

# Github pages

# github.io

easily allows you to host web pages using github servers + workflow

url (uniform resource locator)

http://

yrUsername.github.io



[yrUsername.github.io](https://yrUsername.github.io)

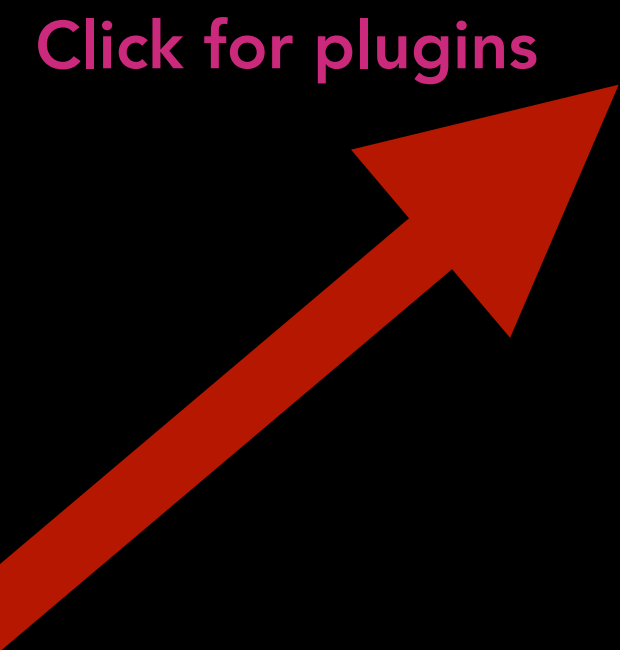
**prototype locally**

# prototype locally

w/ a local http server

**prototype:** local http server  
(using our local machine as a server!!!)

**publish:** pushed to Github Pages

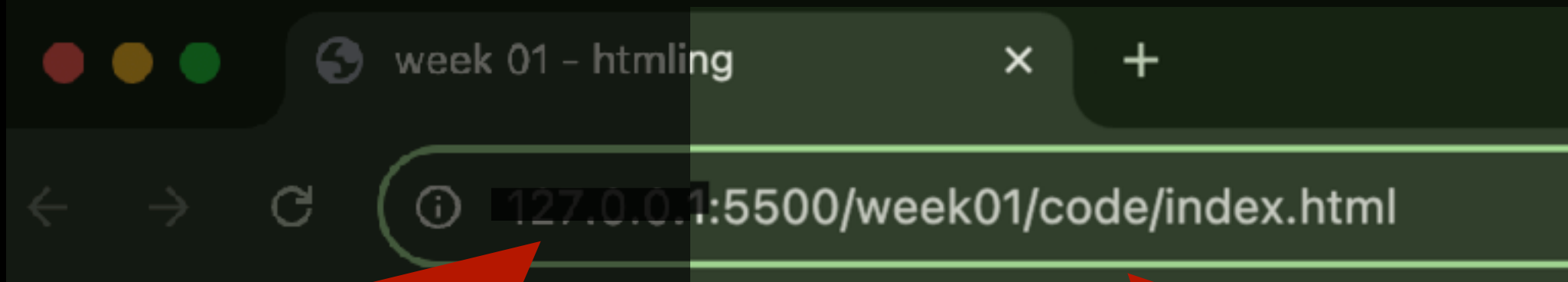


Click for plugins

Search:  
"Live  
Server"

The screenshot shows the Visual Studio Code interface. On the left sidebar, the 'EXTENSIONS' view is active, displaying a list of installed and recommended extensions. The 'Live Server' extension by Ritwick Dey is highlighted in blue. Below it, 'Live Share' by Microsoft and 'Markdown P...' by yzane are listed. Under the 'RECOMMENDED' section, 'markdownlint' by David Anson and 'Microsoft E...' by Microsoft are shown. At the bottom of the sidebar, 'MCP SERVERS' is visible. The main editor area displays the details for the 'Live Server' extension. It features the extension's logo (a purple circle with radiating lines), the name 'Live Server', the author 'Ritwick Dey', and a description 'Launch a development server'. There are buttons for 'Disable' and 'Install'. Below this, there are tabs for 'DETAILS', 'FEATURES', and 'CHANGELOG'. The 'DETAILS' tab is selected, showing a description of the extension's capabilities, including a link to the GitHub repository. At the bottom of the editor, a status bar shows 'vscode marketplace v5.7.9' and 'downloads 69M'. A green circle highlights the 'v5.7.9' version number and the 'downloads 69M' text. Another green circle highlights the 'Go Live' button in the bottom right corner of the status bar.





