

# Web/Python Programming

웹/파이썬 프로그래밍

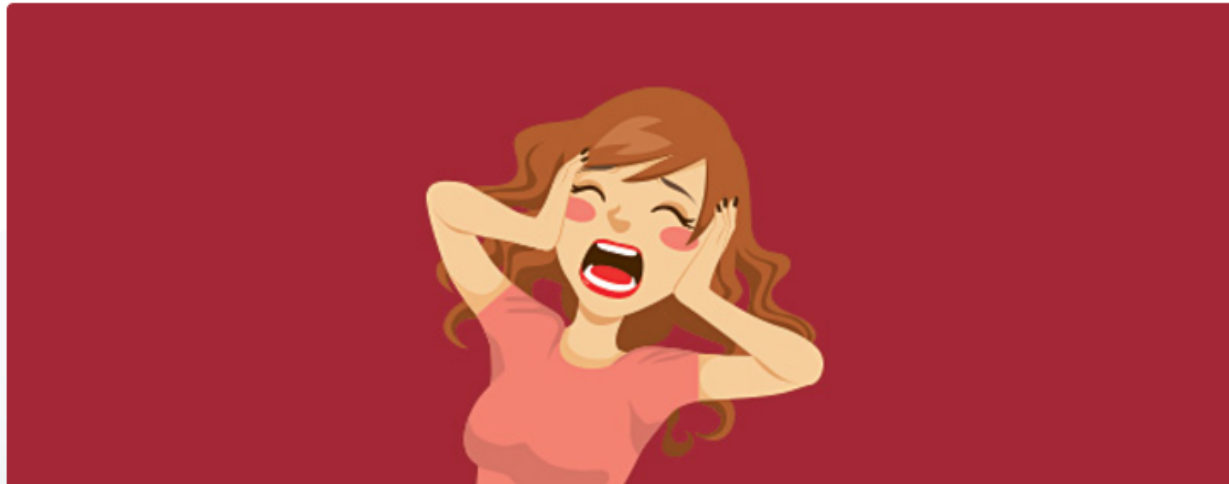
```
23 <?php language_attributes(); ?>
24 <?php bloginfo( 'charset' ); ?>
25 <?php wp_title( '|', true, 'right' ); ?>
26 <?php rel="profile" href="http://gmpg.org/xfn/11" ?>
27 <?php rel="pingback" href="http://gmpg.org/xfn/11" ?>
28 <?php fruitful_get_favicon(); ?>
29 <?php echo get_template_part( 'content', 'page' ); ?>
30 <?php wp_head(); ?>
31 </head>
32 <?php body_class(); ?>
33 <div id="page-header" class="hfeed site">
34 <?php
35 $theme_options = fruitful_get_theme_options();
36 $logo_pos = $menu_pos = '';
37 if (isset($theme_options['logo_position']))
38     $logo_pos = esc_attr($theme_options['logo_position']);
39 if (isset($theme_options['menu_position']))
40     $menu_pos = esc_attr($theme_options['menu_position']);
41 $logo_pos_class = fruitful_get_class($logo_pos);
42 $menu_pos_class = fruitful_get_class($menu_pos);
43 $responsive_menu_type = fruitful_get_type($menu_pos);
44 $responsive_menu_type = fruitful_get_type($menu_pos);
45 $responsive_menu_type = fruitful_get_type($menu_pos);
46 $responsive_menu_type = fruitful_get_type($menu_pos);
47 $responsive_menu_type = fruitful_get_type($menu_pos);
48 $responsive_menu_type = fruitful_get_type($menu_pos);
49 $responsive_menu_type = fruitful_get_type($menu_pos);
50 $responsive_menu_type = fruitful_get_type($menu_pos);
51 $responsive_menu_type = fruitful_get_type($menu_pos);
52 $responsive_menu_type = fruitful_get_type($menu_pos);
53 $responsive_menu_type = fruitful_get_type($menu_pos);
54 $responsive_menu_type = fruitful_get_type($menu_pos);
55 $responsive_menu_type = fruitful_get_type($menu_pos);
56 $responsive_menu_type = fruitful_get_type($menu_pos);
57 $responsive_menu_type = fruitful_get_type($menu_pos);
58 $responsive_menu_type = fruitful_get_type($menu_pos);
59 $responsive_menu_type = fruitful_get_type($menu_pos);
60 $responsive_menu_type = fruitful_get_type($menu_pos);
61 $responsive_menu_type = fruitful_get_type($menu_pos);
62 $responsive_menu_type = fruitful_get_type($menu_pos);
63 $responsive_menu_type = fruitful_get_type($menu_pos);
