



InterviewBit

Emmet Cheat Sheet



To view the live version of the page, [click here](#).

© Copyright by Interviewbit

Contents

Emmet Tutorial: Basic to Advanced

1. Adding ID and CLASS attributes
2. Adding Custom Attributes
3. Adding Text
4. Adding Child
5. Adding Sibling
6. Multiplication
7. Grouping
8. Emmet Commands



InterviewBit

Let's get Started

Introduction

Emmet is a toolkit for web developers that can significantly improve your HTML and CSS workflows.

Basically, you can shop with the largest text content editor available, or shop and reuse a block of commonly used code called a "snippet". Snippets are a great way to increase productivity, but there are no unusual pitfalls in all implementations. You need to sketch the snippet first and it cannot be magnified at run time.

What is Emmet and Why use it?

Emmet takes the concept of snippets to a whole new level. You can enter a CSS-like expression that can be dynamically parsed and send output that depends on the input in the abbreviation. Emmet's workflow is based on HTML / XML and CSS but is designed and optimized for web builders that can also be used in programming languages.

Emmet allows you to write lightning-fast code. Simple CSS-like acronyms are turned into complex code. Easily generate lorem ipsum text, use many keyboard shortcuts, and more.

Features

- You learn how to employ Emmet from a web-developer perspective. As a web developer, you know how to use Emmet using abbreviations. They are just like CSS Selectors with shortcuts for id, class, custom attributes, detail nesting, and so on. They are dynamic, parsed as you type, and provide dynamic snippets.
- Dynamic snippets are unlike default editor snippets, in that they are dynamic and parsed as you type. For instance, to create a tag, write `MyComponent>custom-element`.
- Emmet provides a unique syntax for CSS values embedded within Emmet. For instance, `bd1-s#0.5` may be phrased to `border: 1px solid rgba (0, 0, 0, 0.5)`.
- Available for maximum famous syntaxes: use the simple abbreviation to supply code for maximum famous syntaxes like HAML, Pug, JSX, SCSS, SASS, etc.

How To Use Emmet?

Emmet can be utilized by including emmet plugins in the IDEs. Adding emmet plugins to IDEs and the use of emmet is mentioned in upcoming sections.

Adding Emmet to the IDE:

- Almost all modern IDEs, including VS Code, allow emmet support. However, if the plugin is not available by default in an IDE, it can be downloaded from the emmet website.

Using Emmet in the Code:

- When the emmet plugin is added to the IDE, simply enter the HTML tag without `<>` and the editor will show suggestions for the tag. Emmet uses a CSS-like selector syntax. Write a CSS-like abbreviation, place your cursor at the end of the abbreviation, and press Tab, Ctrl+E, or any other keyboard key configured to convert the abbreviation to actual HTML code. Emmet expands to. You can also provide a value, but if you don't, a tab will be created within each empty property. You can paste the target URL and press Tab to go to the next edit point where you can paste the next desired value.

Emmet Tutorial: Basic to Advanced

1. Adding ID and CLASS attributes

- **Adding ID**

Using emmet, we can add ID selectors. Write the “#” symbol and then the name of the id to add an id.

```
#sample_1ge
```

The code editor will create a div with the id "sample_1" after you click the tab key.

```
<div id="sample_1"></div>
```

For other elements, we can also create an id. For instance, the element name must be written after the “#” symbol followed by the id name in order to construct a <p> with the id "sample_1".

```
p#sample_1
```

The code editor will create a p with the id "sample_1" after you click the tab key.

```
<p id="sample_1"></p>
```

- **Adding CLASS**

Using emmet, we can add ID selectors. Write the “.” symbol and then the name of the id to add an id.

```
.sample_1
```

The code editor will create a div with the id "sample_1" after you click the tab key.

```
<div class="sample_1"></div>
```

For other elements, we can also create an id. For instance, the element name must be written after the “.” symbol and the id name in order to construct a `<p>` with the id “sample_1”.

```
p.sample_1
```

The code editor will create a `p` with the id “sample_1” after you click the tab key.

```
<p class="sample_1"></p>
```

- **Adding both CLASS and ID**

Using emmet, we can build tags that contain both a class and an id. Such tags must be written with the element name, followed by “#” the id name, followed by “.” and the class name.

```
p#sample_Name.class_Name
```

The code editor will generate a `p` with the class “class_Name” and ID “sample_Name” after you click the tab key.

```
<p id="sample_Name" class="class_Name"></p>
```

- **Adding multiple CLASSES**

Using emmet, we are able to generate tags with various classes. This requires writing the element name followed by “.”. This is followed by the names of the classes we want to add.

```
div.class_1.class_2.class_3
```

The code editor will construct a `div` with the classes “class1,” “class2,” and “class3” when you click the tab key.

```
<div class="class_1 class_2 class_3"></div>
```

2. Adding Custom Attributes

Using emmet, we may construct a tag with a certain attribute and pass its value. To accomplish this, we must enclose the element name in square brackets “[]”. We can include the name(s) of one or more attributes inside the bracket along with the value.

```
p[title="Custom Title"]
```

