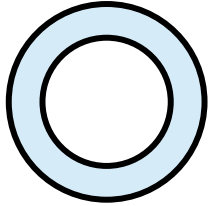
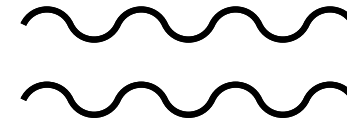


The background features a light gray gradient. On the left, there is a white rectangular box with a thin black border. To the right of the box, there are flowing, wavy lines in shades of blue and purple, creating a sense of motion. On the far left edge, there are some black zigzag lines.

CSS

CASCADING STYLE
SHEET

Why was CSS create?



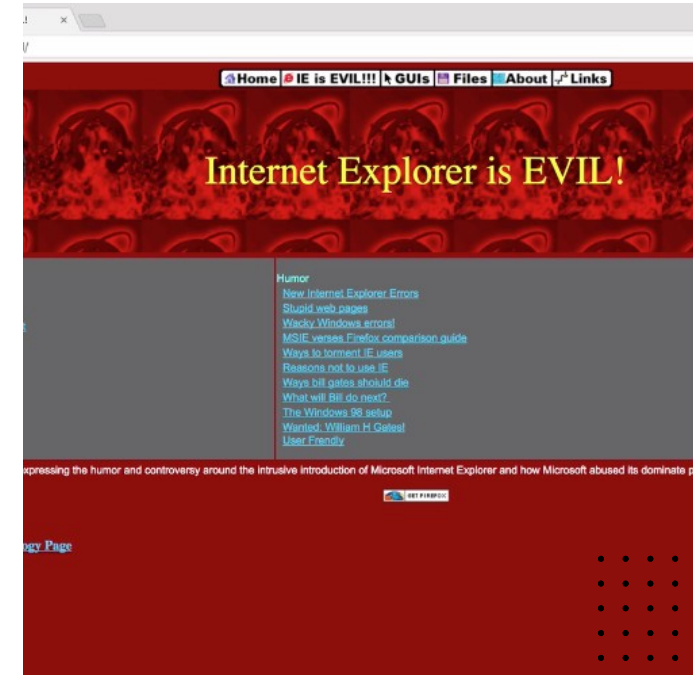
- For example:
 - The center tag <center></center>
 - <h1 bgcolor="#990000">>/h1>

```
<h1 bgcolor="#990000">
```

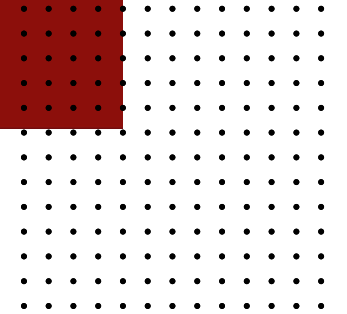
Get Firefox

```
<h1 bgcolor="#990000">
```

Get Firefox



The problem really comes along when you start messing around with the layout using just HTML because as we've seen in order to change our layout for our personal website so we can have text and images that exist side by side we have to use tables.





