

# Website Styling

## Contents

HTML Basics .....	3
Why learn HTML?.....	3
HTML and CSS .....	3
Basic terminology.....	4
Make the head .....	4
Paragraphs in the body .....	5
Paragraphs and headings.....	6
Working with hyperlink.....	8
Adding images.....	8
Click that image.....	9
HTML Basics II .....	9
Ordered lists.....	10
Style Switcher.....	10
Creating the HTML part .....	11
Designing multiple alternative stylesheets.....	12
Implementing style switcher on your webpage .....	17

## HTML Basics

In this chapter, you will learn:

- Why learn HTML?
- HTML and CSS
- Basic terminology
- Make the head
- Paragraphs in the body
- Paragraphs and headings
- Working with hyperlink
- Adding images
- Click that image

### Why learn HTML?

Every webpage you look at is written in a language called HTML. You can think of HTML as the skeleton that gives every webpage structure. In this module, we'll use HTML to add paragraphs, headings, images and links to a webpage.

You can use **text editor** (e.g. Notepad, Sublime Text, Notepad++) to write HTML script and use **browser** (e.g. Chrome, Firefox, Internet Explorer) to render your script. A browser's job is to transform the code in your HTML file into a recognizable webpage. It knows how to lay out the page by following the HTML syntax.

Example:

Write the following HTML code in your HTML file.

```
<!DOCTYPE html>  
<strong>Place your text here.</strong>
```

You can change the text between `<strong>` and `</strong>` to anything you like. Open the file in your browser and you'll see how your HTML file would look in a browser. Did you see that? The `<strong></strong>` tags made our text bold.

Result:

**Place your text here.**

### HTML and CSS

HTML stands for **HyperText Markup Language**. **Hypertext** means "text with links in it." Any time you click on a word that brings you to a new webpage, you've clicked on hypertext.

A **markup language** is a programming language used to make text do more than just sit on a page: it can turn text into images, links, tables, lists, and much more. HTML is the markup language we'll be learning.

What makes webpages pretty? That's CSS—**Cascading Style Sheets**. Think of it like skin and makeup that covers the bones of HTML.

The first thing we should do is set up the skeleton of the page.

- a. Always put `<!DOCTYPE html>` on the first line. This tells the browser what language it's reading (in this case, HTML).
- b. Always put `<html>` on the next line. This starts the HTML document.
- c. Always put `</html>` on the last line. This ends the HTML document.

Example:

Write the following HTML code in your HTML file.

```
<!DOCTYPE html>
<html>
  Hi, there!
</html>
```

In between the second and last line (between the `<html>` and the `</html>`), feel free to write whatever message you like.

Result:

Hi, there!

## Basic terminology

To learn more HTML, we should learn how to talk about HTML. Already you have seen we use `<>`s a lot.

1. Things inside `<>`s are called **tags**.
2. Tags nearly always come in pairs: an opening tag and a closing tag.
3. Example of opening tag: `<html>`
4. Example of closing tag: `</html>`

You can think of tags as being like parentheses: whenever you open one, you should close it. Tags also **nest**, so you should close them in the right order: the most recently opened tag should be the first one closed, like in the example below.

```
<first-tag>
  <second-tag>Some text</second-tag>
</first-tag>
```

## Make the head

Everything in our HTML file will go between the opening `<html>` and closing `</html>` tags.

There are always two parts to an HTML file: the head and the body. Let's start with the head.

The head contains information about your HTML file, like its title. The title is what we see in the browser's title bar or page tab. For example the default title of our page is same with the file name: "test.html".



Let's add a head and a title to our webpage.

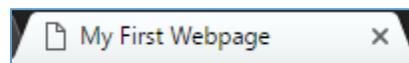
Example:

Write the following HTML code in your HTML file.

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Webpage</title>
</head>
</html>
```

In between the <title> tags, feel free to write in a title for your page. For example, I write "My First Webpage."

Result:



### Paragraphs in the body

To review, an HTML file has both a head and a body. The head is where you put information about your HTML file, like its title.

The body is where you put your content, such as text, images, and links. The content in the body is what will be visible on the actual page.

The body goes inside the <html> tags, right after the <head> tags, like this:

```
<html>
<head>
  <title>My First Webpage</title>
</head>

<body>
</body>
</html>
```

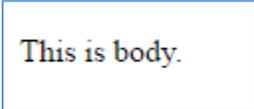
Example:

Write the following HTML code in your HTML file.

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Webpage</title>
</head>
  <p>This is body.</p>
</body>
</html>
</html>
```

Inside the body, we create a paragraph. A paragraph starts with an opening <p> tag and ends with a closing </p> tag.

Result:



This is body.

