

ASSIGNMENT - 1

WEB DESIGNING

Abhay Shanker Pathak

March 2020

About

This assignment contains screenshots of solved examples
for **Cascading Style Sheet**

Contents

1	Introduction	1
2	Adding CSS	1
2.1	External Style Sheet	1
2.2	Embedded Style Sheet	2
2.3	Imported Style Sheet	2
2.4	Inline Style Sheet	2
3	Selectors In CSS	3
3.0.1	Simple Selector	3
3.0.2	Grouping:	3
3.0.3	Universal selectors:	4
3.0.4	Descendent Selector	4
3.0.5	Child Selector	5
3.0.6	Class Selector	6
3.0.7	ID Selector	6
3.0.8	Pseudo Class	7
3.0.9	Pseudo Element	7

List of Tables

List of Figures

1	Showing external style sheet with vsplit	1
2	Example of Embedded Style Sheet	2
3	example showing imported style sheet	2
4	example showing inline style sheet	3

5	an example of simple selector	3
6	example of grouping	4
7	example of universal selector	4
8	exaple showing descendant selectors	5
9	Child selector example	5
10	Screenshot showing class selector	6
11	example of ide selector	6
12	this is how pseudo class works	7
13	second example of pseudo class selector	7
14	third example of pseudo class selector	8
15	fourth example of pseudo class selector	8
16	first example of pseudo element selector	9
17	second example of pseudo element selector	9
18	third example of pseudo element selector	10

1 Introduction

- **Style Sheet** is a document that contains style information about one or more documents written in markup languages.
- A style sheet is composed of a set of style rules written in a specified format.
- **Cascading Style Sheet (CSS)** is a style sheet language that specifies how to incorporate style information in a style sheet. The term cascading indicates that several style sheets can be blended to present a document on the browser's screen.

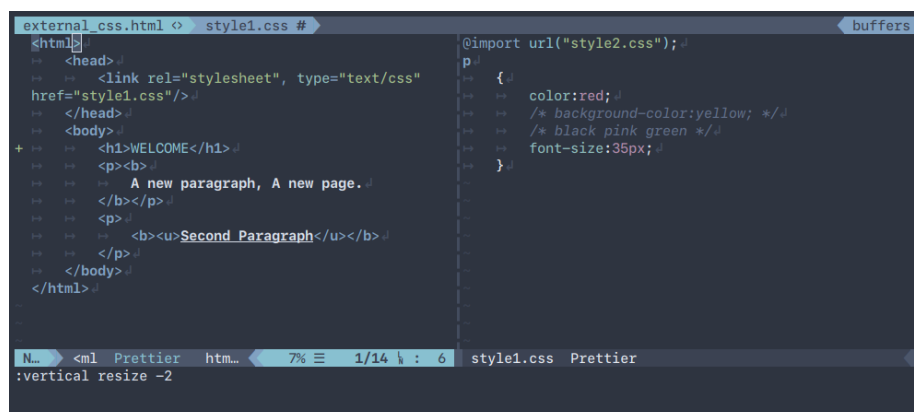
2 Adding CSS

- There are four ways to specify style information in a document
 - External Style Sheet
 - Embedded Style Sheet
 - Imported Style Sheet
 - Inline Style Sheet

2.1 External Style Sheet

- In this case style information is written in a separate file and referenced from an HTML document.
- An external style sheet is useful when the same style is applied on different documents.
- The external style sheet is specified using the HTML `<link>` tag.
- `<link rel = "stylesheet" type = "text/css" href = "mystyle.css" / >`