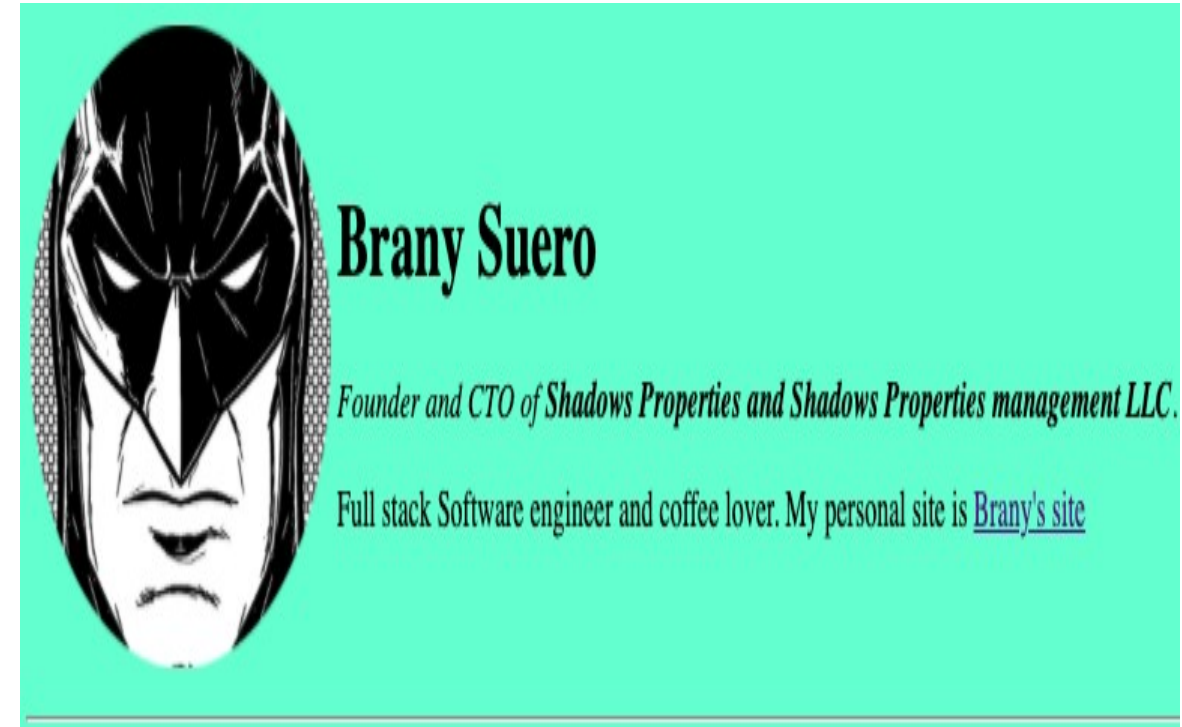
Inline CSS

allows you to apply a unique style to one HTML element at a time.

Example: Changing the background color of the website.

`<body style="background-color: aquamarine;">`

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>🐼 Brany's personal site</title>
8 </head>
9
10 <body style="background-color: ■aquamarine;">
11   <table>
12     <tr>
13       <td></td>
14       <td><h1>Brany Suero</h1>
15       <p><em>Founder and CTO of <Strong>Shadows Properties and Shadows Properties management LLC</Strong>.</em></p>
16
17       <p>Full stack Software engineer and coffee lover. My personal site is <a href="https://www.branysuero.com">Brany's
18         site</a></p></td>
19     </tr>
20   </table>
```



○ Block element

a **block-level element** is any **element** that starts a new line, like a paragraph, and uses the full width of the page or container.

Here are the block-level elements in HTML:

<code><address></code>	<code><article></code>	<code><aside></code>	<code><blockquote></code>	<code><canvas></code>	<code><dd></code>	<code><div></code>	<code><dl></code>	<code><dt></code>	<code><fieldset></code>
<code><figcaption></code>	<code><figure></code>	<code><footer></code>	<code><form></code>	<code><h1>-<h6></code>	<code><header></code>	<code><hr></code>	<code></code>	<code><main></code>	<code><nav></code>
<code><noscript></code>	<code></code>	<code><p></code>	<code><pre></code>	<code><section></code>	<code><table></code>	<code><tfoot></code>	<code></code>	<code><video></code>	

```
<h3><a href="hobbies.html">My Hobbies</a></h3>
<h3><a href="Contactme.html">Contact me</a></h3>
```

[My Hobbies](#)
[Contact me](#)

```
112
113 <h3 style="display:inline; padding:2em"><a href="hobbies.html">My Hobbies</a></h3>
114 <h3 style="display:inline"><a href="Contactme.html">Contact me</a></h3>
115
```

[My Hobbies](#)

[Contact me](#)



Internal CSS

is used to define a style for a single HTML page. An **internal CSS** is defined in the <head> section of an HTML page, within a <style> element.

The essential element is something you want to affect, then what you want to change about it and finally what is the value you are going to change it to.

```
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>♥ Brany's personal site</title>
<style>

  body {
    background-color: powderblue;
  }

  hr {
    background-color: white;
    border-style: none;
    height: 2px;
    width: 100px;
  }

  img {
    height: 20px;
  }

</style>
```



○ I would like to apply different style to my line

dotted



```
hr {  
  background-color: white;  
  border-style: dotted;  
  height: 2px;  
  width: 100px;  
}
```



Does it look like the example?

