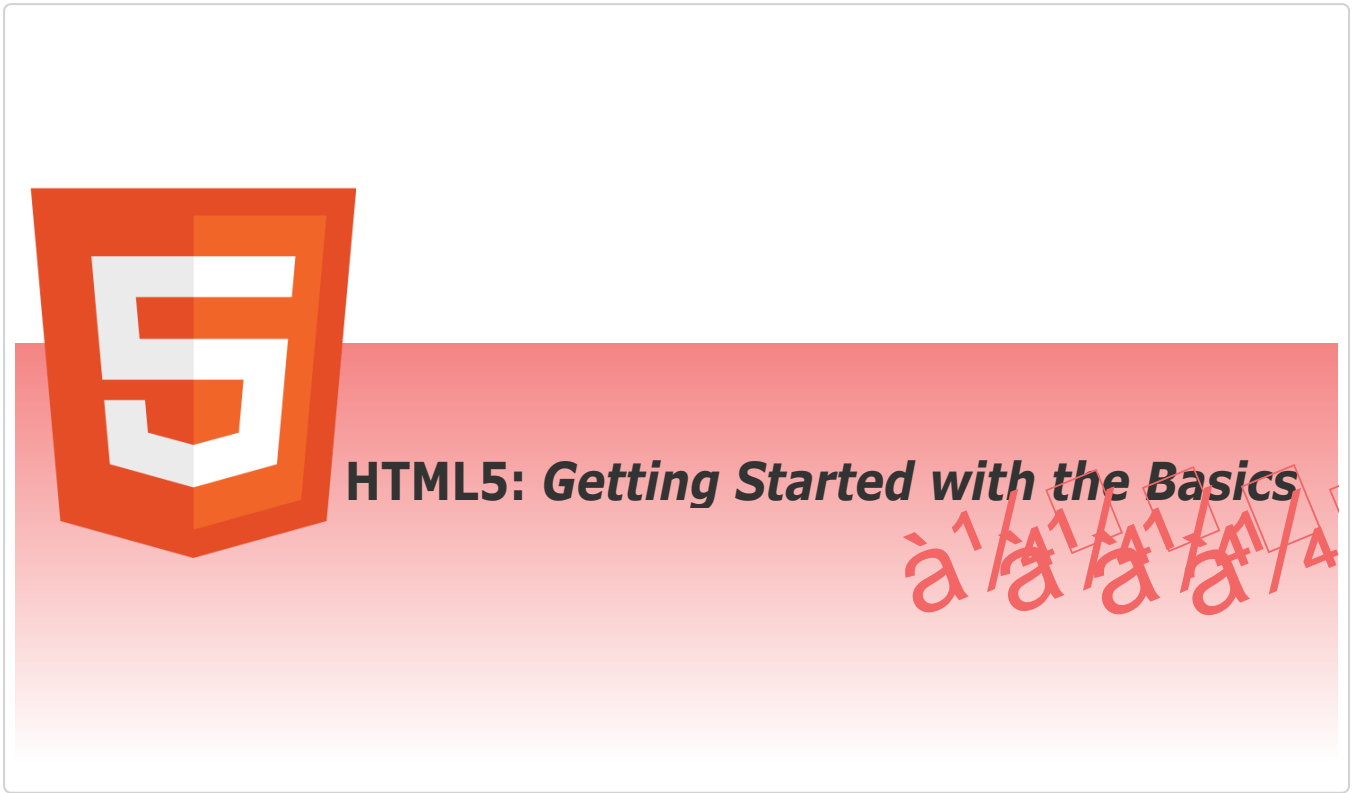
Hands-On: Diving into HTML Basics

In this lesson, we will get acquainted with some really useful HTML basics.
Let's begin!

WE'LL COVER THE FOLLOWING



In today's modern web sites, the fundamental markup technology, **HTML**, is

rarely used alone to create web pages. It is practically always used with its companions, **CSS** and **JavaScript**.