## Paragraphs and headings

We've learned when and why we use HTML. We've also learned how to:

- a. Set up an HTML file with tags
- b. Title the webpage (in the <head>)
- c. Create paragraphs (in the <body> with <p> tags)

The next step is to give our paragraphs headings using **heading tags**. Let's start with the <h1> tag. The content between this tag will be the biggest.

Example:

Write the following HTML code in your HTML file.

```
<!DOCTYPE html>
<html>
<head>
  <title>Headings & Paragraphs</title>
</head>
<body>
  <h1>The BIGGEST text ever</h1>
</body>
</html>
```

Result:

# The BIGGEST text ever

## More about headings

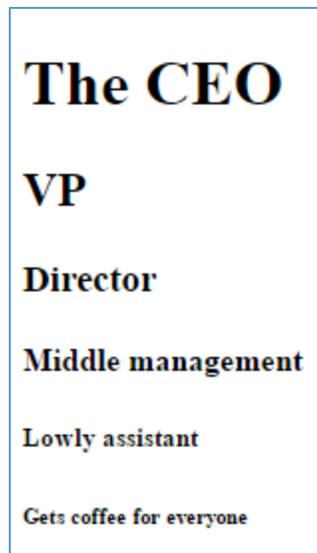
HTML actually lets us have more than one heading size. There are six heading sizes, where <h1> is the boss and <h6> is puny.

- <h1> - The CEO
- <h2> - VP
- <h3> - Director
- <h4> - Middle management
- <h5> - Lowly assistant
- <h6> - Gets coffee for everyone

Below we'll try to add headings of various sizes.

```
<!DOCTYPE html>
<html>
<head>
  <title>Headings & Paragraphs</title>
</head>
<body>
  <h1>The CEO</h1>
  <h2>VP</h2>
  <h3>Director</h3>
  <h4>Middle management</h4>
  <h5>Lowly assistant</h5>
  <h6>Gets coffee for everyone</h6>
</body>
</html>
```

Result:



## Working with hyperlink

What if you wanted to send the user to another part of your website, or another website altogether? You use hyperlinks, or links for short.

```
<a href="https://www.facebook.com">My Favorite Social Media</a>
```

1. First, there's an opening `<a>` tag and that tag has an attribute called `href`. The `href` value tells your link where you want it to go, in this case <https://www.facebook.com>.
2. Then you have a description of your link between your opening `<a>` and your closing `</a>` tags. This is what you will be able to click on.
3. Finally, you have your closing `</a>` tag.

## Adding images

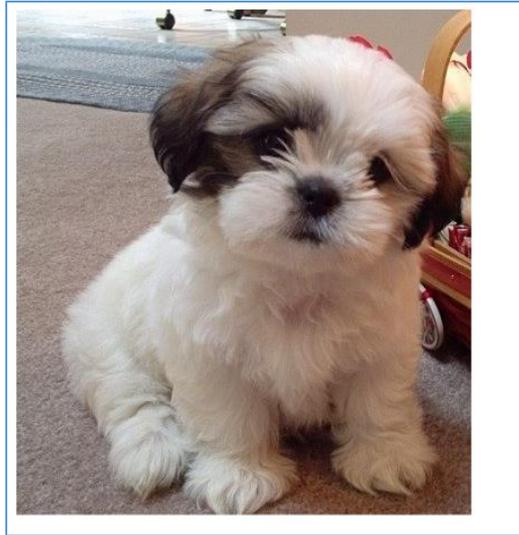
You can add images to your websites to make them look fancy.

We use an image tag, like so: `<img>`. This tag is a bit different from the others. Instead of putting the content between the tags, you tell the tag where to get the picture using `src`. It's also different because there is no ending tag. It has `/` in the tag to close it: ``.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  
</body>
</html>
```

Result:



See the web address (or **URL**) after `src=`? It's "`http://smartfamilypets.com/wp-content/uploads/2016/01/shih-tzu-puppies-for-sale.jpg`". That tells the `<img>` tag where to get the picture from.

Every image on the web has its own image URL. Simply right-click on an image and choose "Copy image URL." Paste that URL in quotes after `src=` to insert with your `<img>` tag.

[Click that image](#)

Now you know how to add links and images to your website. Why not make that image a link? For example:

```
<a href="http://smartfamilypets.com/">
  
</a>
```

1. First we open our `<a>` tag and point the `href` to <http://smartfamilypets.com/> again.
2. But this time, instead of using text inside the `<a>` tag, we use an `<img>` tag.
3. Finally, we have our closing `</a>` tag.

Now when you click on the cute puppy, you will be taken to <http://smartfamilypets.com>.

Placing one HTML tag inside of another is called **nesting**.

## HTML Basics II

In this chapter, you will learn:

- Making ordered and unordered lists
- Changing font size, color and type
- Changing the background color

- Aligning the text

## Ordered lists

Now let's learn how to make **ordered lists**. An ordered list is simply a list that is numbered, like the one below.