```
hr {  
  background-color: white;  
  border-style: dotted none;  
  width: 100px;  
}
```

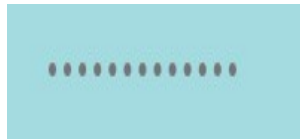


Removed the height

```
hr {  
  background-color: white;  
  border-style: dotted;  
  width: 100px;  
}
```



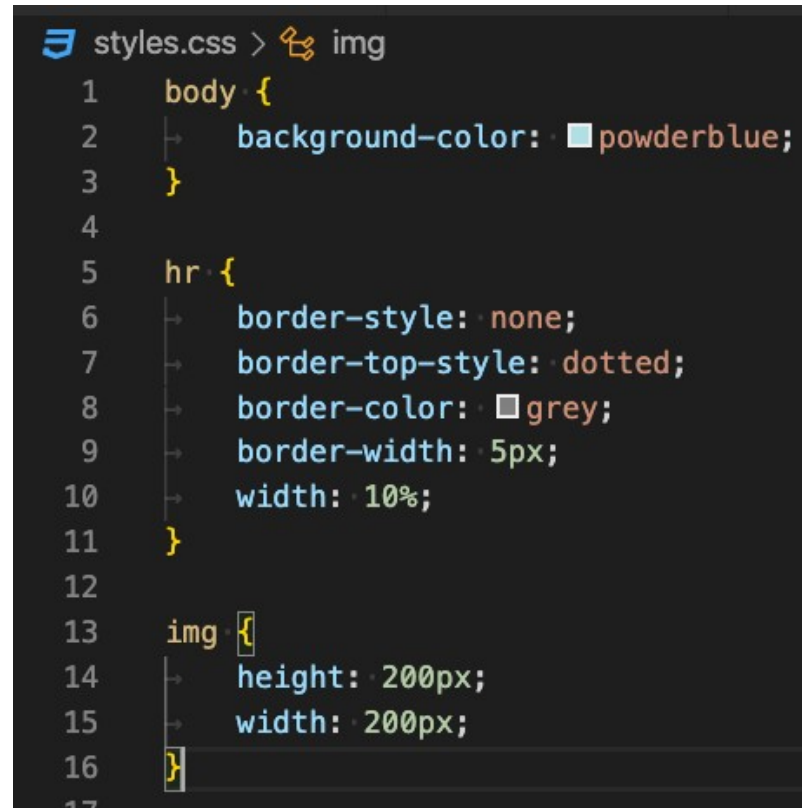
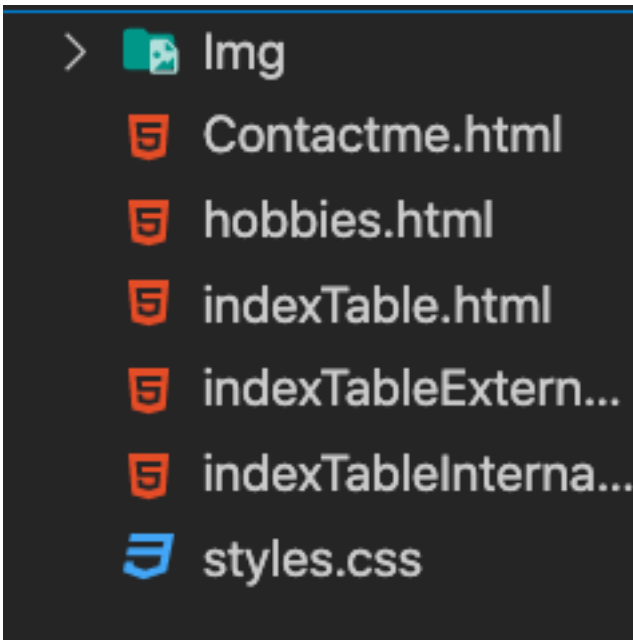
```
hr {  
  border-style: none;  
  border-top-style: dotted;  
  border-color: grey;  
  border-width: 5px;  
  width: 10%;  
}
```





External CSS