In this chapter you will learn how these technologies work together to provide the essential user experience:

As a developer, you have probably met many HTML pages and viewed their source codes. You probably know most of the HTML markup elements and can even create simple pages.

If you already know the basics of HTML, CSS, and JavaScript, this chapter will provide a recap for you. Should you be a novice in this field, you will obtain the fundamentals to master your knowledge on these topics here.

Creating a basic HTML page

An HTML page is a simple file with an `.htm` or `.html` extension and can easily be displayed in a web browser.

You do not need any development tools to create it, the most unadorned text editor, such as **vim** or **nano** in Linux on Mac or **Notepad** in Windows, is enough.

For example, you can type the following text into a file:

```
<html>
<head>
  <title>HTML page written in Notepad</title>
</head>
<body>
  <p>I wouldn't have thought it to be this easy!</p>
</body>
</html>
```

After you save it to the `easy.html` file, or in any other file with `.htm` or `.html` extension, you can open it with your default browser or any other browser installed on your computer.

Most web page development could be done with simple text editors, but as you turn from web pages to web sites with multiple pages and other related files, you recognize that you need a code editor that makes you more productive.

productive.

In our case, at **Educative**, we'll be providing you **embedded coding environments** to make your learning a breeze, no setup will be required begin! See an example of this below by toggling around the HTML and Output tabs:

> **P.S**: You can change the code in the HTML tab and the output will change automatically.

Output

HTML

I wouldn't have thought it to be this easy!

Static HTML Playground

Code along exercise 1-1: Making our first HTML page together!

Step 1:

Type the following code into the **HTML tab** given below to display some content:

```
<!DOCTYPE html>
<html>
<head>
  <title>Table of Contents</title>
```

```
</head>
<body>

  <h1>Introduction</h1>
  <p>We'll create more content later...</p>
</body>
</html>
```

Output

HTML

Static HTML Playground

Click on the **Output tab** above to display the results. The result should look like the following:

Commence live coding!

Step 2:

Here's where the actual **Live Coding** begins!

LIVE coding widget

Use our live coding widget below to write your code. As always, we encourage you to type out the code to get better practice.

Step 3:

To get the code up and running, simply click the `Run` button below and type the following command in the widget **terminal**:

```
npm start
```

This will start the `live-server` which will then begin to listen for any changes made in our code files.

See the output in action both in the **output tab** and the **host link** provided below.

If you make any changes to your code, the `live-server` will detect these; just press `Run`, there is no need to restart the server!

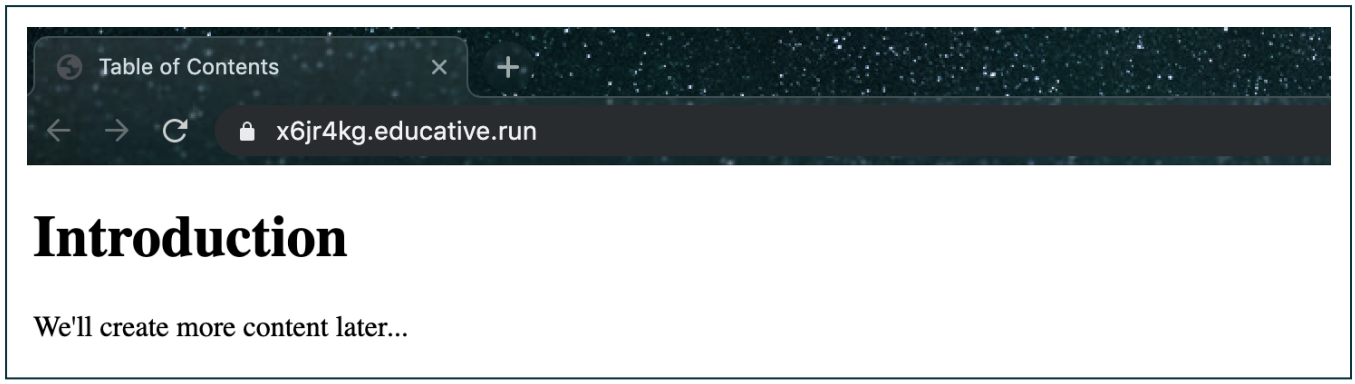
> 📜**Note:** The terminal remains alive for 30 minutes for each lesson session, after which the server is killed; so, you need not type npm start for one session unless the server is killed.

```
<!DOCTYPE html>
<html>
<head>
  <title>Table of Contents</title>
</head>
<body>
  <h1>Introduction</h1>
  <p>We'll create more content later...</p>
</body>
</html>
```