64 $responsive_menu_type = fruitful_get_type($menu_pos);
65 $responsive_menu_type = fruitful_get_type($menu_pos);
66 $responsive_menu_type = fruitful_get_type($menu_pos);
67 $responsive_menu_type = fruitful_get_type($menu_pos);
68 $responsive_menu_type = fruitful_get_type($menu_pos);
69 $responsive_menu_type = fruitful_get_type($menu_pos);
70 $responsive_menu_type = fruitful_get_type($menu_pos);
71 $responsive_menu_type = fruitful_get_type($menu_pos);
72 $responsive_menu_type = fruitful_get_type($menu_pos);
73 $responsive_menu_type = fruitful_get_type($menu_pos);
74 $responsive_menu_type = fruitful_get_type($menu_pos);
75 $responsive_menu_type = fruitful_get_type($menu_pos);
76 $responsive_menu_type = fruitful_get_type($menu_pos);
77 $responsive_menu_type = fruitful_get_type($menu_pos);
78 $responsive_menu_type = fruitful_get_type($menu_pos);
79 $responsive_menu_type = fruitful_get_type($menu_pos);
80 $responsive_menu_type = fruitful_get_type($menu_pos);
81 $responsive_menu_type = fruitful_get_type($menu_pos);
82 $responsive_menu_type = fruitful_get_type($menu_pos);
83 $responsive_menu_type = fruitful_get_type($menu_pos);
84 $responsive_menu_type = fruitful_get_type($menu_pos);
85 $responsive_menu_type = fruitful_get_type($menu_pos);
86 $responsive_menu_type = fruitful_get_type($menu_pos);
87 $responsive_menu_type = fruitful_get_type($menu_pos);
88 $responsive_menu_type = fruitful_get_type($menu_pos);
89 $responsive_menu_type = fruitful_get_type($menu_pos);
90 $responsive_menu_type = fruitful_get_type($menu_pos);
91 $responsive_menu_type = fruitful_get_type($menu_pos);
92 $responsive_menu_type = fruitful_get_type($menu_pos);
93 $responsive_menu_type = fruitful_get_type($menu_pos);
94 $responsive_menu_type = fruitful_get_type($menu_pos);
95 $responsive_menu_type = fruitful_get_type($menu_pos);
96 $responsive_menu_type = fruitful_get_type($menu_pos);
97 $responsive_menu_type = fruitful_get_type($menu_pos);
98 $responsive_menu_type = fruitful_get_type($menu_pos);
99 $responsive_menu_type = fruitful_get_type($menu_pos);
100 $responsive_menu_type = fruitful_get_type($menu_pos);
```