```
<!DOCTYPE html>
<html>
<head>
  <title>Lists</title>
</head>
<body>
  <h1>List of my favorite movies</h1>
  <ol>
    <li>The Internship</li>
    <li>Social Network</li>
    <li>Jobs</li>
    <li>Fast 8</li>
  </ol>
</body>
</html>
```

1. On line 8, we begin the ordered list with the opening tag `<ol>`.
2. On lines 9 – 11, we **wrap** (*i.e.* surround) each individual item with `<li>` and `</li>` tags.
3. Because each listed item is only on one line, we put the entire **element** on one line.
4. On line 13, we finish the ordered list with the closing tag `</ol>`.

Result:

## List of my favorite movies

1. The Internship
2. Social Network
3. Jobs
4. Fast 8

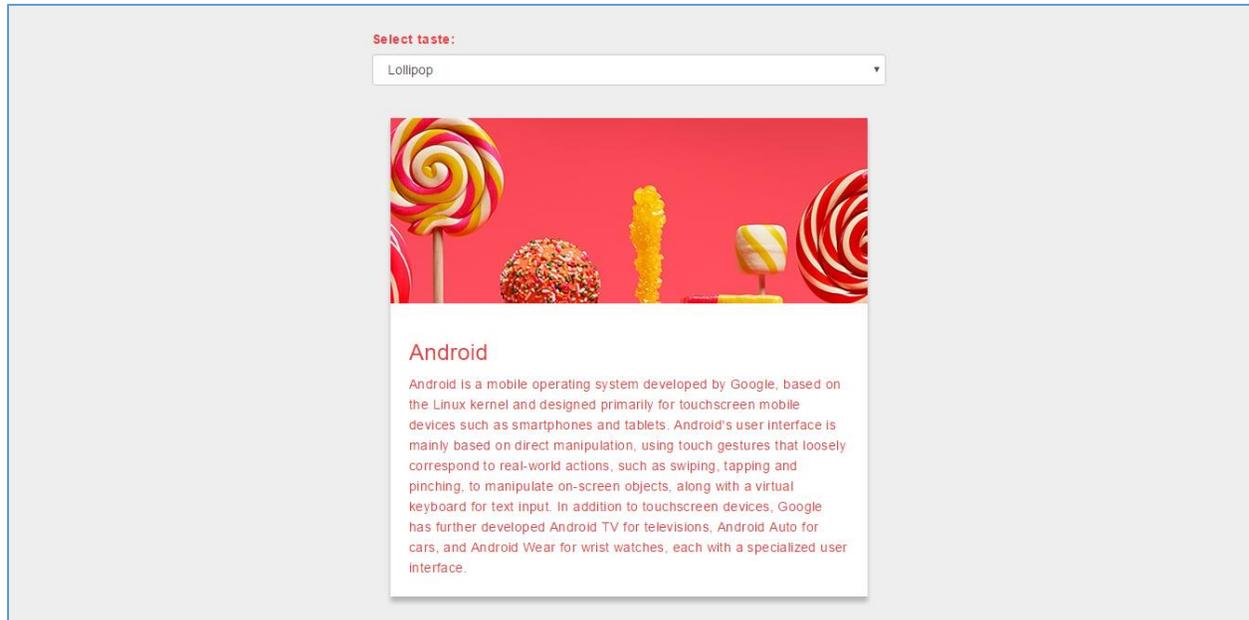
## Style Switcher

In this last chapter, you will learn:

- Creating the HTML part
- Designing multiple alternative stylesheets
- Implementing style switcher on your webpage

## Creating the HTML part

The first thing we'll need is a basic HTML page with a bit of content and styling. We'll also add a select element which will enable the visitor to change between styles. An example of what this page might look like is shown below.



Let's create a new HTML file in your text editor and save as "index.html". Write the following HTML code in your HTML file.

```
<!DOCTYPE html>
<html>
<head>
  <title>Alternating Style with CSS</title>
  <!-- Bootstrap Core -->
  <link rel="stylesheet" type="text/css"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.
min.css">
</head>
<body>

  <div class="container">
    <div class="row">
      <div class="col-md-6 col-sm-8 col-xs-12 col-md-
offset-3 col-sm-offset-2">
        <div class="form-group">
          <label for="styleSwitcher">Select taste:
</label>
          <select id="styleSwitcher" class="form-
control">
```

