

# HTML/CSS Quick Guide

Ten tips to reference as you build out your static HTML/CSS websites.

## 1. It takes two to tango, usually.

Most elements need an opening tag need a closing tag. The wrapping tags dictate the behavior of the text you place in the middle. Some elements, however, are fine flying solo.

### Two examples of elements that DO need closing tags.

- `<p>I'm a paragraph, blah blah</p>`
- `<title>I'm the title of your page used for SEO</title>`

### Two examples of elements that DON'T need closing tags.

- ``
- `<link href="style.css" rel="stylesheet">`

Having trouble figuring out which elements need a mate? Mozilla has your back. [Check out this list of all the HTML elements and click on an element to learn more.](#)

## 2. Use syntax highlighting to find tiny mistakes.

You can stare at the same line of code over and over and still fail to see a missing quotation mark or an unclosed element. When you save your documents as either `.html` or `.css` your code editors know to color-code your markup into the appropriate language.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6 </head>
7 <body>
8   <div class="broken">Look for where the colors start to get out of whack!</div>
9 </body>
10 </html>
```

When that doesn't work, use [W3C's code validation tool](#) to help pinpoint the errors in your code.

## 3. Use parents to control children.

To organize your code in HTML you can use what's known as nesting. In the example below, let's say you want all of the text in the `<p>` tags to be centered. You can accomplish this with nesting and CSS as displayed below:

```
<!-- index.html -->
<div class="text-wrap">
  <p>I'm a child of the parent container, "text-wrap."</p>
  <p>I'm a child as well (and a sibling to the paragraph above).</p>
</div>
```

```
If you apply a style rule to my parent,  
I'll inherit that rule just like kids inherit traits from their parents.  
</p>  
</div>
```

```
/* style.css */  
.text-wrap {  
  text-align: center;  
}
```

Notice we didn't use `<center>` here. **Don't use `<center>`** ; it's a deprecated tag and won't be supported in future updates to browsers.

## 4. Divs - use them, love them, understand them.

Divs are generic block elements used to group and nest elements and text. Divs act as containers for related items in your code. These elements need an opening tag ( `<div>` ) and closing tag ( `</div>` ) and can be used for whatever you want. You don't want to get carried away with divs, however, and go too deep into nesting them because your code has the potential to become hard to read or overly complicated.

```
<div class="row">  
  <div class="column">  
    I'm some text in a div.  
  </div>  
  <div class="column">  
    Me too.  
  </div>  
</div>
```

```
.row {  
  width: 80%;  
}  
.column {  
  background: #eee;  
  border: 2px solid #fff;  
  box-shadow: 1px 1px 1px rgba(0,0,0,.5);  
  width: 40%;  
  display: inline-block;  
  padding: 10px;  
}
```

I'm some text in a div.

`.column`

Me too.

`.column`

`.row`

## 5. Spans - the inline version of divs and your friend in text formatting.

A `<span>` is often used for similar purposes as a `<div>` tag. The only difference is by default a `<span>` is an inline element - not block. That means if you wrap some text in an opening and closing `<span>` tag right in the middle of the paragraph, it won't jump to the next line.

```
<p>  
  I want to make <span class="stand-out">this word</span> really stand out in my paragraph.
```

```
</p>
```

```
.stand-out {  
  font-weight: bold;  
  color: red;  
  text-transform: uppercase;  
}
```

Spans are great for adding special formatting to portions of text or to use as inline wrappers.

## 6. Leave the styling to CSS

It's tempting sometimes to add inline styling to your HTML and call it a day, but your future self will thank you when you want to apply the same style to different portions of your site.

### Yuck!

```
<p style="color: blue; font-size: 30px;">Copy and pasting my life away.</p>  
<p style="color: blue; font-size: 30px;">Still copy and pasting my life away.</p>  
<p style="color: blue; font-size: 30px;">I'm too lazy to link up CSS.</p>
```

### Much better

```
<p>Not me, I don't copy and paste my life away.</p>  
<p>I let CSS rule my styles.</p>  
<p>And I link my style.css file in the <head> section.</p>
```

```
p {  
  color: blue;  
  font-size: 30px;  
}
```

Friendly reminder - link up the CSS in the `<head>` section like this: `<link href="style.css" rel="stylesheet">`

## 7. If you really want to make CSS mad, mess up a curly bracket.

Curly brackets (or braces) are those squiggly, mustache-looking characters you see all over CSS.

```
.class {  
  /* styles go here and will ONLY work with an opening and closing curly bracket */  
}
```

Adding an extra `}` or accidentally removing one, leaving an mateless `{` or `}` will likely break your CSS or prevent a particular style from appearing.

Unlike HTML, **you don't nest CSS** unless you're using .scss, .sass, .less, or stylus, which we are not in this class. The aforementioned extensions are for more advanced approaches to CSS and we won't get to that in this class.

## NOPE, not a thing in basic CSS.

```
/* style.css */
.my-awesome-class-name {
  p {
    color: red
  }
}
```

## YEP, this is the correct method.

```
/* style.css */
.my-awesome-class-name p {
  color: red
}
```

This targets all paragraphs within the class, "my-awesome-class-name."

```
<div class="my-awesome-class-name">
  <p>I am red text.</p>
</div>
```

## 8. Floats are weird sometimes.

The `float` property in CSS makes an element do just that, float. At times you want to float a column left and float another column right. The problem is the third column after that can float toward the top and behave counter to what you want. See this example from a [CSS Tricks](#) article on floats.



Source: CSS Tricks

To get the footer to come down to the next row and not float up to fill the gap, you would need to clear it.

```
.footer {
  clear: both;
}
```

See demo at: <http://codepen.io/ahebler/pen/pRGPQB>

We'll get more in depth on floats and clearfixes when we cover Bootstrap. If floats are really fighting you and you just want something to line up in a row, consider using `display: inline-block;`

## 10. Styles can be both inherited and overridden.

This idea is best described with an example.

## Inheriting CSS properties

```
body {  
  font-family: "Avenir", sans-serif;  
}  
p {  
  /* No font-family is set, so <p> tags will inherit the body font-family of Avenir */  
}
```

## Overriding CSS properties

```
body {  
  font-family: "Avenir", sans-serif;  
}  
p {  
  font-family: "Verdana", sans-serif;  
  /* A font-family is set, so <p> tags will update to have the font-family of Verdana */  
}
```