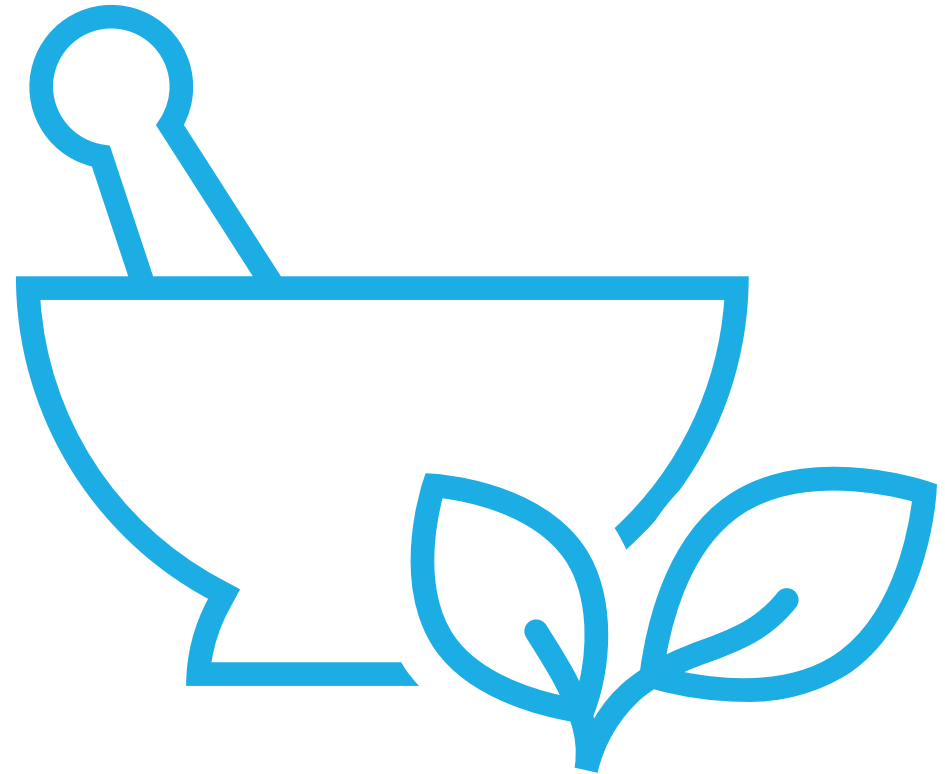


PSEUDO SELECTORS



INTRODUCTION

So now we can target more specific cases of our HTML code, but what if we want to add a hint of reactivity to our page?

What if we want certain style rules to apply only if an element is being hovered? Or while it is being clicked? These kinds of selectors are called **pseudo selectors** and we can use them to give our page a little bit of flare.

We can also use pseudo selectors to target very specific tags. If you specifically want the 6th child of a selector, you can do that!

PSEUDO SELECTORS

So how do we create pseudo selectors? In general it is very easy:

```
selector:pseudo-selector {  
}
```

So we use a normal selector, and then a pseudo selector separated with a : character.

```
p: hover {  
    font-size: 30px;  
    color: orange;  
}
```

In this case, when an p tag is **hovered** the text color will change to orange and grow to 30px. Note this only applies one tag at a time! Only the p tag being hovered will change, not all of them at once.

PSEUDO SELECTORS CONT.

Pseudo selectors are usually (but not always) used to target some kind of “state” that an element can be in. These include but are not limited to:

- The element is in the *hover* state
- The element is in the *active* state
- The element is in the *visited* state
- The element is in the *link* state
- The element is in the *focus* state

Some of these states work on all elements while others only apply to certain kinds of tags. The visited and link state are used for a tags while the focus is used for input tags.

The hover pseudo selector can be used on pretty much any element!

HOVER

If you want a style to only show when an element is being hovered, use the **hover** pseudo selector.

```
#importantHeader:hover {  
    font-size: 70px;  
    color: red;  
}
```

These rules will only happen when the element is hovered!

ACTIVE

If you want a style to only show when an element is being clicked, use the **active** pseudo selector.

```
#importantHeader:active {  
    font-size: 40px;  
    color: blue;  
}
```

These rules will only happen when the element is being clicked, meaning the mouse is being held down.

KNOWLEDGE CHECK

Let's add some basic interactivity:

1. Create a directory called `CssPseudoSelectors` inside `Scratch`
2. Follow the `Git/Github` workflow to turn this into a repository
3. Create a `style.css` file and link it in your `index.html` file
4. Add at least 7 content tags to `index.html`
 1. Make sure you nest some tags within others
5. Create a rule using the `a` **hover** pseudo selector at least 2 times
 1. Change whatever style you want
6. Create a rule using an **active** pseudo selector at least 2 times
 1. Change whatever style you want
7. Add, commit and push your code.

LINK

Your browser keeps track of what links you have clicked on while on a website. The `:link` selector will only target links that have **not** been visited.

```
a:link {  
    font-weight: bold;  
    color: purple;  
}
```

I use the `a` tag selector here, but you could be more specific and target certain classes or an id.

VISITED

If you want to style links that have been visited, use the **visited** pseudo selector.

```
a:visited {  
    color: pink;  
}
```

Again, I use the a tag selector here, but you could be more specific and target certain classes or an id.



KNOWLEDGE CHECK

More interactivity:

1. Using the CssPseudoSelectors folder for the following
2. Create a rule using the a **link** pseudo selector at least 2 times
 1. Change whatever style you want
3. Create a rule using a **visited** pseudo selector at least 2 times
 1. Change whatever style you want
4. Add, commit and push your code.

NOT

You can use the **not** pseudo selector to target anything that does *not* match the selector given.

```
:not(.blueText) {  
    color: red;  
}
```

Notice the difference here, there is no selector before the **:** character. The selector is put inside some round brackets (). So here will match anything that does **not** have the blueText class.

NTH-CHILD

This is a bit of a strange one, if you want to target a specific child element, use the **nth-child**. For example, the following will select any p tag that is the 3rd child to its parent.

```
p:nth-child(3) {  
    font-weight: bold;  
}
```

Again we have something looking a little bit different. We have a selector on the left, but we still use ().



KNOWLEDGE CHECK

More interactivity:

1. Using the CssPseudoSelectors folder for the following
2. Create a rule using the a **not** pseudo selector at least 2 times
 1. Change whatever style you want
3. Create a rule using a **nth-child** pseudo selector at least 2 times
 1. Change whatever style you want
4. Add, commit and push your code.

There are lots of different pseudo selectors for different use cases.

For the most part you will only really need things like hover and active but again, specific use cases might come up where you want to style very particular things!

Here is a reference to the current pseudo selectors:

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes#alphabetical_index