The code editor will create an element with the name "Scaler academy" when you click the tab key.

```
<p title=" Custom Title"></p>
```

We can pass an attribute without a parameter and write several attributes. Without an argument, the empty string (“”) will be applied to the attribute.

```
td[rowspan=4 colspan=5 title]
```

The code editor will create a td with rowspan=4 colspan=4 title="" when you click the tab key.

```
<td rowspan="4" colspan="5" title=""></td>
```

3. Adding Text

Using emmet, we may also add sentences or paragraphs inside of tags. To do this, we must write the element name inside the curly brackets. The text item can be added within these curly brackets.

```
a{Click on the link here}
```

The code editor will generate an a with the word "Click on the link here" when you click the tab key.

```
<a href="">Click on the link here</a>
```

4. Adding Child

Tags can be nested within tags. To accomplish this, we must write the parent tag name followed by the symbol > before the tag that will be nested.

```
div>p_1
```

The code editor will produce a div with a p_1 within when the tab key is clicked.

```
<div>  
  <p_1></p_1>  
</div>
```

Additional items can be nested. We must continue adding children in order to do this.

Kindly Note: that the element to the left of > will act as the parent for the element to the right of >*.

```
div>ul>li_1
```

The code editor will produce a nav with a li_1 nested inside a ul after you click the tab key.

```
<div>  
  <ul>  
    <li_1></li_1>  
  </ul>  
</div>
```

5. Adding Sibling

We can give HTML sibling tags by using emmet. (Elements that have the same parent are considered siblings.) To accomplish this, we must insert + symbols between tags. That is, the code editor will construct siblings for two or more tags if their emmet shortcuts have a + in the middle.

```
div_1+p_1+div_2
```

The code editor will construct the three tags `div_1`, `p_1`, and `div_2` as siblings when the tab key is pressed.

```
<div_1></div_1>  
<p_1></p_1>  
<div_2></div_2>
```

6. Multiplication

We now know how to include a child inside of a tag. But what if we need to put more children inside the tag (all with the same tag)? In certain circumstances, tag multiplication is an option. After the tag that needs to be multiplied and before the number of repetitions, we need to add a `*`.

```
ul_1>li*3
```

The code editor will produce an unordered list with 3 list entries when the tab key is pressed.

```
<ul_1>  
  <li></li>  
  <li></li>  
  <li></li>  
</ul_1>
```

7. Grouping

Emmet can be used to group HTML tags. To accomplish this, a bracket must be placed around the tags that will be gathered (`()`).

```
nav>(header>ul_1>li*2>a)+footer>p_1
```

When the tab key is pressed, the code editor will produce a nav containing child tags for the header, footer, and sibling footer, all of which include the child tag p_1.

```
<nav>
  <header>
    <ul_1>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
    </ul_1>
  </header>
  <footer>
    <p_1></p_1>
  </footer>
</nav>
```

8. Emmet Commands

Alright, now that we have a fair idea about the Emmet Toolkit and its features, let us now move on to the Emmet Commands.

Here, we have all the important Emmet Commands with Examples.

COMMAND	MEANING	EXAMPLE & SYNTAX
Child: >	This code uses the “>” operator to nest elements inside each other.	<p>Syntax : <code>nav>ul_1>li{Name}</code></p> <p>Output:</p> <pre><nav> <ul_1> Name </ul_1> </nav></pre>
Sibling: +	This code uses the “+” operator to place elements near each other, on the same level:	<p>Syntax : <code>div+p_1+bq</code></p> <p>Output:</p> <pre><div></div> <p_1></p_1> <blockquote></blockquote></pre>
Climb-up: ^	With the “^” operator you’re descending down the generated tree and the positions of all sibling elements will be resolved against the deepest element:	<p>Syntax : <code>div_1+div_2>p>span+em</code></p> <p>Output:</p> <pre><div_1></div_1> <div_2> <p> <blockquote></blockquote> </div_2></pre>
Grouping: ()	Parentheses are used by Emmets’ power users for grouping subtrees in complex abbreviations:	<p>Syntax : <code>(nav>(header>Table>t</code></p> <p>Output:</p> <pre><nav> <header> <Table> <td></td> <td></td> </Table> </header></pre>

Conclusion

In this article, we learned about the Emmet toolkit. We can conclude that Emmet provides us shortcuts to help us write Long HTML codes easily. It is a toolkit that is used to write HTML codes. It has various inbuilt powerful modules that follow simple syntax which makes it appealing to both beginners and experienced folks alike. A vast collection of functions makes writing codes of any sort much easier. In this cheat sheet, we have covered the most common fundamentals of Emmet that would help you kickstart your career in Web development.

Important Resources

- [Web Development Projects](#)
- [Web Development Courses](#)
- [Web Developer Interview Questions](#)
- [Web Developer Skills](#)

Links to More Interview Questions

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)