Below is the example code for **External Style Sheet**



```
external_css.html < style1.css # buffers
<html>
<head>
<link rel="stylesheet", type="text/css"
href="style1.css"/>
</head>
<body>
<h1>WELCOME</h1>
<p><b>
A new paragraph, A new page.
</b></p>
<p>
<b><u>Second Paragraph</u></b>
</p>
</body>
</html>

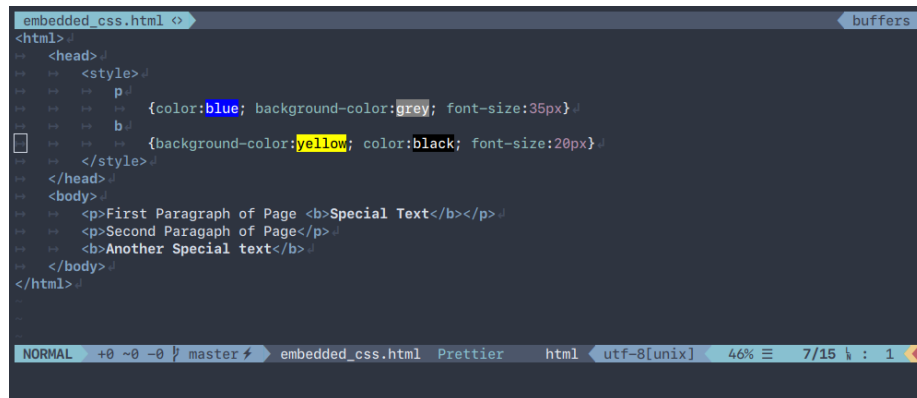
@import url("style2.css");
p {
color:red;
/* background-color:yellow; */
/* black pink green */
font-size:35px;
}
```

Figure 1: Showing external style sheet with vsplit

2.2 Embedded Style Sheet

- In this method style information is placed under the style tag in the head section of an HTML page.

See the example to learn more

A screenshot of a code editor with a dark theme. The editor shows an HTML file named 'embedded_css.html'. The code is as follows:

```
<html>
<head>
  <style>
    p {color:blue; background-color:grey; font-size:35px}
    b {background-color:yellow; color:black; font-size:20px}
  </style>
</head>
<body>
  <p>First Paragraph of Page <b>Special Text</b></p>
  <p>Second Paragraph of Page</p>
  <b>Another Special text</b>
</body>
</html>
```

The code is syntax-highlighted. The status bar at the bottom shows 'NORMAL', '+0 -0 -0', 'master', 'embedded_css.html', 'Prettier', 'html', 'utf-8[unix]', '46%', '7/15', and '1'.

Figure 2: Example of Embedded Style Sheet

2.3 Imported Style Sheet

- Another way of importing stylesheet is to use `@import` statement. It allows importing style sheet from another style sheet.

Here's the example below of imported style sheet

A screenshot of a code editor with a dark theme. The editor shows an HTML file named 'imported_css.html'. The code is as follows:

```
<html>
<head>
  <style>
    @import url("style1.css");
  </style>
</head>
<body>
  <p>
    <h1>WELCOME</h1>
    hello
  </p>
</body>
</html>
```

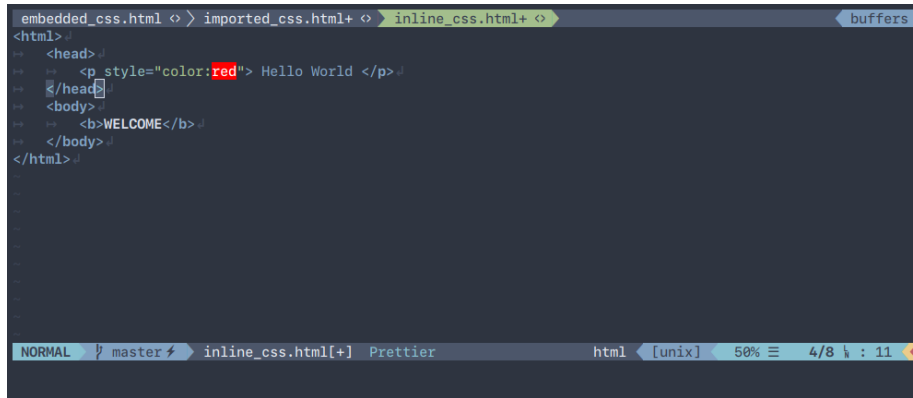
The code is syntax-highlighted. The status bar at the bottom shows 'NORMAL', '+0 -1 -0', 'master', 'imported_css.html[+]', 'html', 'utf-8[unix]', '69%', '9/13', and '23'.