# Today

- CSS
  - Inline
  - Internal
  - external
- Website style and design
  - Artistic and graphic design principles
- Find places to practice CSS
  - Colors and layouts

# Bad Web Design: A Look At The Most Hilariously Terrible Websites From Around The Web

Posted on April 25, 2015 by Brenda Barron in Resources | 67 comments



# Web design principles

## Design Principles

I'm looking at these sites with several design principles in mind. They include:

- Easy to understand navigation
- Proper use of color
- Proper use of animation
- An easy to use layout
- Pleasing to the eye
- Appropriate to the topic
- The design elements don't get in the way of the content
- Great content that's easy to find, navigate, consume, and share



# Bad web designs



- <https://www.elegantthemes.com/blog/resources/bad-web-design-a-look-at-the-most-hilariously-terrible-websites-from-around-the-web>



# CSS

- Cascading Style Sheets
- Inline
  - By using the style attribute in HTML elements
- Internal
  - By using a <style> element in <head> section
- External
  - By using an external CSS file

# Recall testing example (inline CSS)

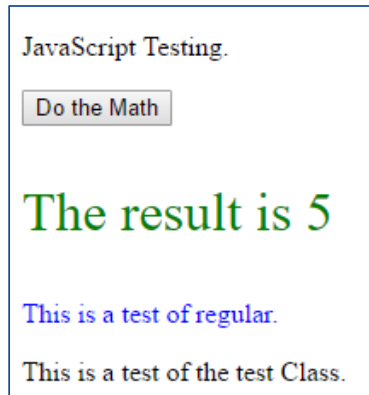
JavaScript Testing.

Do the Math

The result is 5

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Onblur Event Page</title>
5
6      <script>
7
8          function doMath()
9          {
10              var x = 2;
11              var y = 10;
12              var z = 4;
13              var answer = x * y / z;
14
15              document.getElementById("output").innerHTML = "The result is " + answer;
16          }
17      </script>
18
19  </head>
20
21  <body>
22      <p>JavaScript Testing.</p>
23
24      <button onclick="doMath();">Do the Math</button>
25      <p id="output" style="color:green;font-size:200%;"></p>
26
27  </body>
28
29  </html>
30
```

# Internal CSS



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Onblur Event Page</title>
5
6   <script>
7
8     function doMath()
9     {
10       var x = 2;
11       var y = 10;
12       var z = 4;
13       var answer = x * y / z;
14
15       document.getElementById("output").innerHTML = "The result is " + answer;
16     }
17   </script>
18
19   <style type="text/css">
20     p.regular {color:blue;font-size:100%;}
21     p.medium {color:green;font-size:150%;}
22   </style>
23
24 </head>
25
26 <body>
27   <p>JavaScript Testing.</p>
28
29   <button onclick="doMath();">Do the Math</button>
30   <p id="output" style="color:green;font-size:200%;"></p>
31
32   <p class="regular">This is a test of regular.</p>
33   <p class="testClass">This is a test of the test Class.</p>
34
35 </body>
36
37 </html>
```



# External CSS

## ■ basic.css

```
1  body {
2      background-color: lightblue;
3  }
4  h1 {
5      color: navy;
6      margin-left: 20px;
7  }
8  .testClass{
9      color: red;
10     font-size: 300%;
11 }
```

