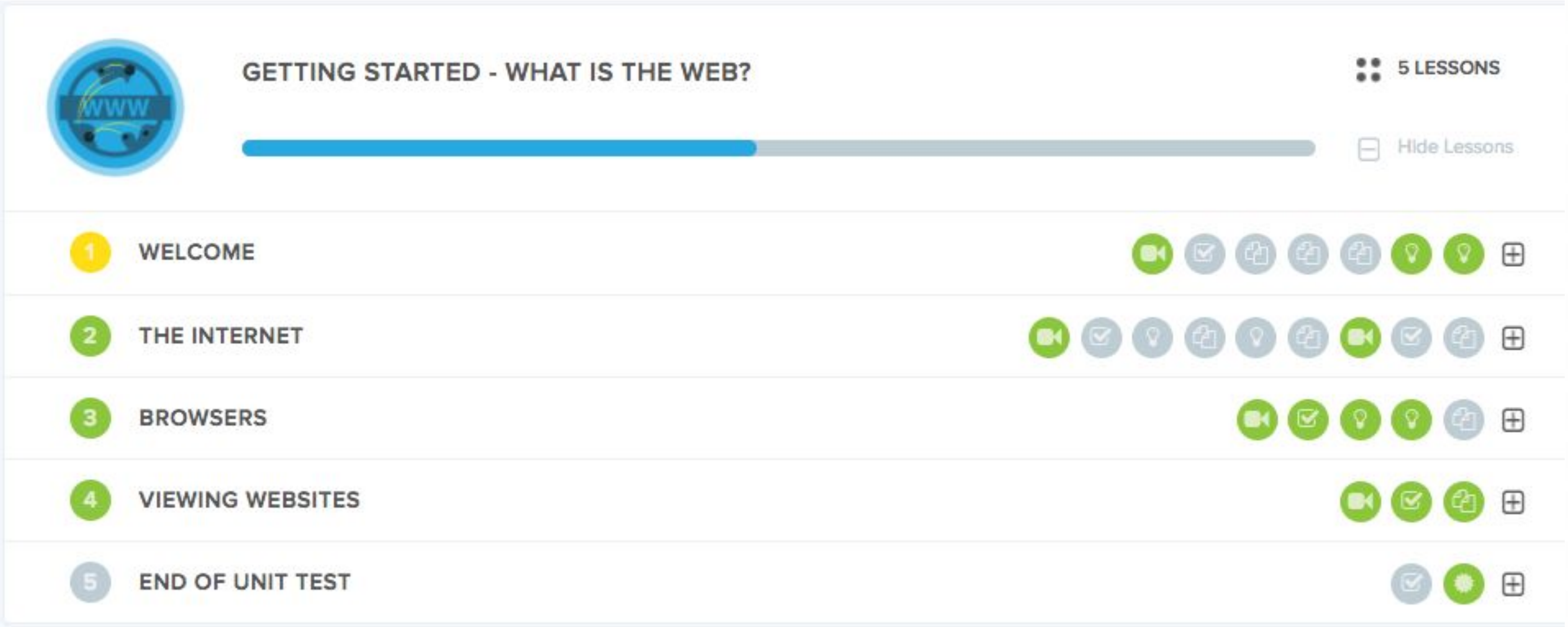
CSS Special Selectors

Styles can change when the user does things!


GETTING STARTED - WHAT IS THE WEB?

5 LESSONS

Hide Lessons

1 WELCOME

2 THE INTERNET

3 BROWSERS

4 VIEWING WEBSITES

5 END OF UNIT TEST

Introducing

CSS Pseudo-classes

CSS Pseudo-classes allow you to select elements that are in a certain state, such as when the mouse is hovering over an element.