Gets done by Placing CSS in a separate file allows the web designer to completely differentiate between content (HTML) and design (CSS). External CSS is a file that contains only CSS code and is saved with a ".css" file extension.



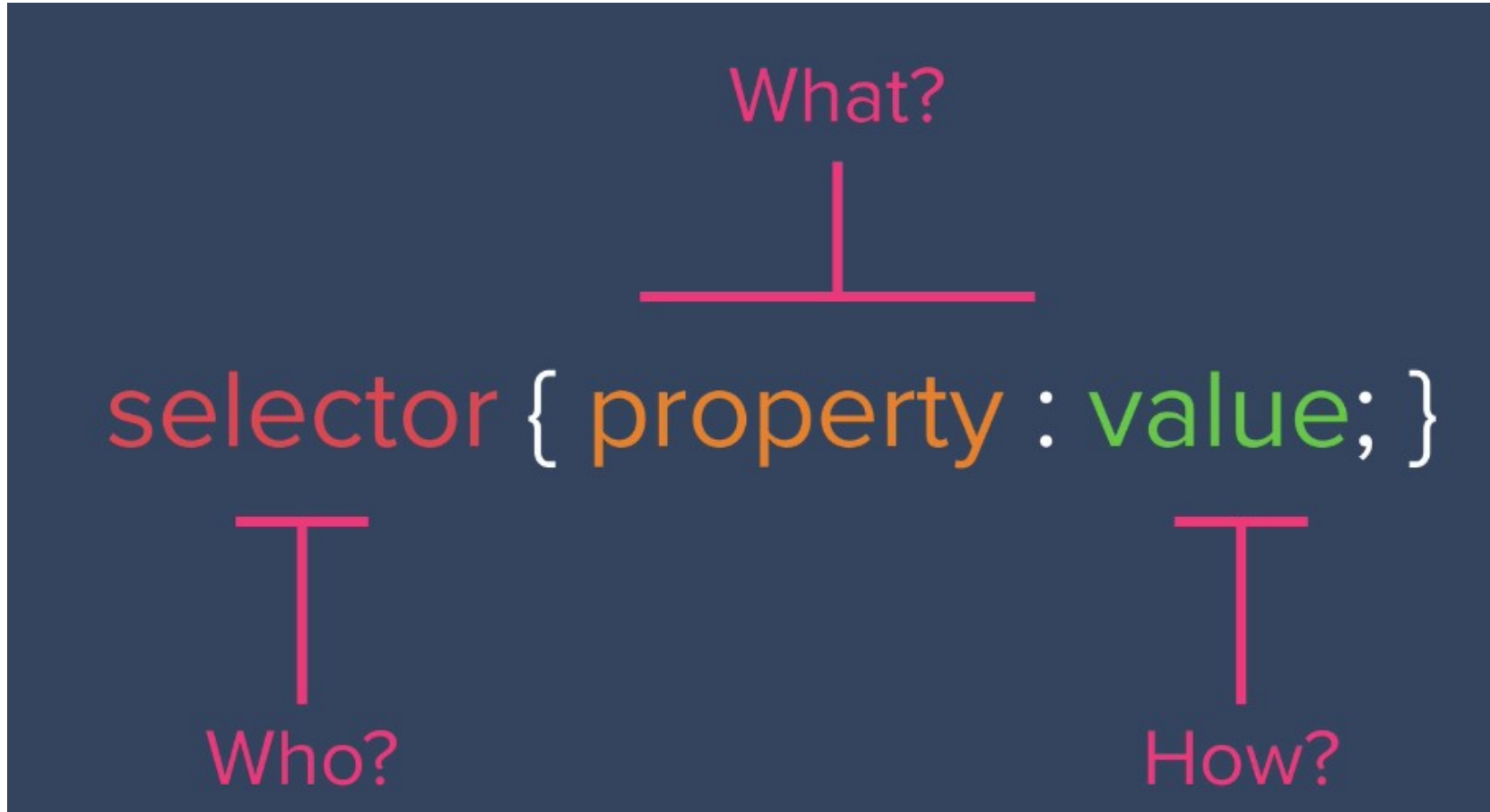
○ Challenge

- Select all the h3 and h1 from the page and change the color to #66BFBF.



○ The Anatomy of CSS Syntax

The grammar of the CSS Language

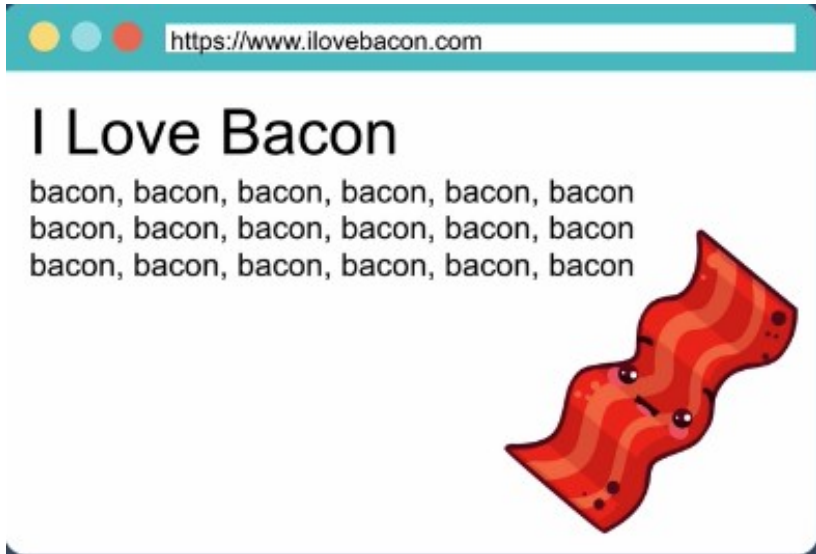