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Onblur Event Page</title>
5
6      <link rel="stylesheet" type="text/css" href="basic.css">
7
8      <script>
9
10         function doMath()
11         {
12             var x = 2;
13             var y = 10;
14             var z = 4;
15             var answer = x * y / z;
16
17             document.getElementById("output").innerHTML = "The re:
18         }
19     </script>
20
```

# External CSS

JavaScript Testing.

Do the Math

This is a test of regular.

This is a test of the test Class.

JavaScript Testing.

Do the Math

The result is 5

This is a test of regular.

This is a test of the test Class.

## CSS Tutorial

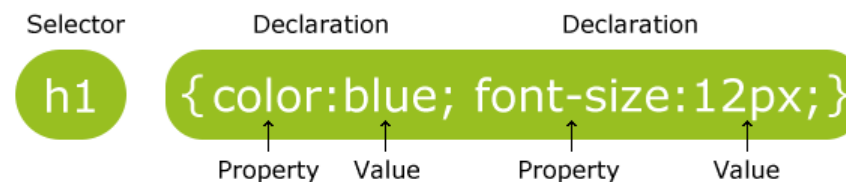
[CSS HOME](#)[CSS Introduction](#)[CSS Syntax](#)[CSS How To](#)[CSS Colors](#)[CSS Backgrounds](#)[CSS Borders](#)[CSS Margins](#)[CSS Padding](#)[CSS Height/Width](#)[CSS Box Model](#)[CSS Outline](#)[CSS Text](#)[CSS Fonts](#)[CSS Icons](#)[CSS Links](#)[CSS Lists](#)[CSS Tables](#)[CSS Display](#)[CSS Max-width](#)[CSS Position](#)[CSS Overflow](#)[CSS Float](#)

## CSS Syntax and Selectors

[< Previous](#)[Next >](#)

### CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

# CSS Selector

- Element selector
- ID selector
- Class selector

In the following example all `<p>` elements will be center-aligned, with a red text color:

## Example

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

## Example

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element should be unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The style rule below will be applied to the HTML element with id="para1":

## Example

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

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

In the example below, all HTML elements with class="center" will be red and center-aligned:

# CSS Comments

## CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment starts with `/*` and ends with `*/`. Comments can also span multiple lines:

### Example

```
p {  
  color: red;  
  /* This is a single-line comment */  
  text-align: center;  
}  
  