CSS Pseudo-classes

```
p {
  color: red;
}
```

Selects all p tags and gives them the color red

CSS Pseudo-classes

```
p:hover {
   color: red;
}
```

Selects all p tags that the ***mouse is hovering over*** and gives them the color red

General Format

```
selector:pseudo-class {
  property: value;
}
```

General Format

```
selector:pseudo-class {
  property: value;
}
```

This can be any CSS selector, including tag, class, combination of tag and class, or even descendant / child selectors

General Format

```
selector:pseudo-class {
  property: value;
}
```

This can be any valid pseudo-class name. We'll learn about several. Examples include **`hover`**, **`active`**, **`visited`**, and **`link`**

General Format

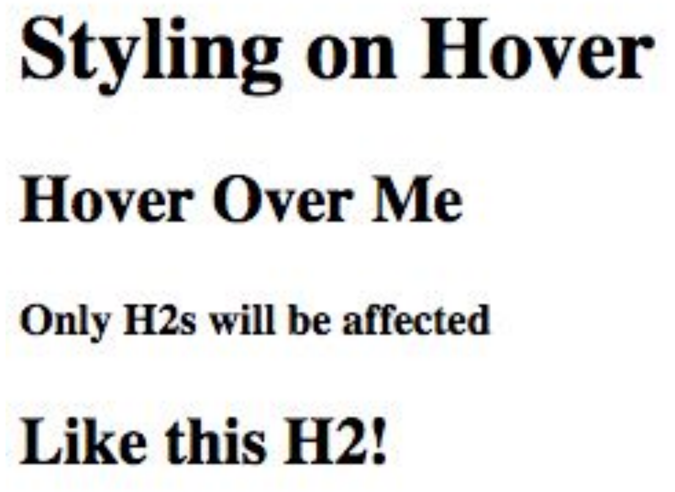
```
selector:pseudo-class {
  property: value;
}
```

Standard CSS declaration

Styling on Hover

```
h2:hover {
   color: red;
   font-style: italic;
}
```

Applies to all h2 tags when the mouse is hovering



Styling on Hover

Hover Over Me

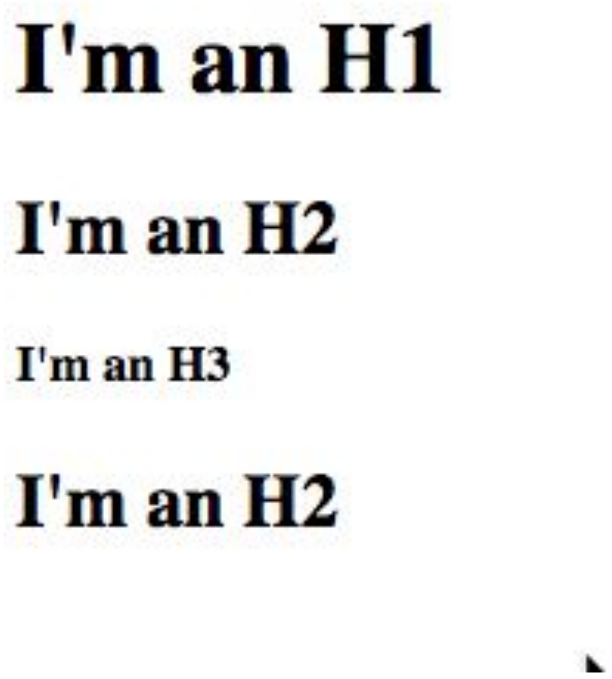
Only H2s will be affected

Like this H2!

Styling on Hover

```
h1 + h2:hover {
   color: red;
   font-style: italic;
}
```

Applies only to h2 tags that immediately follow h1 tags, when the mouse is hovering!



I'm an H1

I'm an H2

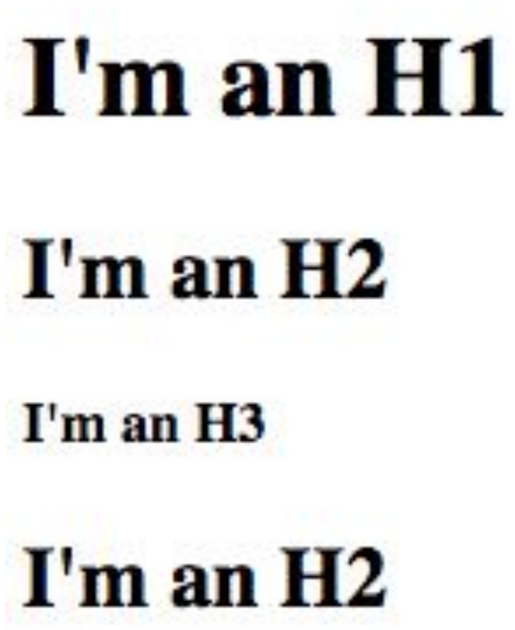
I'm an H3

I'm an H2

Styling on Hover

```
body :hover {
    color: red;
    font-style: italic;
}
```

