posthtml-match-helper

Expand CSS selectors into PostHTML matcher objects

Introduction

This PostHTML plugin can turn simple CSS selectors into matcher objects.

Supported features:

Tags: `"div"` returns `{tag: "div"}`.

Ids: `"#bar"` returns `{attrs: {id: "bar"}}`.

Classes: `.foo` returns `{attrs: { class: /(?:^|\s)foo(?:\s|$)/ }}`. Any number of classnames supported.

Attribute selectors: any number of standard attribute selectors can be used[1] including the following non-standard:

`[attr!=value]`: matches attributes with values that do not contain `value`.

Multiple node selectors: `"div, span"` returns `[{tag: "div"}, {tag: "span"}]`.

[1] Multiple attribute selectors for the same attribute are not supported (this includes mixing classnames and attribute selectors matching `class`).

The basic template for selectors (and order of features) looks like this:

```
"tag#id.class.name[attr*=value]
        [otherattr^='start']"
```

Basic usage

```
import matchHelper from "posthtml-match-
        helper";

tree.match(matchHelper("div.class"),
        function (node) {
  // do stuff with matched node...
});
```

Advanced usage

```
import matchHelper from "posthtml-match-
        helper";
```

```
tree.match(matchHelper("input.my-
        control[type!='radio'][checked],
        input[value^='foo'][checked]"),
        function (node) {
  // do stuff with node that matched
        either of the selectors...
});
```

Classnames with escaped characters

If you need to match nodes with classnames that use escaped characters, like those in Tailwind CSS utilities with arbitrary values, use the following syntax:

```
import matchHelper from "posthtml-match-
        helper";

tree.match(matchHelper("input.\\
        [display:none\\]"), function
        (node) {
  // do stuff with node that matched
        either of the selectors...
});
```


The helper function

Arguments

- `matcher` (string) - A CSS selector that describes the node you want to match in PostHTML.

Returns

A matcher object or an array of matcher objects.