The who: who is that you want to modify in the html. Who style do you want to change?

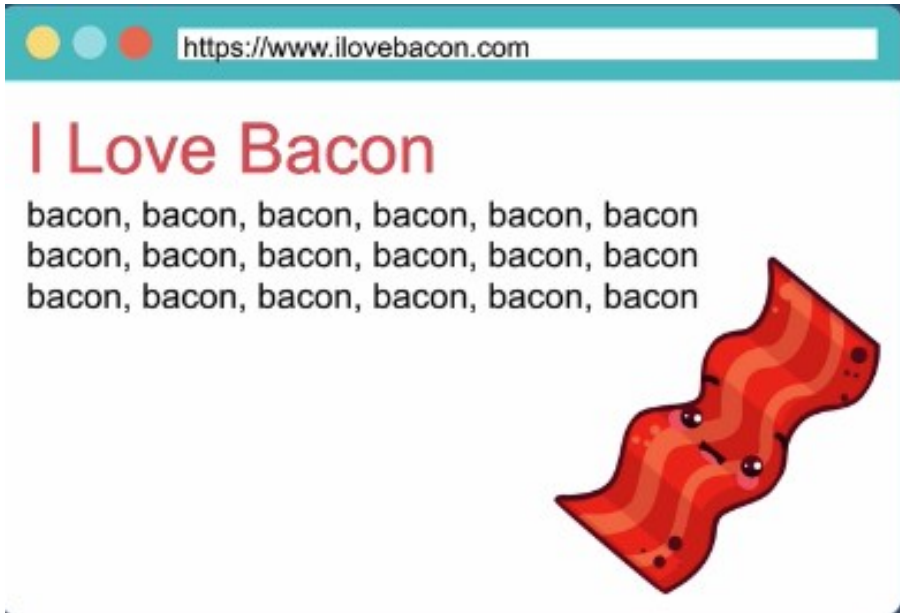
What: what about the h1 do you want to change? Text-color? Position?

How? How do you want to change? Ex background color to blue





This is the result by applying the CSS rules.



What?

└

h1 { color : red; }

T

Who?

T

How?

○ Selectors

Tag selector - is used to redefine existing HTML **tags**.

Class selector - selects elements with a specific **class** attribute.