Applies to each element inside the body, when the mouse is hovering



I'm an H1

I'm an H2

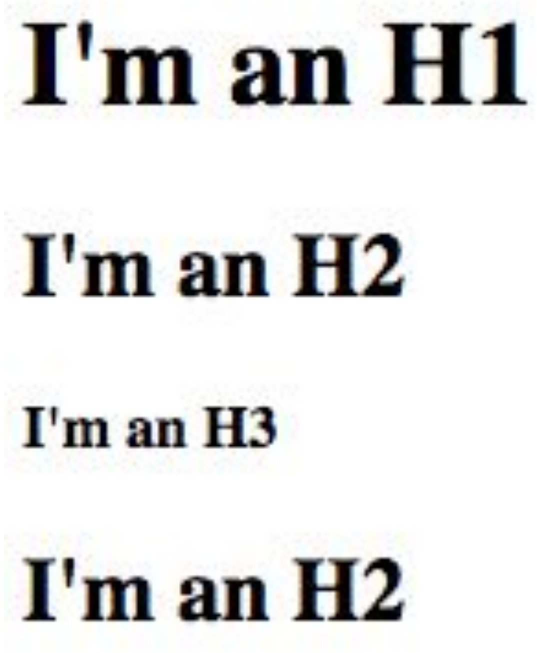
I'm an H3

I'm an H2

Styling on Hover

```
body:hover {
    color: red;
    font-style: italic;
}
```

Applies to the entire body when the mouse is hovering over the body of the page



I'm an H1

I'm an H2

I'm an H3

I'm an H2

Other Pseudo-classes

What other pseudo-classes are there?

`:link` - select unvisited links

`:visited` - select visited links

`:active` - select element currently being clicked on

Full list here: https://www.w3schools.com/css/css_pseudo_classes.asp

Styling Links

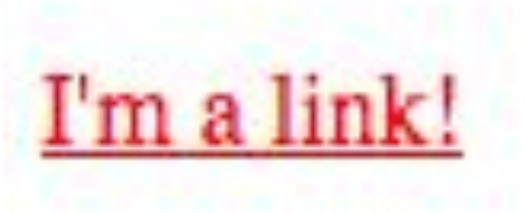
```
a:link {
    color: red;
}
a:visited {
    color: green;
}
a:hover {
    color: pink;
}
a:active {
    color: purple;
}
```

Styling Links

```
a:link {          Unvisited links
    color: red;
}
a:visited {       Visited links
    color: green;
}
a:hover {         Links being hovered over
    color: pink;
}
a:active {        Links being clicked on
    color: purple;
}
```

Styling Links

```
a:link {
    color: red;
}
a:visited {
    color: green;
}
a:hover {
    color: pink;
}
a:active {
    color: purple;
}
```



I'm a link!

Styling Links

```
a:link {
    color: red;
}
a:visited {
    color: green;
}
a:hover {
    color: pink;
}
a:active {
    color: purple;
}
```

Note:

- a:hover must come after a:link and a:visited in order to work
- a:active must come after a:hover in order to work

Styling Links

```
a:link {
    color: red;
}
a:visited {
    color: green;
}
a:hover {
    color: pink;
}
a:active {
    color: purple;
}
```

WARNING
Browsers have default styling for links that people get used to seeing (ex: blue unvisited, red active, purple visited)

Only change these defaults if you have a good reason (ie you're styling your links as buttons)

Styling during Mouse Clicks

```css
p, h1 {
   background-color: yellow;
}
```

Selects all p and h1 elements and gives them a background color of yellow

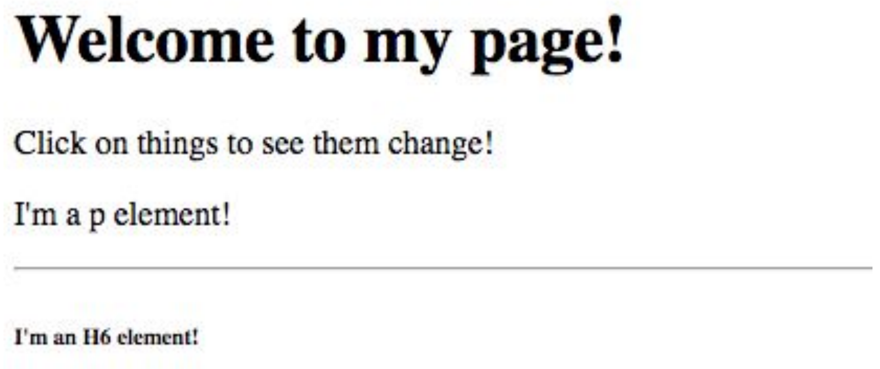
Styling during Mouse Clicks

```
p:active, h1:active {
  background-color: yellow;
}
```