Figure 3: exaple showing imported style sheet

2.4 Inline Style Sheet

- In this case style information is incorporated directly into the HTML tags.
- `<p style = "color:red" >Hello World </p>`

See the example of inline style sheet

A screenshot of a code editor with a dark theme. The editor shows an HTML file named 'inline_css.html'. The code is as follows:

```
<html>
<head>
  <p style="color:red"> Hello World </p>
</head>
<body>
  <b>WELCOME</b>
</body>
</html>
```

The status bar at the bottom indicates the file is 'inline_css.html', uses 'Prettier' formatting, is in 'html' mode, on a 'unix' system, at 50% zoom, and the cursor is at line 4, column 8.

Figure 4: exaple showing inline style sheet

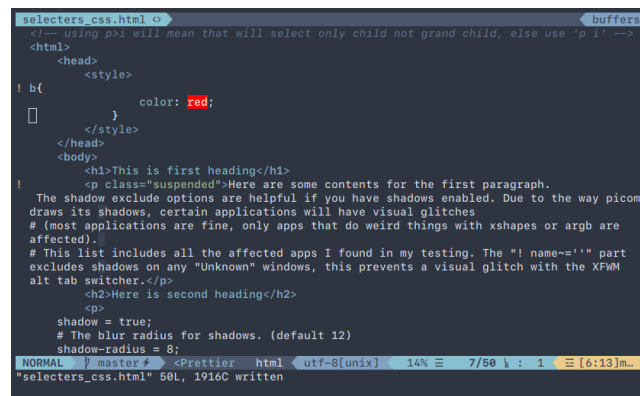
3 Selectors In CSS

Selector determines on which rules are to be applied. The elements selected by the selector are called **subjects of selectors**.

Here are few types of selectors:

3.0.1 Simple Selector

Element is selected by it's name.

A screenshot of a code editor with a dark theme. The editor shows an HTML file named 'selectors_css.html'. The code is as follows:

```
<!-- using p>1 will mean that will select only child not grand child, else use 'p i' -->
<html>
<head>
  <style>
    p {
      color: red;
    }
  </style>
</head>
<body>
  <h1>This is first heading</h1>
  <p class="suspended">Here are some contents for the first paragraph.
  The shadow exclude options are helpful if you have shadows enabled. Due to the way picom
  draws its shadows, certain applications will have visual glitches
  # (most applications are fine, only apps that do weird things with xshapes or argb are
  affected).
  # This list includes all the affected apps I found in my testing. The "! name~="" part
  excludes shadows on any "Unknown" windows, this prevents a visual glitch with the XFWM
  alt tab switcher.</p>
  <h2>Here is second heading</h2>
  <p>
    shadow = true;
    # The blur radius for shadows. (default 12)
    shadow-radius = 8;
  </p>
</body>
</html>
```

The status bar at the bottom indicates the file is 'selectors_css.html', uses 'Prettier' formatting, is in 'html' mode, on a 'unix' system, at 14% zoom, and the cursor is at line 1, column 1.

Figure 5: an example of simple selector

3.0.2 Grouping:

Selectors having common declaration are grouped into a common separated list

here's the screenshot

```

selectors_css.html
buffers
<!-- using p>i will mean that will select only child not grand child, else use 'p i' -->
<html>
<head>
<style>
! h1, p, i {
    color: red;
}
</style>
</head>
<body>
<h1>This is first heading</h1>
<p>Here are some contents for the first paragraph.
The shadow exclude options are helpful if you have shadows enabled. Due to the way picom draws its shadows, certain applications will have
visual glitches
# (most applications are fine, only apps that do weird things with xshapes or argb are affected).
# This list includes all the affected apps I found in my testing. The '! name=""' part excludes shadows on any "Unknown" windows, this
prevents a visual glitch with the XFWM alt tab switcher.</p>
<h2>Here is second headings</h2>
<p>
shadow = true;
# The blur radius for shadows. (default 12)
shadow-radius = 8;
# The left offset for shadows. (default -15)
shadow-offset-x = -2;
# The top offset for shadows. (default -15)
shadow-offset-y = 0;
# The translucency for shadows. (default .75)
shadow-opacity = 1;
</p>
<h3>Here is third and maybe last heading</h3>
<p>
NORMAL +2 -1 -0 } master # selectors_css.html Prettier html utf-8[unix] 10% 5/50 9 [6:13]mix-indent-file
"selectors_css.html" 50L, 1898C written

```

Figure 6: example of grouping

3.0.3 Universal selectors:

CSS has special type of selector * which matches with every single element in the document

