     {margin-left: 10%;
         margin-right: 10%;}

</style>
```

- **Multiple selectors are OK**

```
h1, h2          {color:purple;
                  text-align:center;}
```

- **Nested selectors are OK**

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

- **ID References are OK**

```
<h1 id="beginning">
```

```
<style> #beginning {color:red;} </style>
```

- **Classes**

- A class allows you to set different styles for the same tag
- Example:

```
<style>
```

```
    h1.majorheading {font-family: arial;  
                      color: blue;}
```

```
</style>
```

then you refer to the **class** in the tag:

```
<h1 class=majorheading>
```

- **Types of Style Rules**
 - element (tag)
 - # id
 - .class
 - Element.class
 - Group (multiple selectors)
 - Nested (selector within a selector)

- **“!important” override indicator**
 - For a rule that must NOT be overridden in cascade
 - “!important” must follow the rule
`<p style=“color: green !important;”>`
- **Best Web Tutorials:**
www.w3cschools.com/css

- **Some Style Properties**
 - Font Size
 - Font Family
 - Font-weight
 - Text-transform – capitalization
 - Word-spacing
 - Letter-spacing
 - Text-align
 - Text-indent

- **More Style Properties**
 - Color – for text
 - Background-color
 - Background-image
 - Box properties

- **Bootstrap**
 - A style library
 - Easy formatting using templates
 - Easy transition from computer-based viewing to phone-based viewing

- **Bootstrap**
 - A style library
 - Easy formatting using templates
 - Easy transition from computer-based viewing to phone-based viewing

<https://www.w3schools.com/bootstrap/default.asp>

- **Bringing Bootstrap into your pages**
 1. **Include an external CSS sheet from a CDN (Content Delivery Network)**
 2. **Include a jquery (Java Script) library (only needed for certain js plug-ins)**
 3. **Include a client java script engine from a CDN**

- **Bringing Bootstrap into your pages**

```
<!-- Latest compiled and minified CSS -->
```

```
<link rel="stylesheet"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/boo  
tstrap.min.css">
```

```
<!-- jQuery library -->
```

```
<script
```

```
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquer  
y.min.js"></script>
```

```
<!-- Latest compiled JavaScript -->
```

```
<script
```

```
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boots  
trap.min.js"></script>
```

Bootstrap Grid System

Bootstrap's grid system allows up to 12 columns across the page.

If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Bootstrap's grid system is responsive, and the columns will re-arrange automatically depending on the screen size.

Grid Classes

The Bootstrap grid system has four classes:

- `xs` (for phones - screens less than 768px wide)
- `sm` (for tablets - screens equal to or greater than 768px wide)
- `md` (for small laptops - screens equal to or greater than 992px wide)
- `lg` (for laptops and desktops - screens equal to or greater than 1200px wide)

The classes above can be combined to create more dynamic and flexible layouts.