My great website!

This is your IP address

This is your file path

This is your TCP Port Number

- one
- two
- three

**the Box Model**

# the Box Model

## Border

All boxes have borders even if invisible or 0px wide. It separates the edge of one box from another.

## Padding

Padding is the space btw the border + any content contained within it. More padding increases the readability of its contents.



## Margin

Margins sit outside the edge of the border. You can set the width to create a gap btw borders of adjacent boxes.

**Content**

## HTML - Hyper Text Mark Up

is a grammar for structuring web pages. It defines paragraphs, headings, data tables + media elements. HTML describes the content of the page - not how it looks.

## CSS - Cascading Style Sheet

rules for styling a web page. Setting colors, typeface, and the layout. It can be used to consider the design of your **page across different platforms and screen sizes.**

The key to understanding how **CSS** works is to imagine that there is an invisible box around every **HTML** element.

Block level elements are outlined w/ red + inline elements in green.

**<body>** creates 1st box, then **<h1>**, **<h2>**, **<p>**, **<i>** + **<a>** each create their own boxes within it.

## The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in England and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained English estate gardens.

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

## The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in England and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained English estate gardens.

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

**Quick Getting Ahead of Ourselves:  
responsive web design**

## Metadata: `viewport`

The user's visible area of a web page

HTML5 introduced a method to let web designers take control over the viewport, through the `<meta>` tag.

<!

