Emmet Documentation

Back to main website

Abbreviations

CSS Abbreviations

Actions

Filters

Customization

Cheat Sheet

More developer tools:

Emmet LiveStyle

Real-time bi-directional edit tool for CSS, LESS and SCSS.

Emmet Re:view

Fast and easy way to test responsive design side-by-side.

Element types

in HTML and XML documents, when you expand abbreviations, all abbreviation parts are transformed on-the-fly into HTML/XML tags. But certain elements like `a` or `img` are transformed into elements with predefined attributes: `<a href=""></a>` and `<img src="" alt="" />`. How does Emmet know when to add those attributes?

All Emmet elements definitions are stored in snippets.json file in the following format:

```
{
    "html": {
        "abbreviations": {
            "a": "<a href=\"\">",
            "link": "<link rel=\"stylesheet\" href=\"\" />"
            ...
        },
        "snippets": {
            "cc:ie6": "<!--[if lte IE 6]>\n\t${child}|\n<![endif]-->"
            ...
        }
    },

    "css": {
        ...
    }
}
```

As you can see, at first level there are syntax names for which elements are defined. Inside the syntax section there are elements definitions split across two sections: *snippets* and *abbreviations*.

Snippets

Snippets are just blocks of plain code, just like in all programmers' editors. You can type anything there and it will be outputted "as-is", without any transformation.

Abbreviations

Abbreviations are actually building blocks with some data hints. Since Emmet is mostly used for writing HTML/XML tags, *abbreviation definition uses XML format to describe element*.

Emmet parses abbreviation definition and retrieves the following data:

- element name;
- default attributes;
- attributes' order;
- attributes' default values;
- should element contain closing tag.

Let's take a closer look on HTML abbreviations' definitions above. The `link` element is defined as `<link rel="stylesheet" href="" />` (double quotes should be escaped in JSON; or use single quotes instead). This definition says that tag, generated for `link` abbreviation, should be named *link* and should contain two attributes: *rel* with default value "stylesheet" and *href* with empty value (exactly in this order), and generated element should not contain closing tag.

When the `link` abbreviation is expanded, you'll receive the following output for HTML syntax:

```
<link rel="stylesheet" href="">
```

You can override default attribute values and add new ones as well:

```
link[rel=prefetch title="Hello world"]
```

...expands to

```
<link rel="prefetch" href="" title="Hello world">
```

You can add child elements as well, which forces Emmet to output closing tag:

```
link>xsl:apply-templates
```

...will output

```
<link rel="stylesheet" href="">
    <xsl:apply-templates></xsl:apply-templates>
</link>
```

Aliases

In the abbreviations section of `snippets.json` you can also define *aliases*: a short-hands for commonly used abbreviations. Aliases can be used to define:

- short names for long tag names;
- referencing commonly used abbreviations.

In `snippets.json` file, you can find the following definitions:

```
...
"html": {
    "abbreviations": {
        "bq": "blockquote",
        "ol+": "ol>li"
    }
}
```

In the example above, when you expand `bq` abbreviation, Emmet will look for `blockquote` abbreviation's definition. If it doesn't exist, it will simply output `<blockquote></blockquote>` element. The `ol+` abbreviation actually outputs the same result as `ol>li` does.

The `ol+` definition may look ambiguous since it contains `+` at the end which is also a sibling operator. Emmet correctly expands such abbreviations and the plus sign is left here for historical reasons. Just remember that you don't need to use plus sign to create abbreviation alias.

comments powered by Disqus

Support: info@emmet.io
Created with DocPad and Gulp.js
Minimal theme by orderedlist

View page source on GitHub