Selects all p and h1 elements ***currently being clicked*** and gives them a background color of yellow

Styling during Mouse Clicks

```
p:active, h1:active {
  background-color: yellow;
}
```



Welcome to my page!

Click on things to see them change!

I'm a p element!

I'm an H6 element!

Introducing

CSS Pseudo-Elements

CSS pseudo-elements allow us to style specific parts of an element's content

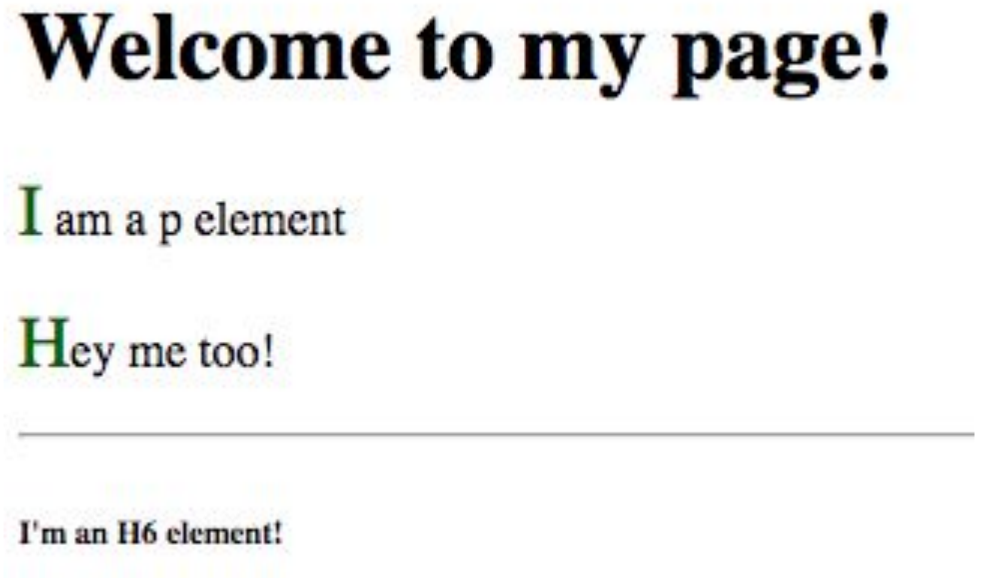
CSS Pseudo-Element Example

```
p::first-letter {
   color: DarkGreen;
   font-size: 24px;
}
```

Selects and styles only the first letter of every p element

CSS Pseudo-Element Example

```
p::first-letter {
   color: DarkGreen;
   font-size: 24px;
}
```



Welcome to my page!

I am a p element

Hey me too!

I'm an H6 element!

General Format

```
selector::pseudo-element {
  property: value;
}
```

General Format

```
selector::pseudo-element {
  property: value;
}
```

This can be any CSS selector, including tag, class, combination of tag and class, or even descendant / child selectors

General Format

```
selector::pseudo-element {
  property: value;
}
```

This can be any valid pseudo-element name.

Examples include **`first-letter`**, and **`first-line`**

General Format

```
selector::pseudo-element {
  property: value;
}
```

Standard CSS declaration

General Format

```
selector::pseudo-element {
  property: value;
}
```

Notice pseudo-elements have two colons!