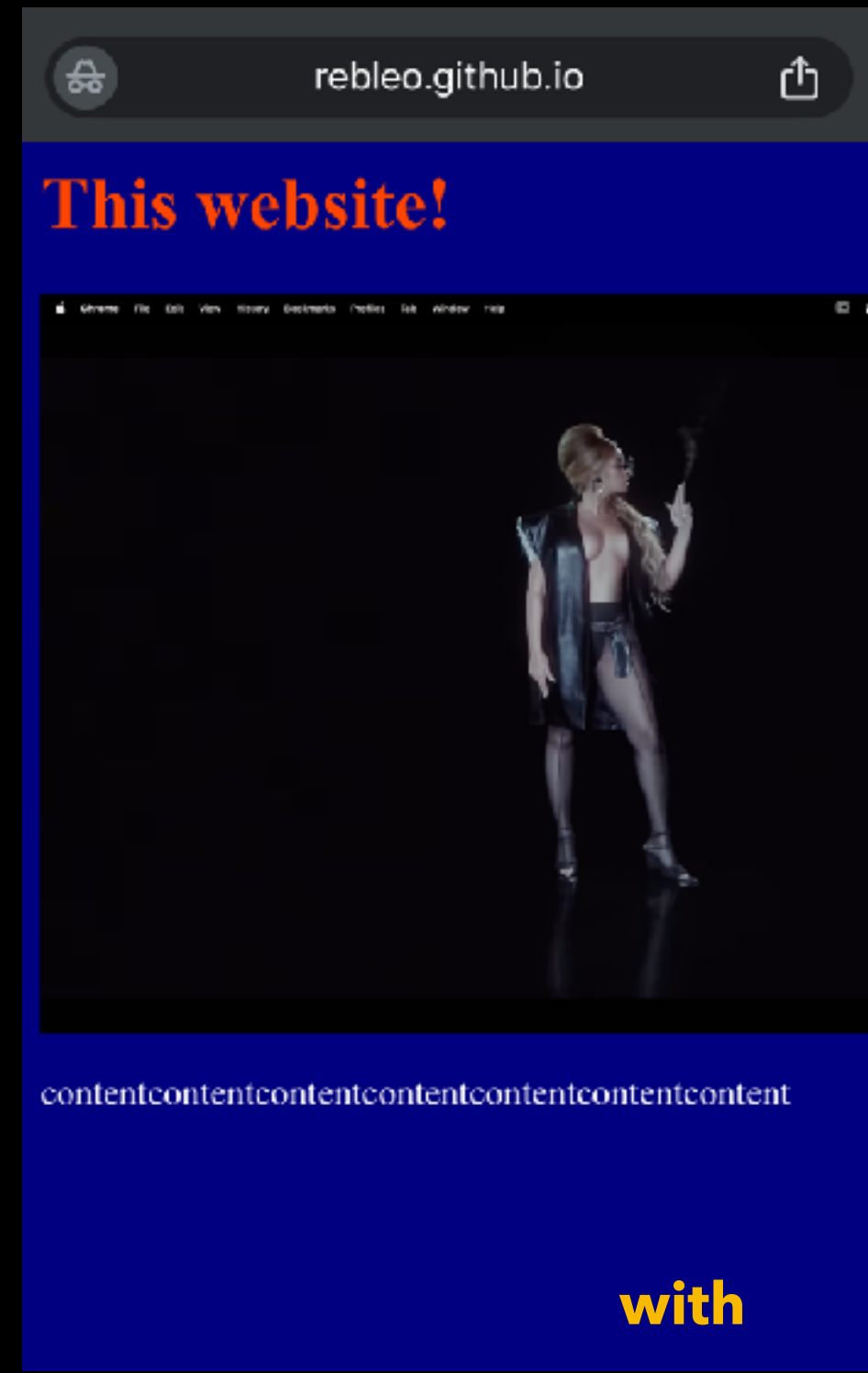
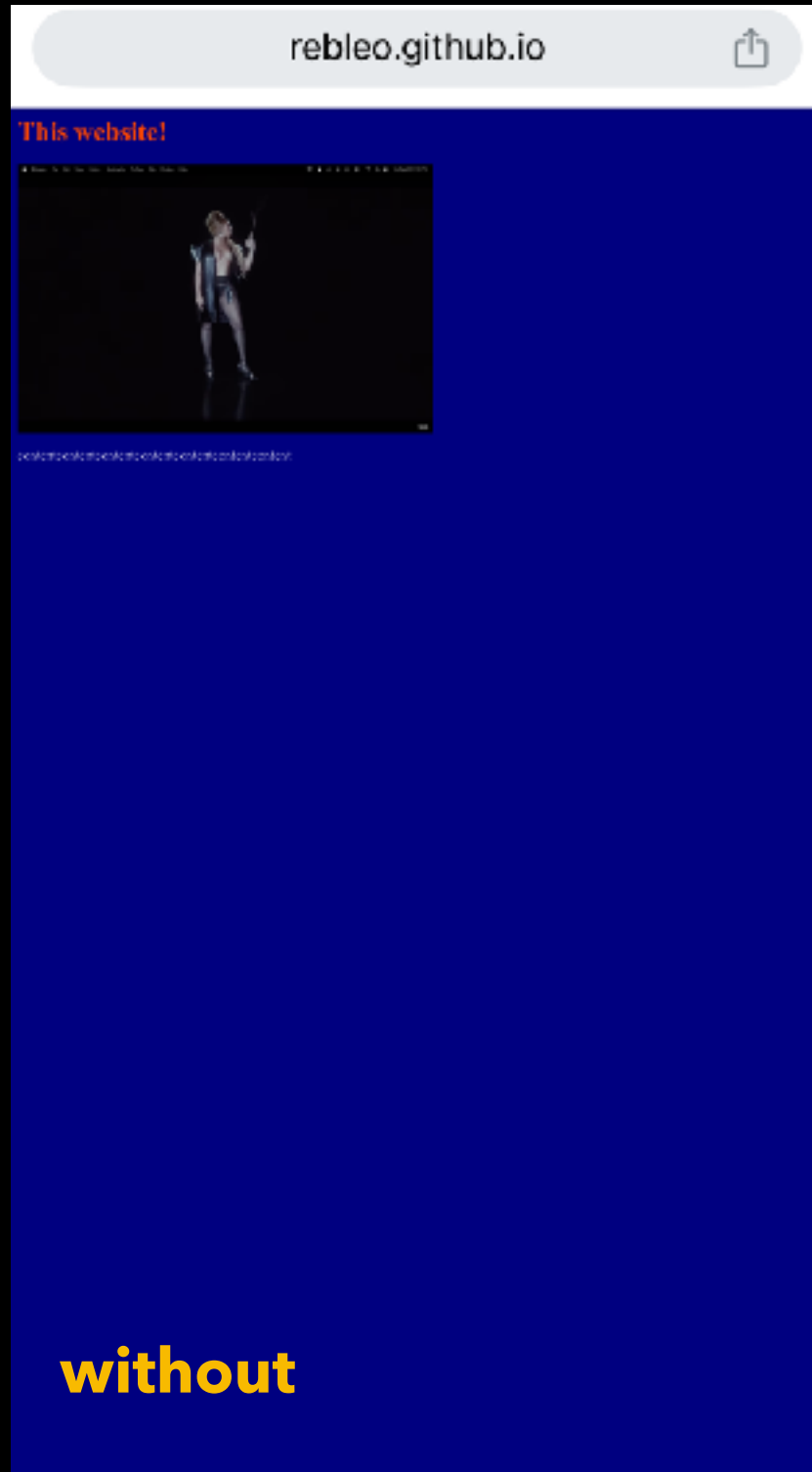
- - Tells the browser to match the device's width for the viewport
- Sets an initial zoom value -->