ID selector - uses the **id** attribute of an HTML element to select a specific element.

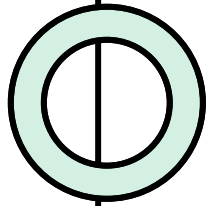
When should you use IDs or when should you use classes?

- You can only use 1 ID instance per tag in the page. You can have as many classes as you wish.
- You can use classes to select a group relate elements that should have the same style.
- You can use an ID to apply a specific style to a single element in the page.

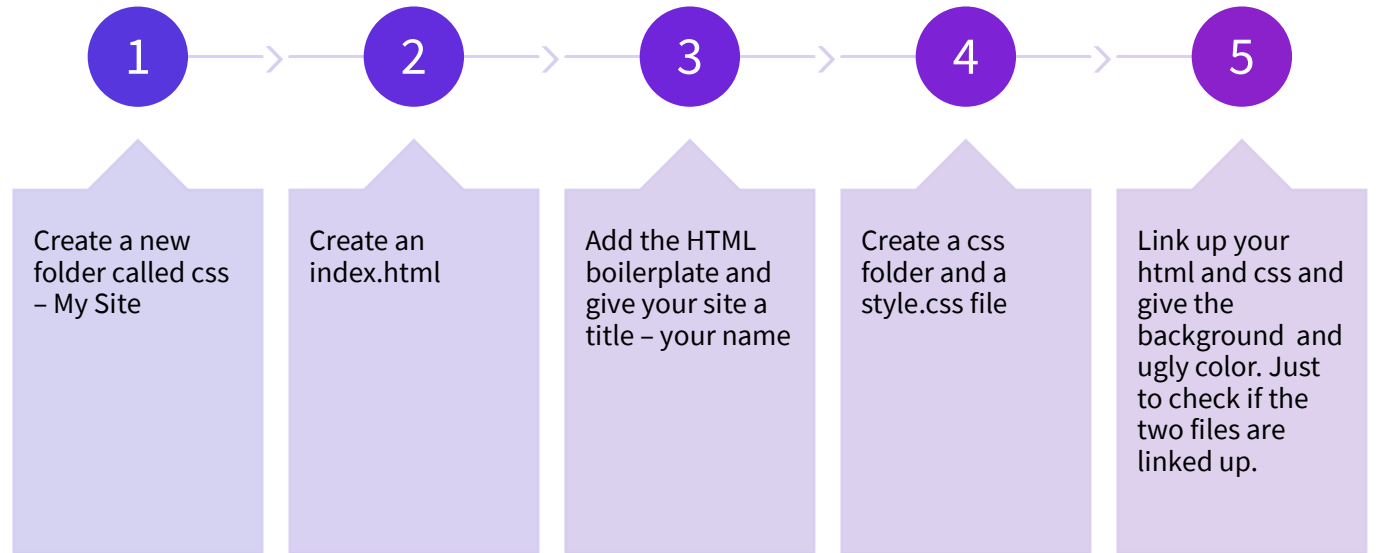