/* This is  
a multi-line  
comment */
```

[Try it Yourself »](#)

# Comments

## ■ Python

# Python comment for one line  
""" Python comment for  
multiple lines  
"""

## ■ HTML

<!-- HTML comment tag -->

## ■ CSS

/\* Single-line comment \*/  
/\* Multiple lines  
work as well \*/

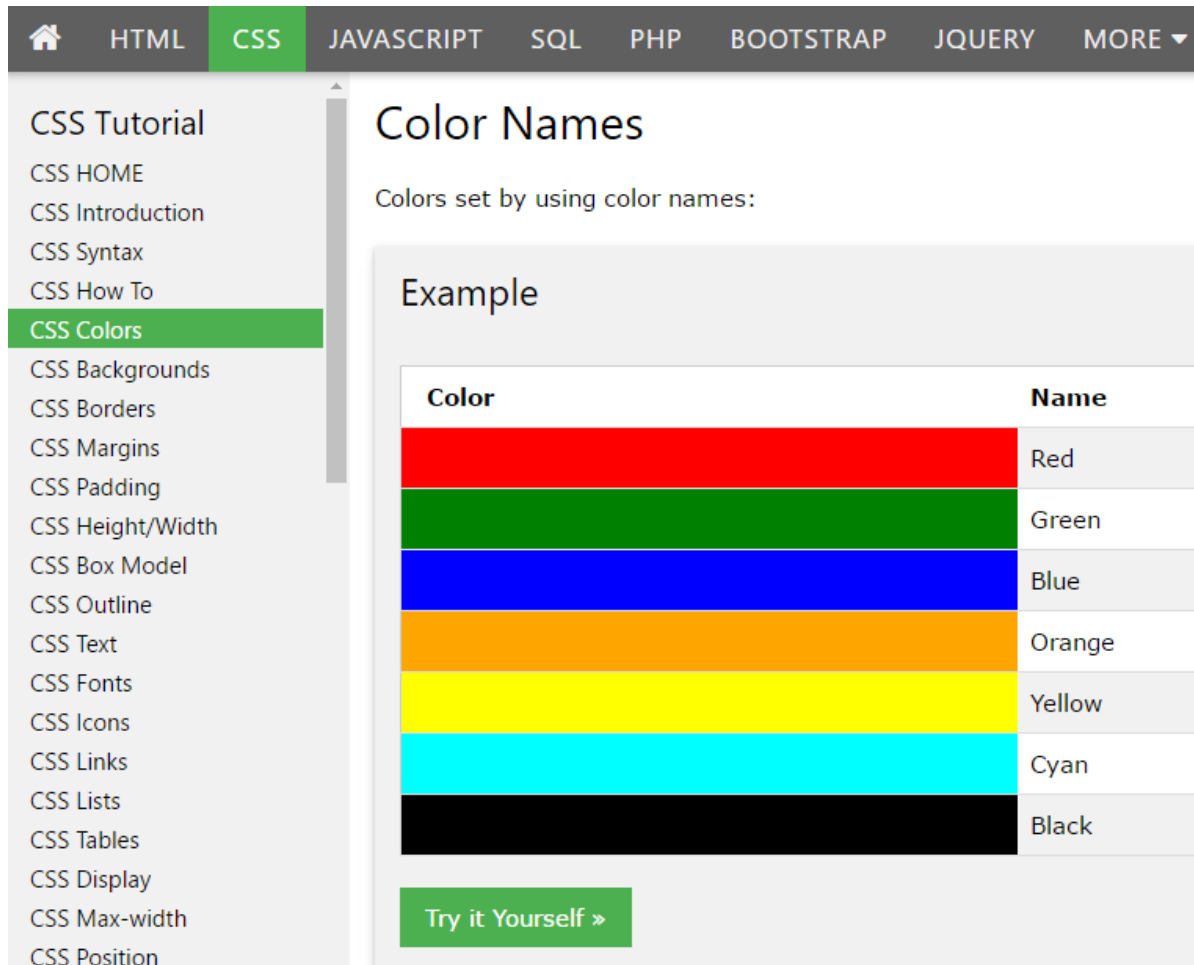
## ■ Javascript

// JavaScript comment for one line  
/\* This also works \*/  
/\* Comment can span  
As many lines as you like \*/

<!-- This is treated as a one-line JS comment  
<!-- --> JS ignores the closing characters



# CSS Colors



The screenshot shows a web page titled "Color Names" from a CSS tutorial. The page has a navigation bar with links to HTML, CSS (highlighted), JAVASCRIPT, SQL, PHP, BOOTSTRAP, JQUERY, and MORE. A sidebar on the left lists various CSS topics, with "CSS Colors" highlighted. The main content area shows a table of color names with a header "Color" and "Name". The table lists seven colors: Red, Green, Blue, Orange, Yellow, Cyan, and Black. Below the table is a green button that says "Try it Yourself »".

Color	Name
Red	Red
Green	Green
Blue	Blue
Orange	Orange
Yellow	Yellow
Cyan	Cyan
Black	Black

- Google: rgb color table
- [http://www.rapidtables.com/web/color/RGB\\_Color.htm](http://www.rapidtables.com/web/color/RGB_Color.htm)
- Naver: ppt 색조합
- <http://seiru523.blog.me/220954665056>

```
<!DOCTYPE html>
<html>
<body>

<h2>Color Names Examples</h2>
<p>Note: You will learn more about the background-color and the color
property later in our tutorial.</p>

<h2 style="background-color:red">
Red background-color
</h2>

<h2 style="background-color:green">
Green background-color
</h2>

<h2 style="background-color:blue;color:white">
Blue background-color and white text color
</h2>
|
<h2 style="background-color:orange">
Orange background-color
</h2>

<h2 style="background-color:yellow">
Yellow background-color
</h2>

<h2 style="background-color:cyan">
Cyan background-color
</h2>

<h2 style="background-color:black;color:white">
Black background-color and white text color
</h2>

</body>
</html>
```

## Color Names Examples

Note: You will learn more about the background-color and the color property later in our tutorial.

**Red background-color**

**Green background-color**

**Blue background-color and white text color**




**Orange background-color**

**Yellow background-color**

**Cyan background-color**

**Black background-color and white text color**

# <div>



Run »

Result Size: 616 x 733

```
<!DOCTYPE html>
<html>
<style>
div {height:50px; width:100%;}
</style>

<body>

<div style="background-color:green"></div>

<h2>Color Names Examples</h2>
<p>Note: You will learn more about the background-color and the color
property later in our tutorial.</p>