Showed in following screenshot

```

selectors_css.html
buffers
<!-- using p>i will mean that will select only child not grand child, else use 'p i' -->
<html>
<head>
<style>
! *{
    color: red;
}
</style>
</head>
<body>
<h1>This is first heading</h1>
<p>Here are some contents for the first paragraph.
The shadow exclude options are helpful if you have shadows enabled. Due to the way picom draws its shadows, certain applications will have
visual glitches
# (most applications are fine, only apps that do weird things with xshapes or argb are affected).
# This list includes all the affected apps I found in my testing. The '! name=""' part excludes shadows on any "Unknown" windows, this
prevents a visual glitch with the XFWM alt tab switcher.</p>
<h2>Here is second headings</h2>
<p>
shadow = true;
# The blur radius for shadows. (default 12)
shadow-radius = 8;
# The left offset for shadows. (default -15)
shadow-offset-x = -2;
# The top offset for shadows. (default -15)
shadow-offset-y = 0;
# The translucency for shadows. (default .75)
shadow-opacity = 1;
</p>
<h3>Here is third and maybe last heading</h3>
<p>
NORMAL +2 -1 -0 } master # selectors_css.html Prettier This is first heading < html utf-8[unix] 24% 12/50 1 [6:13]mix-indent-file

```

Figure 7: example of universal selector

3.0.4 Descendent Selector

Descendent selectors, allow us to determine the elements depending upon their hierarchical relationship.

For example:

```

selectors_css.html
<!-- using p>i will mean that will select only child not grand child, else use 'p i' -->
<html>
  <head>
    <style>
! p u i{
    color: red;
    }
  </style>
</head>
<body>
  <h1>This is first heading</h1>
  <p>Here are some contents for the first paragraph.
  The shadow exclude options are helpful if you have shadows enabled. Due to the way picom draws its shadows, certain applications will have
  visual glitches
  # (most applications are fine, only apps that do weird things with xshapes or argb are affected).
  # This list includes all the affected apps I found in my testing. The "! name=*" part excludes shadows on any "Unknown" windows, this
  prevents a visual glitch with the XFWM alt tab switcher.</p>
  <h2>Here is second headings</h2>
  <p>
    shadow = true;
    # The blur radius for shadows. (default 12)
    shadow-radius = 8;
    # The left offset for shadows. (default -15)
    shadow-offset-x = -2;
    # The top offset for shadows. (default -15)
    shadow-offset-y = 0;
    # The translucency for shadows. (default .75)
    shadow-opacity = 1;
  </p>
  <h3>Here is third and maybe last heading</h3>
  <p>
NORMAL +2 -2 -0 } master # selectors_css.html Prettier html utf-8[unix] 8% 4/50 1 : 15 [47:13]mix-indent-file
"selectors_css.html" 50L, 1902C written

```

Figure 8: exaple showing descendant selectors

3.0.5 Child Selector

Child selector select elements that are immediate children of a specified element. The combinator used for child selector is “>”.

For example:

```

selectors_css.html
<!-- using p>i will mean that will select only child not grand child, else use 'p i' -->
<html>
  <head>
    <style>
! p>h3{
    color: red;
    }
  </style>
</head>
<body>
  <h1>This is first heading</h1>
  <p>Here are some contents for the first paragraph.
  The shadow exclude options are helpful if you have shadows enabled. Due to the way picom draws its shadows, certain applications will have
  visual glitches
  # (most applications are fine, only apps that do weird things with xshapes or argb are affected).
  # This list includes all the affected apps I found in my testing. The "! name=*" part excludes shadows on any "Unknown" windows, this
  prevents a visual glitch with the XFWM alt tab switcher.</p>
  <h2>Here is second headings</h2>
  <p>
    shadow = true;
    # The blur radius for shadows. (default 12)
    shadow-radius = 8;
    # The left offset for shadows. (default -15)
    shadow-offset-x = -2;
    # The top offset for shadows. (default -15)
    shadow-offset-y = 0;
    # The translucency for shadows. (default .75)
    shadow-opacity = 1;
  </p>
  <h3>Here is third and maybe last heading</h3>
  <p>
NORMAL +2 -2 -0 } master # selectors_css.html[+] Prettier html utf-8[unix] 12% 6/50 1 : 1 [6:13]mix-indent-file

```

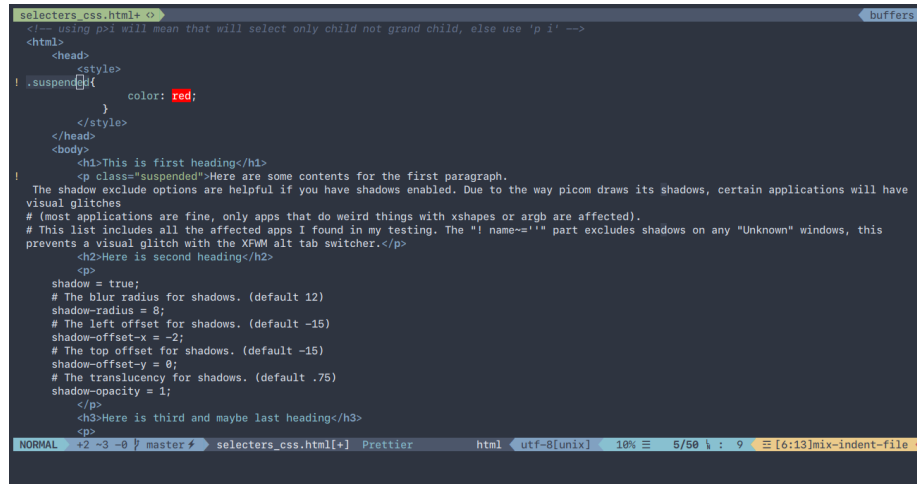
Figure 9: Child selector example

Only area of <p >under <h3 >will be highlighted.

Child selector can be combined with other selectors e.g., Body<*> Selects all children of the <body>element. Body<*><*> Selects all grandchildren of the body element. Body<*><p> Selects all grandchildren of <body>element.

3.0.6 Class Selector

These selector provide a flexible way to apply style elements. Class selectors deal with the elements having that attribute class. To select elements with a specific class, write a period(.) character, followed by the name of the class.



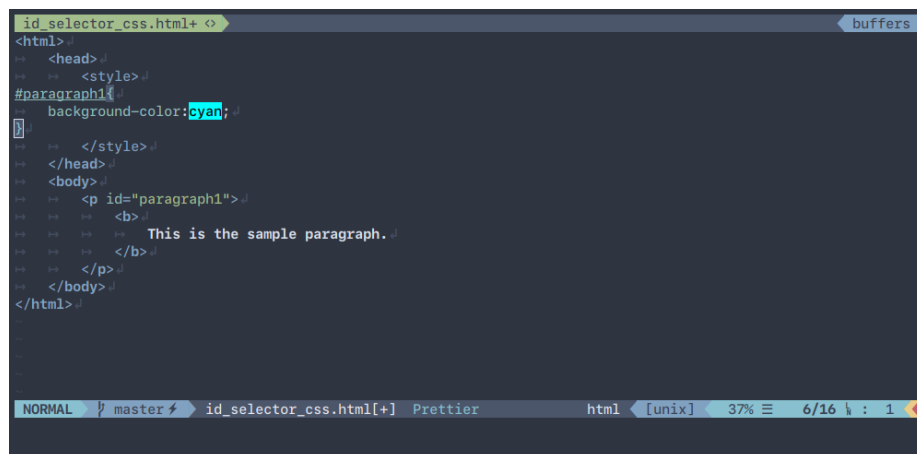
```
selectors_css.html+ <> buffers
<!-- using p.i will mean that will select only child not grand child, else use 'p i' -->
<html>
  <head>
    <style>
      ! .suspended{
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>This is first headings</h1>
    <p class="suspended">Here are some contents for the first paragraph.
    The shadow exclude options are helpful if you have shadows enabled. Due to the way picom draws its shadows, certain applications will have
    visual glitches
    # (most applications are fine, only apps that do weird things with xshapes or argb are affected).
    # This list includes all the affected apps I found in my testing. The "! name==" part excludes shadows on any "Unknown" windows, this
    prevents a visual glitch with the XFWM alt tab switcher.</p>
    <h2>Here is second headings</h2>
    <p>
      shadow = true;
      # The blur radius for shadows. (default 12)
      shadow-radius = 0;
      # The left offset for shadows. (default -15)
      shadow-offset-x = -2;
      # The top offset for shadows. (default -15)
      shadow-offset-y = 0;
      # The translucency for shadows. (default .75)
      shadow-opacity = 1;
    </p>
    <h3>Here is third and maybe last heading</h3>
  </body>
</html>
```

Figure 10: Screenshot showing class selector

3.0.7 ID Selector

The attribute id of an element is a unique identifier in a web page. This means that no two id attribute can have the same value within the document. The id differs from class in that id identifies a single element uniquely whereas class identifies a set of an identity.

An id selector is defined by placing a # symbol before selector name



```
id_selector_css.html+ <> buffers
<html>
  <head>
    <style>
      #paragraph1{
        background-color:cyan;
      }
    </style>
  </head>
  <body>
    <p id="paragraph1">
      <b>
        This is the sample paragraph.
      </b>
    </p>
  </body>
</html>
```

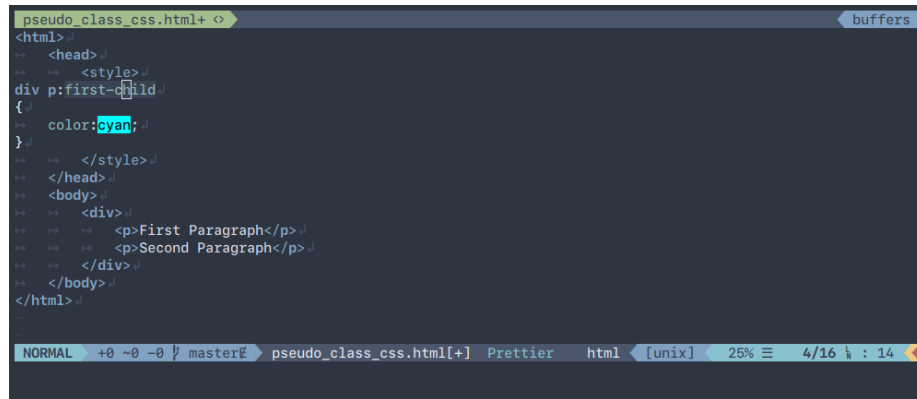
Figure 11: example of ide selector

3.0.8 Pseudo Class

Pseudo class is a selector that selects the elements that are in specific states, e.g., They are the first element of their type or they are being hovered by mouse pointer.

Pseudo classes are keywords that start with a colon

Here's an example

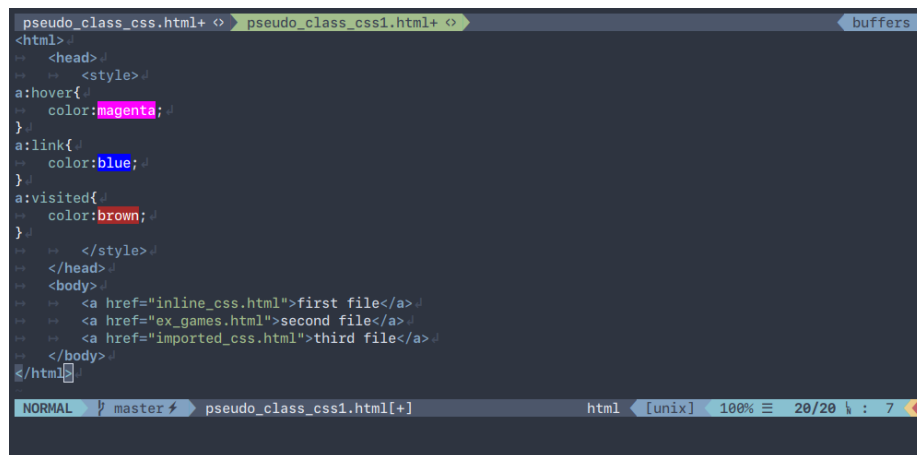


```
pseudo_class_css.html+ buffers
<html>
<head>
<style>
div p:first-child
{
color:cyan;
}
</style>
</head>
<body>
<div>
<p>First Paragraph</p>
<p>Second Paragraph</p>
</div>
</body>
</html>
```

The screenshot shows a code editor with a dark theme. The file name is 'pseudo_class_css.html+'. The code defines a CSS rule for 'div p:first-child' with the color 'cyan'. The HTML body contains a 'div' with two paragraphs. The first paragraph is highlighted in cyan in the original image. The status bar at the bottom shows 'NORMAL', '+0 ~0 -0', 'master', 'pseudo_class_css.html[+]', 'Prettier', 'html', '[unix]', '25%', '4/16', and '14'.

Figure 12: this is how pseudo class works

Here are few more examples of pseudo class selectors:



```
pseudo_class_css.html+ pseudo_class_css1.html+ buffers
<html>
<head>
<style>
a:hover{
color:magenta;
}
a:link{
color:blue;
}
a:visited{
color:brown;
}
</style>
</head>
<body>
<a href="inline_css.html">first file</a>
<a href="ex_games.html">second file</a>
<a href="imported_css.html">third file</a>
</body>
</html>
```

The screenshot shows a code editor with two tabs: 'pseudo_class_css.html+' and 'pseudo_class_css1.html+'. The code defines three CSS rules for links: 'a:hover' with color 'magenta', 'a:link' with color 'blue', and 'a:visited' with color 'brown'. The HTML body contains three links. The status bar at the bottom shows 'NORMAL', 'master', 'pseudo_class_css1.html[+]', 'html', '[unix]', '100%', '20/20', and '7'.

Figure 13: second example of pseudo class selector

3.0.9 Pseudo Element

Pseudo element behave in similar way, however they act as if you have added a whole new html element into the markup, rather than applying a class to existing elements.

```
pseudo_class_css.html+ <> pseudo_class_css1.html+ <> pseudo_class_css2.html <> buffers
<html>
<head>
<style>
rd1:checked{
height:75px;
width:100px;
}
</style>
</head>
<body>
<input type="radio" checked class="rd1" name="gp1"/>radio1
<input type="radio" checked class="rd1" name="gp1"/>radio2
<input type="radio" checked class="rd1" name="gp1"/>radio3
</body>
</html>

NORMAL +0 -0 -0 master pseudo_class_css2.html Prettier html [unix] 46% 7/15 : 1
:Buffers
```

Figure 14: third example of pseudo class selector

```
... pseudo_class_css1.html+ <> pseudo_class_css2.html <> pseudo_class_css3.html+ <> buffers
<html>
<head>
<style>
input:invalid{
background-color:red;
}
input:required{
background-color:blue;
}
</style>
</head>
<body>
<input type="email"/>
<input type="test" require/>
</body>
</html>

NORMAL master pseudo_class_css3.html+ html [unix] 68% 11/16 : 11
```

Figure 15: fourth example of pseudo class selector

Pseudo elements start with double colon ::
Here are few examples of pseudo element selector

```
...> pseudo_class_css2.html <> pseudo_class_css3.html+ <> pseudo_class_css4.html+ <> buffers
<html>
<head>
<style>
p::first-letter{
color:green;
}
</style>
</head>
<body>
<p><b>Welcome</b>
This is a sample paragraph.
</p>
</body>
</html>

NORMAL master pseudo_class_css4.html[+] html [unix] 42% 6/14 : 9
```

Figure 16: first example of pseudo element selector

```
id_selector_css.html+ <> selectors_css1.html+ <> selectors_css2.html+ <> buffers
<html>
<head>
<style>
.abox::before{
content:"this should show before other content";
color:yellow;
}
</style>
</head>
<body>
<p class="abox">
<u>
This is a sample paragraph for a box.
</u>
</p>
</body>
</html>

NORMAL +0 ~0 master selectors_css1.html[+] Prettier html [unix] 29% 5/17 : 1
```

Figure 17: second example of pseudo element selector

```
id_selector_css.html+ <> > selectors_css1.html+ <> selectors_css2.html+ <> buffers
<html>
  <head>
    <style>
      .antbox::after{
        content:"this should show after the other content";
        color:yellow;
      }
    </style>
  </head>
  <body>
    <p class="antbox">
      <b><i>Here's a sample paragraph for another box.</i></b>
    </p>
  </body>
</html>

NORMAL master selectors_css2.html[+] Prettier html [unix] 53% 8/15 : 16
```

Figure 18: third example of pseudo element selector