CSS Specificity : If there are two or more conflicting CSS rules that point to the same element, the browser follows some rules to determine which one is most specific and therefore wins out.

Sudo class: classes with a different state. They usually have a colon : in front.





Challenge





Favicons

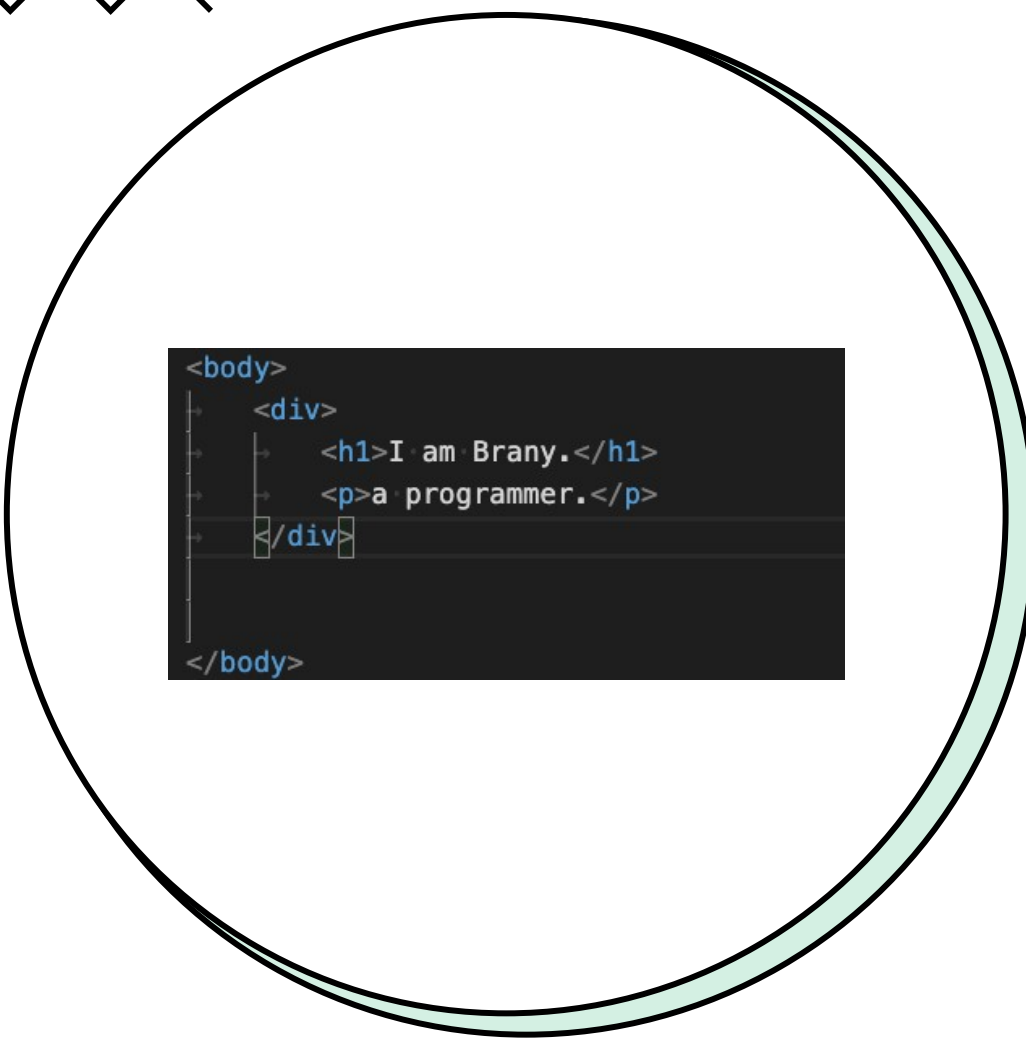
(favorite icon) is a small, iconic image that represents your website.

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
```