CSS Pseudo-Elements

`::first-line` - Select the first line of an element

`::first-letter` - Select the first letter of an element

`::before` - Insert something before an element

`::after` - Insert something after an element

`::selection` - Select the portion of an element currently selected by the user

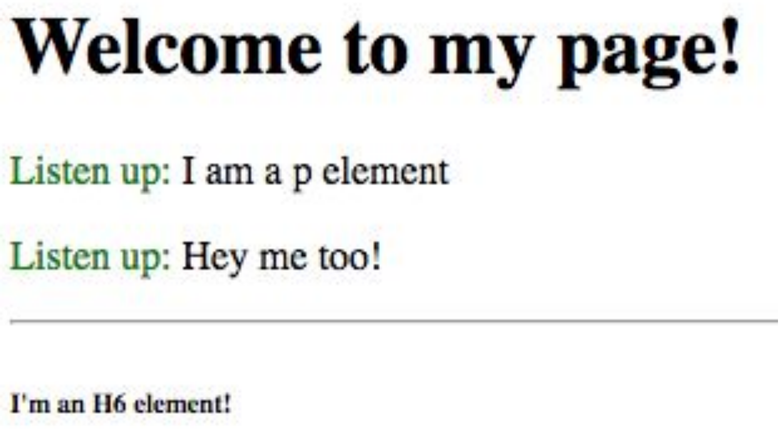
Insert Content Before

```
p::before {
   content: “Listen up: ”;
   color: green;
}
```

Inserts a green “Listen up: ” before every p element. Helps us avoid repeated code!

Insert Content Before

```
p::before {
   content: “Listen up: ”;
   color: green;
}
```


Welcome to my page!

Listen up: I am a p element

Listen up: Hey me too!

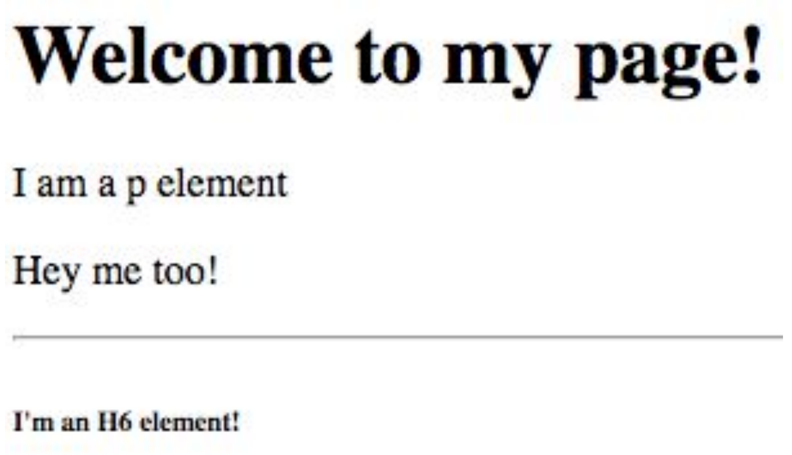
I'm an H6 element!

Style the User’s Selection

```
p::selection {
   background-color: DarkBlue;
   color: white;
}
```

Style the User’s Selection

```
p::selection {
   background-color: DarkBlue;
   color: white;
}
```



Welcome to my page!

I am a p element

Hey me too!

I'm an H6 element!

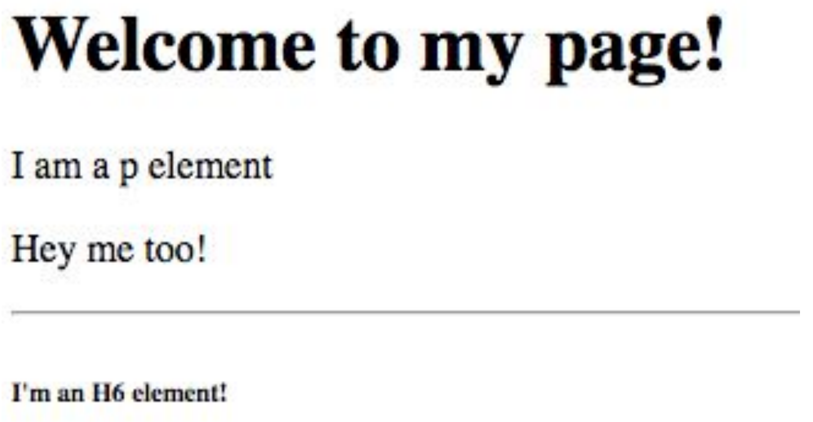
Combining a Pseudo-Class and Pseudo-Element

```
p:active::after {
   content: " Thanks!"
}
```

What do you think this will do?

Combining a Pseudo-Class and Pseudo-Element

```
p:active::after {
   content: " Thanks!"
}
```



Welcome to my page!

I am a p element

Hey me too!

I'm an H6 element!

Let’s look at some examples