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

``

`<p>contentcontentcontentcontentcontentcontentcontent</p>`



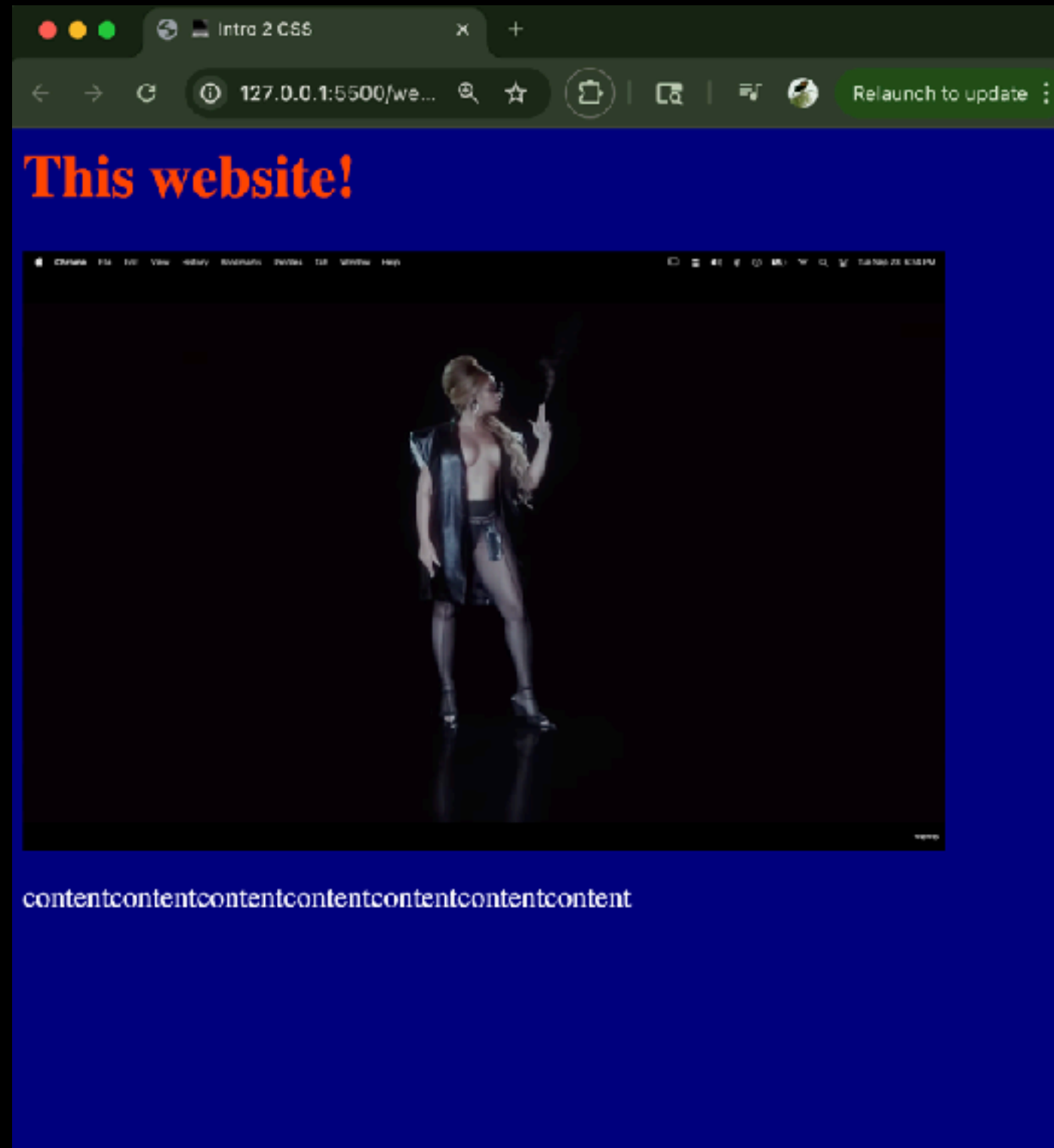


## Inline Styles

```
<h1 style="color:#FF4500;">This Webpage though...</h1>  
<body style="background-color: #000080;">
```

## Embedded Styles

```
<html>  
  <head>  
    <title> 🖥 Intro 2 CSS </title>  
    <style type="text/css">  
      h1 {  
        color: #FF4500  
      }  
  
      body {  
        background: #000080;  
      }  
  
      p {  
        Color:white  
      }  
    </style>  
  </head>
```



## External Styles \*\*

```
<head>
  <title> 🖥 Intro 2 CSS </title>
  <link rel="stylesheet" type="text/css" href="theStyle.css">
</head>
<body>

</body>
```

This website!



contentcontentcontentcontentcontentcontent

```
h1 {  
    color: #FF4500  
}
```

```
p {  
    color: rgb(255,255,255);  
}
```

```
body {  
    background-color: #000080;  
}
```

selector {

property: value ;

}

## Selector

## Meaning

## Example

### Universal Selector

Applies to all elements in the document

```
* { }
```

### Type Selector

Matches element names

```
h1, h2, h3 { }
```

### Class Selector

Matches an element whose class attribute has a value that matches the one specified after the period (or full stop) symbol

```
.theNote { }
```

targets any element whose class attribute has a value of "note"

```
p.note { }
```

targets only `<p>` elements whose class attribute has a value of "note"

### ID Selector

Matches an element whose id attribute has a value that matches then specified after the # symbol

```
#introduction { }
```

targets the element whose id attribute has value of "introduction"

## Selector

## Meaning

## Example

### Child Selector

Matches an element that is a direct child of another

`li > a { }`

targets any `<a>` element that are children of an `<li>` element (but not other `<a>` elements in the page).

### Descendant Selector

Matches an element that is a descendent of another specified element (not just a direct child of that element)

`p a { }`

targets any `<a>` elements that sit inside a `<p>` element, even if there are other elements nested btw them

## Selector

## Meaning

## Example

### Adjacent Sibling Selector

Matches an element that is the next sibling of another

**h1+p { }**  
targets the first **<p>** element after any **<h1>** element (but not other **<p>** elements)

### General Sibling Selector

Matches an element that is a sibling of another, although it does not have to be the directly preceding element

**h1~p { }**  
tif you have two **<p>** elements that are siblings of an **<h1>** element, this rule would apply to both

```
/* type/element selector */
```

```
p {  
  color: blue;  
  font-size: 50vh;  
}
```

```
/* class attribute selector */
```

```
.myBlueText {  
  color: blue;  
}
```

```
/* id attribute selector */
```

```
#blue-par {  
  color: blue;  
}
```

```
/* BONUS: grouping  
selector */
```

```
p,  
.blue-text,  
#blue-par {  
  color: blue;  
}
```

selecting multiple elements:

```
h1, h2, h3 {
```

```
    color: red;  
    background-color: blue;  
    width: 500px;
```

```
}
```

```
p,  
li {
```

```
    background-color: red;  
    font-color: blue;
```

```
}
```



**HTML comments are written like this**

```
<!-- This is a comment -->
```

**CSS comments are written like this**

```
/* This is a comment */
```

```
{  
text-align:  
    left ;  
    right ;  
    center ;  
    justify ;  
}
```

**a: link {**

**a: visited {**

**: hover {** Applied when a user hovers over an element w/ a mouse. This changes the appearance of links and buttons when a user places their cursor over them. Does not work on mobile.

**: active {** Applied when an element is being activated by a user, like when a button is pressed or a link clicked. This added to UX. Applied when an element has focus. Any thing you can interact with.

**: focus {** Focus occurs when a browser discovers that you are ready to interact w/ an element. For example when yr cursor is in an input - that element is said to have focus.

**}**