

## Basic HTML5 Template For Any Project

### Table Of Contents

Last updated : April, 2nd 2021

As you learn HTML5 and add new technologies to your toolbox, you may want to build a boilerplate for yourself to start all HTML5-based projects. We encourage this, and you can also consider using one of the online resources, which gives you an essential starting point for HTML5.

In this article, we will see how to get started. Let's start with a simple, basic HTML5 page:

```
<!doctype html>
<html>
<head>
  <meta charset = "utf-8">

  <title> The HTML5 Herald </ title>
  <meta name = "description" content = "The HTML5 Herald">
  <meta name = "author" content = "SitePoint">

  <link rel = "stylesheet" href = "css/styles.css?v = 1.0">

</head>

<body>
  <script src = "js/scripts.js"> </script>
</body>

</html>
```

With the basic template in place, let's examine some of the markup's important parts and how they differ from the way HTML was written before HTML5.

### DOCTYPE

First, we have a “document type declaration” or doctype. This is just one way to tell the browser (or any other parser) what type of document is being viewed. For HTML files, it represents a specific version and style of HTML.

The doctype should always be the first item at the top of any HTML file. Many years ago, the doctype declaration was ugly and hard to remember mess. For XHTML 1.0 Strict:

```
<! DOCTYPE html PUBLIC "-// W3C // DTD XHTML 1.0 Strict // EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

HTML4 conversion:

```
<! DOCTYPE HTML PUBLIC "-// W3C // DTD HTML 4.01 Transitional // EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Although the long string of text at the top of the document didn't hurt us (apart from forcing visitors to our website to download some extra bytes), HTML5 has eliminated the obscure things. What you need now is:

```
<!doctype html>
```

Simple and fair. Doctype can be written in uppercase, lowercase, or mixed case. You will notice that the “5” is missing from the statement. Although the current iteration of web markup is called “HTML5”, it is really just an evolution of the previous HTML standard-future specifications will only be an evolution of what we have today.

Since browsers are often required to support all existing content on the web, there is no need to rely on document types to tell them which features a given document should support. In other words, doctype alone will not make your page compatible with HTML5. It all depends on the browser.

You can use one of these two old document types with new HTML5 elements on the page, and the page will look the same as when using the new doctype.

---

## HTML Element

The following HTML document is an HTML element, which has not changed significantly in HTML5. In our example, we set `lang` the attribute’s value to `en`, which specifies that the document is in English. In the based `xhtmlsyntax`, `xmlns` attributes need to be included. In HTML5, this is no longer necessary, and even `lang` attributes are not necessary for the validation or correct operation of the document.

This is what we have so far, including closed `</html>` tags:

```
<!doctype html>
<html lang = "en">
</ html>
```

---

## HEAD Element

The next part of the page is the `<head>` section. head The first line defines the document’s character encoding. This is another element that has been simplified since XHTML and HTML4 and is an optional feature, but recommended. In the past, you could write:

```
<meta http-equiv = "Content-Type" content = "text / html; charset = utf-8">
```

HTML5 improves this by minimizing character encoding tags:

```
<meta charset = "utf-8">
```

In almost all cases, UTF-8 is the value you will use in the document. A complete explanation of character encoding is beyond this article’s scope, and you may not be interested in it. Nonetheless, if you want to dig deeper, you can read topics about [W3C](#) or [WHATWG](#) .

**Note:** To ensure that all browsers can read character encoding correctly, the encoding declaration must be included within the first 512 characters of the document. It should also appear before any content-based elements (such as the