<h2 style="background-color:red">
Red background-color
</h2>
```

## Color Names Examples

Note: You will learn more about the background-color and the color property later in our tutorial.

Red background-color

Green background-color

Blue background-color and white text color



Run »

Result Size: 616 x 733

```
<!DOCTYPE html>
<html>
<body>

<h2>RGB Color Examples</h2>

<h2 style="background-color:rgb(255, 0, 0)">
Background-color set by using rgb(255, 0, 0)
</h2>

<h2 style="background-color:rgb(0, 255, 0)">
Background-color set by using rgb(0, 255, 0)
</h2>

<h2 style="background-color:rgb(0, 0, 255)">
Background-color set by using rgb(0, 0, 255)
</h2>

<h2 style="background-color:rgb(255, 165, 0)">
Background-color set by using rgb(255, 165, 0)
</h2>

<h2 style="background-color:rgb(255, 255, 0)">
Background-color set by using rgb(255, 255, 0)
</h2>

<h2 style="background-color:rgb(0, 255, 255)">
Background-color set by using rgb(0, 255, 255)
</h2>

</body>
</html>
```

## RGB Color Examples

Background-color set by using rgb(255, 0, 0)

Background-color set by using rgb(0, 255, 0)

Background-color set by using rgb(0, 0, 255)

Background-color set by using rgb(255, 165, 0)

Background-color set by using rgb(255, 255, 0)

Background-color set by using rgb(0, 255, 255)

# CSS layout

Html2.html

-> CSSlayout.html

google.png

-> khu\_logo.png

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>CSS Layout Page</title>
5
6
7  </head>
8
9  <body>
10
11     <H1 style="color:blue;">CSS Layout</H1>
12
13     <a href="http://swcon.khu.ac.kr">Department of Software Convergence</a>
14
15     
16
17     <p style="font-size:150%; color:red">Practicing HTML is fun. <br>So far
18     <strong>HTML</strong> is a language made of tags that
19     help us describe how we want our webpages to look.
20     We have tried the <em>head, title, H(heading),</em>
21     and paragraph tags so far. <br>
22     Very few people know all the tags in HTML,
23     but that is okay because we can do a lot with just
24     some basic tags. We can learn about new tags when
25     we need them.</p>
26
27 </body>
28
29 </html>
30
```

# CSS layout

CSSLayout.html basictest.css JavaScriptTestingCSS2.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>CSS Layout Page</title>
5
6
7 </head>
8
9 <body>
10
11 <H1 style="color:blue;">CSS Layout</H1>
12
13 <a href="http://swcon.khu.ac.kr">Department of Software Convergence</a>
14
15 
16
17 <p style="font-size:150%; color:red">Practicing HTML is fun. <br>So far
18 <strong>HTML</strong> is a language made of tags that
19 help us describe how we want our webpages to look.
20 We have tried the <em>head, title, H(heading),</em>
21 and paragraph tags so far. <br>
22 Very few people know all the tags in HTML,
23 but that is okay because we can do a lot with just
24 some basic tags. We can learn about new tags when
25 we need them.</p>
26
27 </body>
28
29 </html>
```

## CSS Layout



[Department of Software Convergence](#)

Practicing HTML is fun.  
So far we have learned  
**HTML** is a language  
made of tags that help us  
describe how we want  
our webpages to look.  
We have tried the *head*,  
*title*, *H(heading)*, and  
paragraph tags so far.  
Very few people know  
all the tags in HTML,  
but that is okay because we can do a lot with just some basic tags. We can learn about new tags when we need them.



## CSS layout

