

Basic CSS

Cascading Style Sheets (CSS)

Cascading style sheets provide the style directives to marked-up content.

They determine:

- Color**

- Size**

- Arrangement and more...**

The basic syntax is simple, and written in a text file, but CSS can grow quite complex. Through practice you can learn various style directives..

**CSS uses a different
syntax from HTML.**

Implementing Styles

There are three ways to implement CSS to your HTML:

- Inline

- Internal

- External

Inline

Specified as an attribute inside of an individual HTML element. It applies only to the element it is specified in. Sometimes this way is inefficient and can be inconsistent. However, depending on the situation it may be necessary.

Inline

Specified as an attribute inside of an individual HTML element. It applies only to the element it is specified in. Sometimes this way is inefficient and can be inconsistent. However, depending on the situation it may be necessary.

example without style:

```
<p> The house is blue. </p>
```

Inline

Specified as an attribute inside of an individual HTML element. It applies only to the element it is specified in. Sometimes this way is inefficient and can be inconsistent. However, depending on the situation it may be necessary.

example with no style applied:

```
<p> The house is blue. </p>
```

example with **Inline style** applied:

```
<p style= “color:yellow;”> The house is blue. </p>
```

Internal

Specified within the head, ie. `<head>`
... `</head>`, of the document. Can
style any element on your page. More
efficient and consistent than inline
styles, but less effective than
external. Nice for “sketching” with
code quickly on a single page.

Internal

Specified within the head, ie. `<head>` ... `</head>`, of the document. Can style any element on your page. More efficient and consistent than inline styles, but less effective than external. Nice for “sketching” with code quickly on a single page.

```
<head>
```

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

```
p {  
    color: blue;  
}
```

```
</style>
```

```
</head>
```

External

Rules are specified in an independent text file that is linked to in the head, ie. `<head> ... </head>`, of the document. This is the most efficient and consistent way to apply styles. A single CSS file can be applied to multiple HTML documents.

HTML Document

```
index.html
1 <html>
2
3 <head>
4
5 <link href="http://fonts.googleapis.
  com/css?family=Open+Sans:400,700" rel='stylesheet'
  type='text/css'>
6
7 <link href="styles.css" rel="stylesheet" type="
  text/css">
8
9 </style>
10
11
12 </head>
13
14 <title>HAIR SOUNDS</title>
15
16
17 <body>
18
19 <div id="container">
20
21 
22 
23 
24 
25
26 </div>
27
28 </body>
29
30
31 </html>
32
```

CSS

```
styles.css
1 body {
2 background: #ffffff;
3 font-family: 'Open Sans', sans-serif;
4 text-align: center;
5 font-size: .6em;
6
7 }
8
9 #container {
10 max-width: 100%;
11 height: 2000px;
12 background-color: green;
13 color: red;
14 }
15
16 #job {
17 width: 270px;
18 margin: 5px;
19 top: 0px;
20 left: 20px;
21 height: 20px;
22 }
23
24 #job2 {
25 position: absolute;
26 left: 300px;
27 width: 270px;
28 margin: 5px;
29 top: -6px;
30 height: 270px;
31 }
32
33 #job3 {
34 position: absolute;
35 left: 600px;
36 width: 270px;
37 margin: 5px;
38 top: -6px;
39 height: 270px;
```

CSS Rules

```
p {  
    font-family: Arial, sans-serif;  
    color: #ffffff;  
    background-color: green;  
}
```

CSS Rules

```
Selector .... p {  
    font-family: Arial, sans-serif;  
    color: #ffffff;  
    background-color: green;  
}
```

CSS Rules

```
      p {  
Declaration ..... font-family: Arial, sans-serif;  
                  color: #ffffff;  
                  background-color: green;  
      }
```

CSS Rules

Declarations always need a **colon** between the **property** and **value**. All declarations end with a **semi-colon**.

```
p {  
Declaration ..... font-family: Arial, sans-serif;  
                    color: #ffffff;  
                    background-color: green;  
}
```

Defining specific elements

Thus far these examples apply CSS rules to all elements like: `<h2>` or `<p>` that occur in your document. The most direct way to assign rules to specific elements is to add a **class** or **id** attribute to an HTML element. We will go over these advanced selectors and use of the cascade later to scope CSS rules to particular elements in class tomorrow.

Exercise #3:

Style Carl Dair's Seven *Principles of Typographic Contrast*