```

<option value="lollipop.css">Lollipop</option>
      <option
value="marshmallow.css">Marshmallow</option>
      <option
value="nougat.css">Nougat</option>
    </select>
  </div>
  <div class="card">
    <div class="image"></div>
    <div class="text">
      <h3>Android</h3>
      <p>Android is a mobile operating
system developed by Google, based on the Linux kernel and designed
primarily for touchscreen mobile devices such as smartphones and
tablets. Android's user interface is mainly based on direct
manipulation, using touch gestures that loosely correspond to real-
world actions, such as swiping, tapping and pinching, to manipulate
on-screen objects, along with a virtual keyboard for text input. In
addition to touchscreen devices, Google has further developed
Android TV for televisions, Android Auto for cars, and Android Wear
for wrist watches, each with a specialized user interface.</p>
    </div>
  </div>
</div>
</div>
</div>
</body>
</html>

```

## Designing multiple alternative stylesheets

Now we need to make three kinds of styles (Lollipop taste, Marshmallow taste, and Nougat taste). For the example, we'll make them different at the image and text color.

Let's create 3 new CSS files in your text editor and save them all as "lollipop.css", "marshmallow.css", and "nougat.css". Write the following CSS code in your CSS file.

```

/**
 * File: lollipop.css
 */

body {
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  color: #e53e48;
  font-size: 14px;
}

```

```
font-weight: 300;
  line-height: 22px;
  letter-spacing: .4px;
  background: #eee;
  padding-top: 30px;
}

.card {
  position: relative;
  display: inline-block;
  box-shadow: 0 1px 2px 0 rgba(0, 0, 0, .15);
  margin: 20px;
  margin-bottom: 50px;
  transition: all .2s ease-in-out;
}

.card:hover {
  box-shadow: 0 6px 6px rgba(0, 0, 0, .23);
}

.image {
  height: 200px;
  opacity: .7;
  overflow: hidden;
  background:
url('https://www.android.com/static/2016/img/versions/lollipop/lolli
pop-1600.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center;
  transition: all .2s ease-in-out;
}

.image:hover,
.card:hover .image {
  height: 200px;
  opacity: 1;
}

.text {
  background: #fff;
  padding: 20px;
  min-height: 200px;
}
```

```
.text p {
    margin-bottom: 0px;
}
```

```
/**
 * File: marshmallow.css
 */

body {
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
    color: #53b3e5;
    font-size: 14px;
    font-weight: 300;
    line-height: 22px;
    letter-spacing: .4px;
    background: #eee;
    padding-top: 30px;
}

.card {
    position: relative;
    display: inline-block;
    box-shadow: 0 1px 2px 0 rgba(0, 0, 0, .15);
    margin: 20px;
    margin-bottom: 50px;
    transition: all .2s ease-in-out;
}

.card:hover {
    box-shadow: 0 6px 6px rgba(0, 0, 0, .23);
}

.image {
    height: 200px;
    opacity: .7;
    overflow: hidden;
    background:
    url('https://www.android.com/static/2016/img/banners/marshmallow-6-0.jpg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: bottom;
    transition: all .2s ease-in-out;
}
```

```
.image:hover,
.card:hover .image {
    height: 200px;
    opacity: 1;
}

.text {
    background: #fff;
    padding: 20px;
    min-height: 200px;
}

.text p {
    margin-bottom: 0px;
}
```

```
/**
 * File: nougat.css
 */

body {
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
    color: #b7865b;
    font-size: 14px;
    font-weight: 300;
    line-height: 22px;
    letter-spacing: .4px;
    background: #eee;
    padding-top: 30px;
}

.card {
    position: relative;
    display: inline-block;
    box-shadow: 0 1px 2px 0 rgba(0, 0, 0, .15);
    margin: 20px;
    margin-bottom: 50px;
    transition: all .2s ease-in-out;
}

.card:hover {
    box-shadow: 0 6px 6px rgba(0, 0, 0, .23);
}
```

```

.image {
  height: 200px;
  opacity: .7;
  overflow: hidden;
  background:
url('http://static.trustedreviews.com/94/000039da2/5bfd_orh370w630/n
ougat-2.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center;
  transition: all .2s ease-in-out;
}

.image:hover,
.card:hover .image {
  height: 200px;
  opacity: 1;
}

.text {
  background: #fff;
  padding: 20px;
  min-height: 200px;
}

.text p {
  margin-bottom: 0px;
}

```

Don't forget to include your CSS file in the HTML file. We just need to include CSS file of the default theme. In this case, the default theme is lollipop. Put this code underneath the Bootstrap core link.

```

<!-- Custom CSS -->
<link rel="stylesheet" type="text/css" href="lollipop.css"
id="styletaste">

```

## Implementing style switcher on your webpage

Last, we need to add a bit of Javascript code to implementing the style switching event. Put this code underneath our Custom CSS link.

```
<script type="text/javascript">
  var switchTaste = function() {
    document.querySelector('#styletaste').href =
document.querySelector('#styleSwitcher').value;
  };
</script>
```

And to make that script work, add the onchange event on the switcher element.

```
<select id="styleSwitcher" class="form-control"
onchange="switchTaste();">
```

Open the file in your browser and you'll see how your page would look in a browser. Now you can try to change the style to the taste you prefer using the style switcher.

Result:

