EXPLORING CSS SELECTOR

A selector is a pattern that is used to select to apply the css style rules

Selector can be used as a condition

The css rule is divided into two parts

1. Selectors

2.Declaration

- The rules defined in the declaration part are applied the element specified by the selector the different types of selectors are the follows
- The universal selector
- The type selector
- The class selector
- The id selector
- The child selector
- The descendant selector
- The adjacent sibling selector
- The attribute selector
- The query selector

THE UNIVERSAL SELECTOR:

The universal selector selects the all the elements that are present in a html document you can use this selector to apply the same rule to all the elements of an html it is represented by an asterisk symbol (*{}).

SYNTAX: *{

```
            margin:0;

            padding:0;

      }
```

THE TYPE SELECTOR:

The type selector matches all the elements specified in a list with the given value to determine the elements to which the CSS rules

SYNTAX:

P, h2, h3 {font-size:30px}

THE CLASS SELECTOR:

The class selector allows you to apply CSS rules to the elements that carry a class attribute whose value matches with the class attribute specified in the selector

SYNTAX :<h1 class=”intro”>Header 1</H1>

Applying the CSS rule to all the elements that have the class attribute of the same value .The following code snippet shows how to apply the CSS rule

SYNTAX: .intro {font-family :fantasy}

Applying the CSS rule to the H1 element whose class attribute contains into as its value the

Following code snippet shows how to apply the CSS style on H1element

SYNTAX: h1.intro{font-family :fantasy}

THE ID SECTOR:

The value of the id attribute is unique with in a document; therefore the selector is applied only to the content of one element the following code snippet shows the h1 element having my Header as the value of the attribute

SYNTAX:

<H1 id =”selector”>Hello World!</H1>

The following code snippet shows the id selector ,which is represented by a hash symbol(#)and

Followed by the value of the id attribute

SYNTAX:

#selector{font-color:green}

THE CHILD SELECTOR:

The child selector matches the element that is an immediate child of another element.

Selector greater than symbol(>) is used as the combinator

SYNTAX:

table>th{font-family: sans-serif}

A combinator is a symbol ,such as >,<and + which shows the relationship between two elements

THE DESENDANT SELECTOR:

The descendant selector matches an element that is a descendant of another element .A descendant element is an element that is nested inside another element.

Selector white space is used as the combinator as shown in the following code snippet

SYNTAX: table th{font-family :sans-serif}

THE ADJACENT SIBLING SELECTOR :

the adjacent sibling selector selects all the elements that are adjacent siblings of a specified element.Sibling elements must have the same parent element .The word adjacent means side-by-side so no other element could exist between that adjacent sibling element

SYNTAX:

H2+p{font-weight:bold}

Lets apply the preceding code snippet of a CSS file to the following HTML code snippet

<H2>Heading</H2>

<p>the selector above matches this paragraph</p>

<p>the selector above does not match this paragraph</P>

THE ATTRIBUTE SELECTOR:

The CSS attribute selector selects elements on the basis of some specific attributes or attribute values table lists the most common type of attribute

Selectors.

Name	Syntax	Match	Example

Hyphen selector	[attribute|=value]	Matches if the element has an attribute followed by a hyphen	[lang|=fr]{
			Background-color:red;
			}

Existence selector	[attribute]	Matches if element has specific attribute	a[title]{
			Color:green;
			}

Equality selector	[attribute=value]	Matches if the element has an attribute with a specific value	a[href=http://wp.com/]
			{
			Font-color:pink;
			}

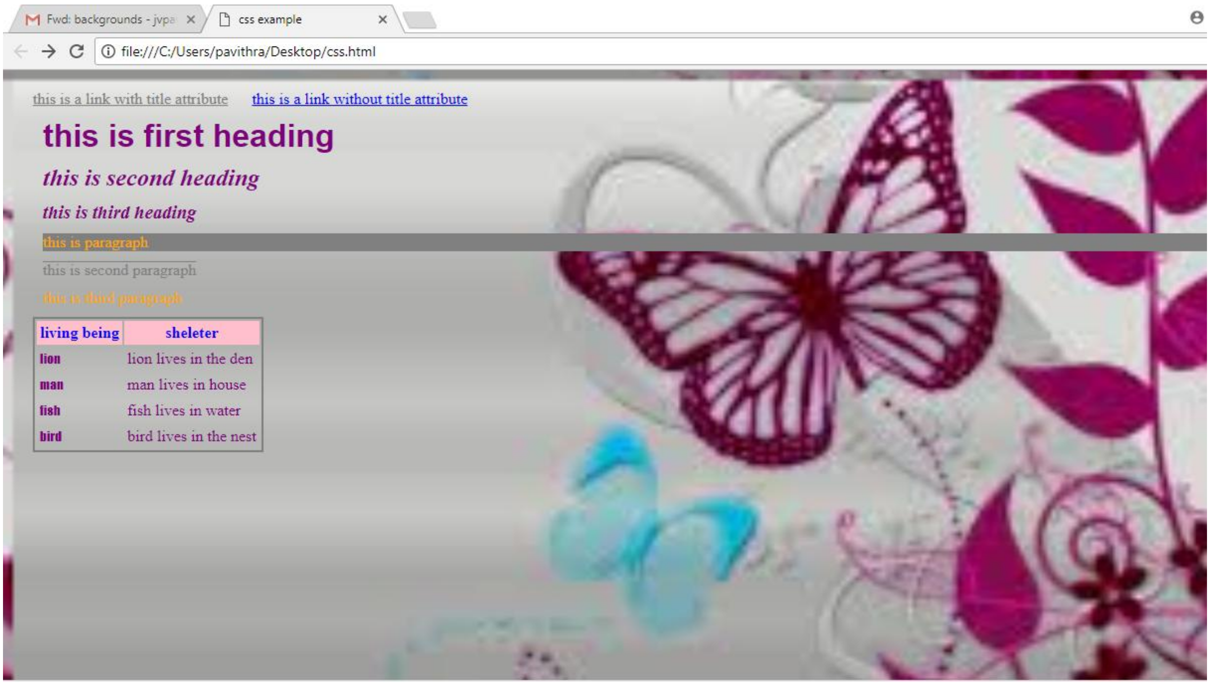
Space selector	[attribute~=value]	Matches if element has an element with space separated items that match with the value	A[title~=web]
			{
			Background-color:cyan;
			}

THE QUERY SELECTOR:

The querySelector() and querySelectorAll() methods accept CSS selectors as parameters and return the matching element node in document tree

```
    <html>
    <head>
        <title>css example</title>
        <link rel="stylesheet"type="text/css"href="example.css">
        <a href="www.google.com" title="google">this is a link with title attribute</a>
        <a href="www.wp.com">this is a link without title attribute</a>
    </head>
    <body>
        <div id="div">
            <h1 class="code1">this is first heading</h1>
            <h2>this is second heading</h2>
            <h3>this is third heading</h3>
                <p>this is paragraph</p>
                <p id="code2">this is second paragraph</p>
                <p>this is third paragraph</p>

        </div>
        <table border="0">
            <tr>
                <th>living being</th>
                <th>sheleter</th>
                </tr>

                <tr>

                        <td class="code">lion</td>

                        <td>lion lives in the den</td>

                </tr>

                <tr>

                        <td class="code">man</td>

                        <td>man lives in house</td>

                </tr>

                <tr>

                        <td class="code">fish</td>

                        <td>fish lives in water</td>

                </tr>

                <tr>

                        <td class="code">bird</td>

                        <td>bird lives in the nest</td>

                </tr>

        </table>

    </body>

</html>
```

Result:



this is a link with title attribute   this is a link without title attribute

this is first heading

this is second heading

this is third heading

this is paragraph

this is second paragraph

this is third paragraph

living being	sheleter
lion	lion lives in the den
man	man lives in house
fish	fish lives in water
bird	bird lives in the nest

Code of css file:

```
*{margin:10; }

body

{

        color:purple;

        background-color:#fff00;

        background-image:url(image3.jpg);

        background-size:cover;

}

.code1

{
        font-family:sans-serif;
}
h2,h3{font-style:italic}
p{font-size:12pt}
table
{
        background-color:magneta;
        border-style:solid;
        border-width:2px;
        border-color:magneta;
}
th
{
        background-color:pink;
        font-weight:bold;
        padding:3px;
}
td{padding:3px}
.code
{
        font-size:10pt;
        font-family:fantasy;
```

```
	font-weight:lighter;

}

#code2

{

	text-decoration:overline;

}

p[id]

{

	color:grey;

}

a[title]

{

	color:grey;

}

div>p{color:orange}

table th{color:blue}

#div h3+p{background:grey}
```

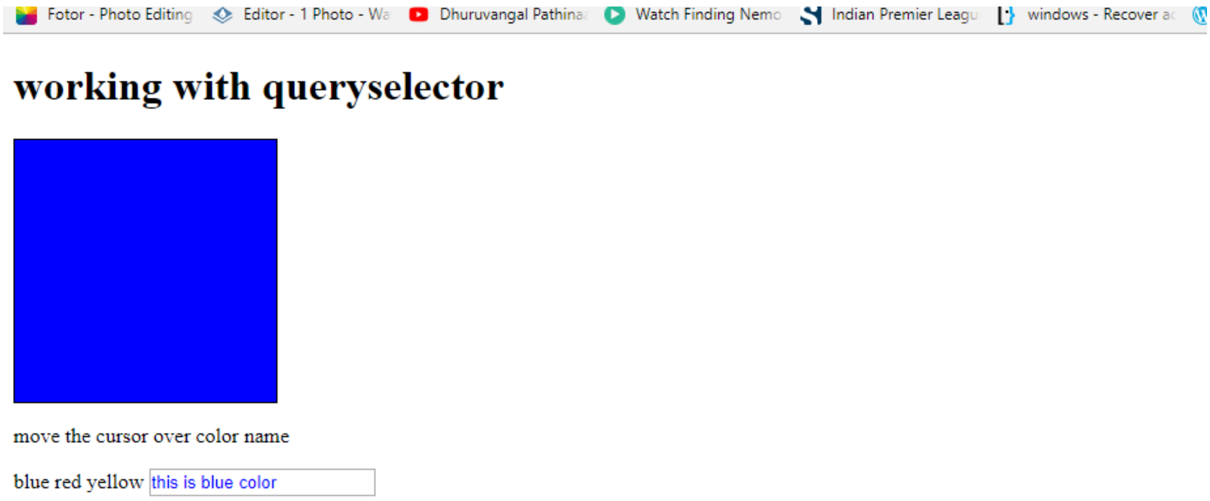
Working with the querySelector() method :

```
<!doctype html>

<html>

<head>
```

```
        <title>working with queryselector</title>
</head>
<body>
        <h1>working with queryselector</h1>
        <div id="div1" style="padding:50px; width:100px;
height:100px;border:1px solid black">
        </div>
        <p> move the cursor over color name</p>
        <label id="label1"> blue</label>
        <label id="label2"> red</label>
        <label id="label3"> yellow</label>
        <input id="text">
        <script type="text/javascript">
                if(document.querySelector)
                {
                        var lb1=document.querySelector('#label1')
                        var lb2=document.querySelector('#label2')
                        var lb3=document.querySelector('#label3')


                        lb1.onmouseover=function()
                        {
                                document.querySelector('#text').value="this is
blue color";

        document.querySelector('#text').style.color="blue";

        document.querySelector('#div1').style.background="blue"
                        }
                        lb2.onmouseover=function()
                        {
                                document.querySelector('#text').value="this is
red color";

        document.querySelector('#text').style.color="red";

        document.querySelector('#div1').style.background="red"
                        }
                        lb3.onmouseover=function()
                        {
                                document.querySelector('#text').value="this is
yellow color";

        document.querySelector('#text').style.color="yellow";

        document.querySelector('#div1').style.background="yellow"
                        }
```

```
                }
        </script>
</body>
</html>
```

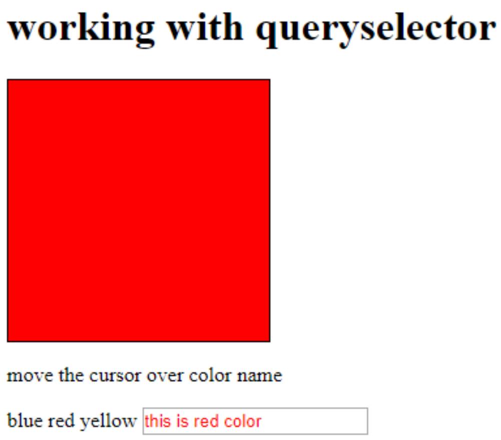
Result:



working with queryselector

move the cursor over color name

blue red yellow this is blue color



working with queryselector

move the cursor over color name

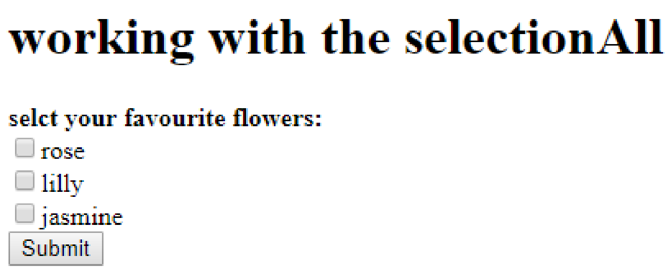
blue red yellow this is red color

Working with querySelectorAll()

Code:

```
<!DOCTYPE HTML>

<head>

<Title>working with queryselecionall</TITLE>

</HEAD>

<BODY>

<H1>working with the selectionAll</H1>

<FORM id="myform">

<B>selct your favourite flowers:</B><BR/>

<INPUT name="flowers" type="checkbox" value="rose" />rose <BR/>

<INPUT name="flowers" type="checkbox" value="lilly" />lilly<BR/>

<INPUT name="flowers" type="checkbox" value="jasmine"
/>jasmine<BR/>

<INPUT type="submit"/>



</FORM>

<SCRIPT type="text/javascript">

if(document.querySelector)

{

        docment.queryselector('#myform').onsubmit=function()

        {
            var
checkflowers=this.querySelectorAll('input[name="flowers"]:checked')

            document.write("<B>you have selected the following flowers:

                  </B></BR>")


            for(vari=0;i<checkflowers.length;i++)

            {

            var value=""value +=checkflowers[i].value+ "<BR/>"

            document.write("<LI>"+value+"<Li/>" )

            }

            return false

      }

}

</SCRIPT>

</BODY>

</HTML>
```

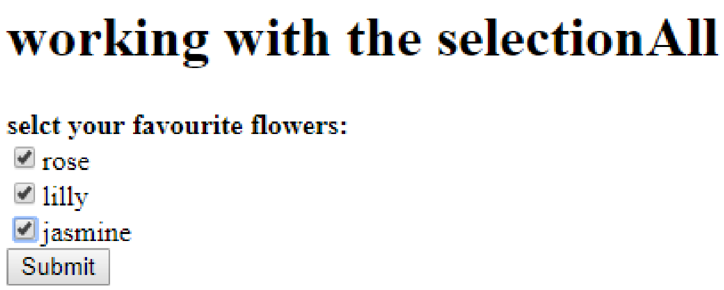
Result:



working with the selectionAll

selct your favourite flowers:

☐ rose
☐ lilly
☐ jasmine



working with the selectionAll

selct your favourite flowers:

☑ rose
☑ lilly
☑ jasmine