Step 4:

Click on the `link` next to the `Your app can be found at:` message to see the output in the browser.

This should open the HTML page in your browser, as in the image below:



HTML Webpage Output by Our Code

Step 5:

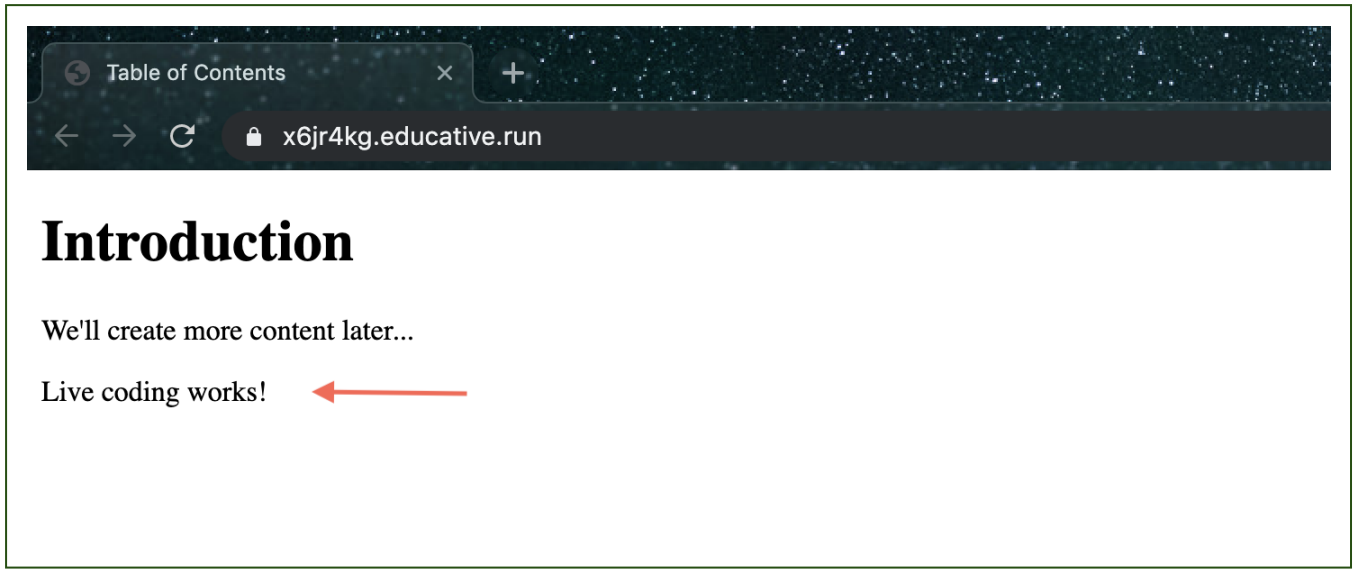
Keep your browser open and go back to the live coding widget above. In the **index.html** file, add the following line right before the closing tag:

```
<p>Live coding works!</p>
```

Click the `Run` button again and turn back to the browser.

Thanks to `live-server`, the modified markup is immediately displayed in the browser.

The updated output should look like the following:



Updated HTML WebPage

Great! Now that we've created our very first HTML page, in the *next lesson* we'll see how this all actually works.

Stay tuned! :)