```

```

```

```

```

```

```
<p style="color:blue;font-size:200%; clear:left;">
```

[HTML](#)[CSS](#)[JAVASCRIPT](#)[SQL](#)[PHP](#)[BOOTSTRAP](#)[JQUERY](#)[MORE ▾](#)[REFERENCES ▾](#)

## CSS Reference

### CSS Reference

[CSS Selectors](#)[CSS Functions](#)[CSS Reference Aural](#)[CSS Web Safe Fonts](#)[CSS Animatable](#)[CSS Units](#)[CSS PX-EM Converter](#)[CSS Colors](#)[CSS Color Values](#)[CSS Default Values](#)[CSS3 Browser Support](#)[CSS Entities](#)

## CSS Properties

[align-content](#)[align-items](#)[align-self](#)[all](#)[animation](#)[animation-delay](#)[animation-direction](#)[animation-duration](#)

# CSS Reference

[◀ Home](#)[Next ▶](#)

W3Schools' CSS reference is tested regularly with all major browsers.

## CSS Properties

### CSS Property Groups

- [Color](#)
- [Background and Borders](#)
- [Basic Box](#)
- [Flexible Box](#)
- [Text](#)
- [Text Decoration](#)
- [Fonts](#)
- [Writing Modes](#)
- [Table](#)
- [Lists and Counters](#)
- [Animation](#)
- [Transform](#)
- [Transition](#)
- [Basic User Interface](#)
- [Multi-column](#)
- [Paged Media](#)
- [Generated Content](#)
- [Filter Effects](#)
- [Image/Replaced Content](#)
- [Masking](#)
- [Speech](#)
- [Marquee](#)

The "CSS" column indicates in which CSS version the property is defined (CSS1, CSS2, or CSS3).

## Color Properties

# CSS Properties

## CSS border Property

[< Previous](#)[Co](#)

### Example

Set the style of the four border:

```
p {  
    border: 5px solid red;  
}
```

[Try it Yourself »](#)

## CSS display Property

[< Previous](#)[Co](#)

### Example

Display <p> elements as inline elements:

```
p.inline {  
    display: inline;  
}
```

[Try it Yourself »](#)

## CSS height Property

[< Previous](#)[Co](#)

### Example

Set the height and width of a paragraph:

```
p.ex {  
    height: 100px;  
    width: 100px;  
}
```

[Try it Yourself »](#)

inline/block

# Practice CSS

- together.html
- together.css
- How to put a picture inside the table?
- How to put a link inside the table?

```
together.html x
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Together Page</title>
5
6      <link rel="stylesheet" type="text/css" href="together.css">
7
8      <script>
9
10         function doMath()
11         {
12             var x = 2;
13             var y = 10;
14             var z = 4;
15             var answer = x * y / z;
16
17             document.getElementById("output").innerHTML = "The result is " + answer;
18         }
19     </script>
20
21 </head>
22
```

```
23 </body>
24     <H1>Together Page</H1>
25
26     <button onclick="doMath();">Do the Math</button>
27     <p id="output"></p>
28
29     
30
31     <p id="text">Practicing HTML is fun. <em>So far</em> we have learned
32     <strong>HTML</strong> is a language made of tags that
33     help us describe how we want our webpages to look.
34     We have tried the <em>head, title, H(heading),</em>
35     and paragraph tags so far. <br>
36     Very few people know all the tags in HTML,
37     but that is okay because we can do a lot with just
38     some basic tags. We can learn about new tags when
39     we need them.</p>
40
41     <p class="regular">This is a test of regular.</p>
42
43     <p class="testClass">This is a test of test Class.</p>
44
45     <table>
46     <tr>
47     <td>Day 1</td><td>Walking</td><td>30 minutes</td>
48     </tr>
49     <tr>
50     <td>Day 2</td><td>Swimming</td><td>45 minutes</td>
51     </tr>
52     <tr>
53     <td>Day 3</td><td>Resting</td><td>500 minutes</td>
54     </tr>
55     <tr>
56     <td>Day 4</td><td>Resting</td><td>500 minutes</td>
57     </tr>
58     </table>
59 </body>
60
61 </html>
```