Html Divs element

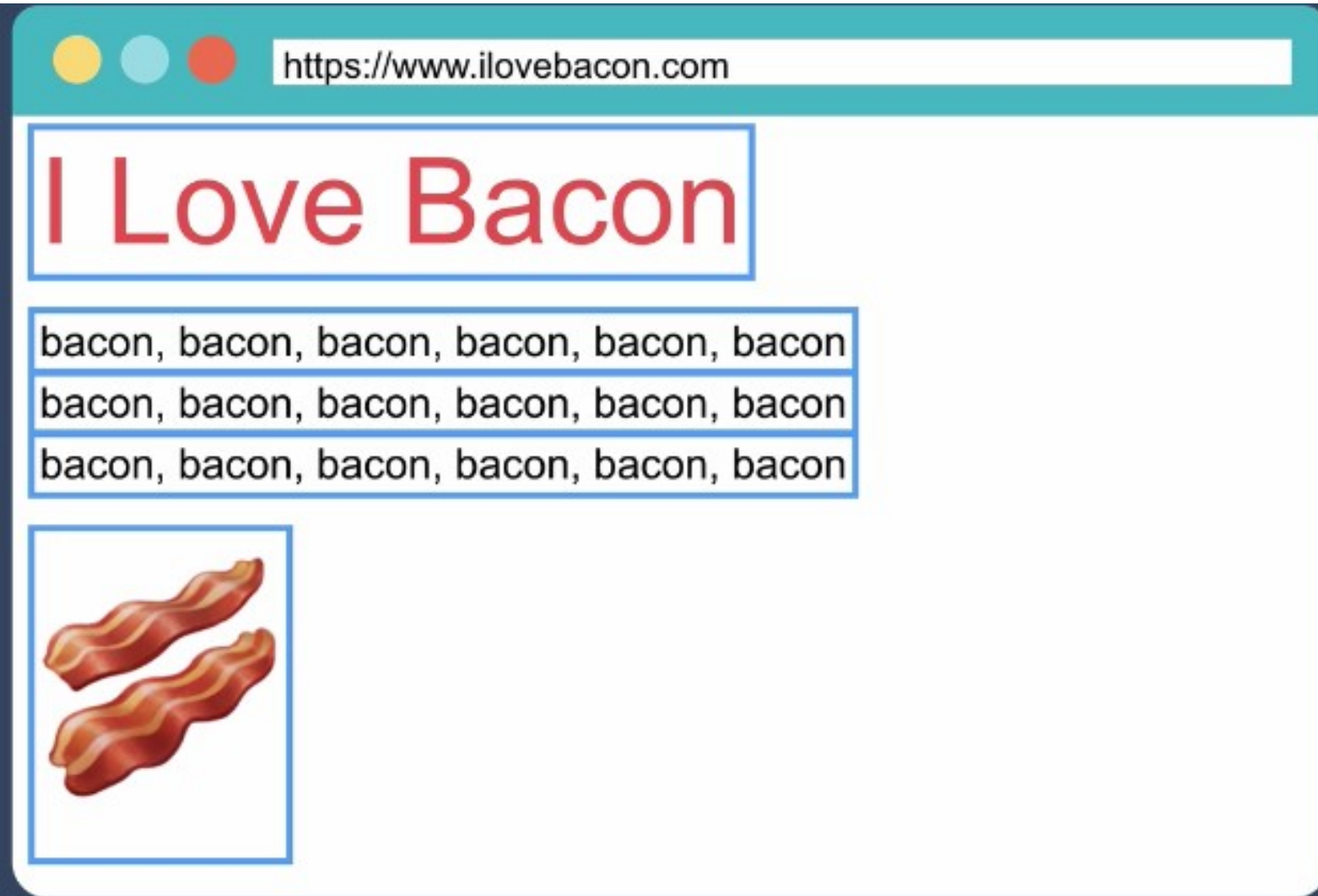


```
<body>
  <div>
    <h1>I am Brany.</h1>
    <p>a programmer.</p>
  </div>
</body>
```

- is a generic container with no particular semantic meaning. It allows to divide our content up on our website. So we can structure each div separately. In other words, allows to split up or divide your content into separate container or boxes so you can style the layout of each box separately.



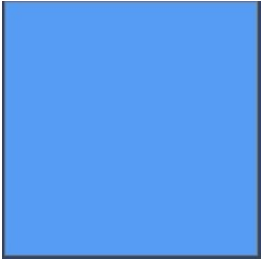
○ CSS Box Model



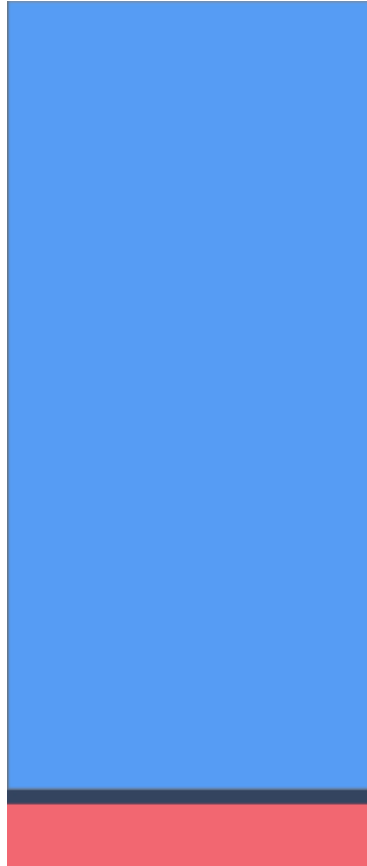
```
img { width : 100%; }
```

- Lets take a simple div and give a height of 300px.

```
{ height : 300px; }
```



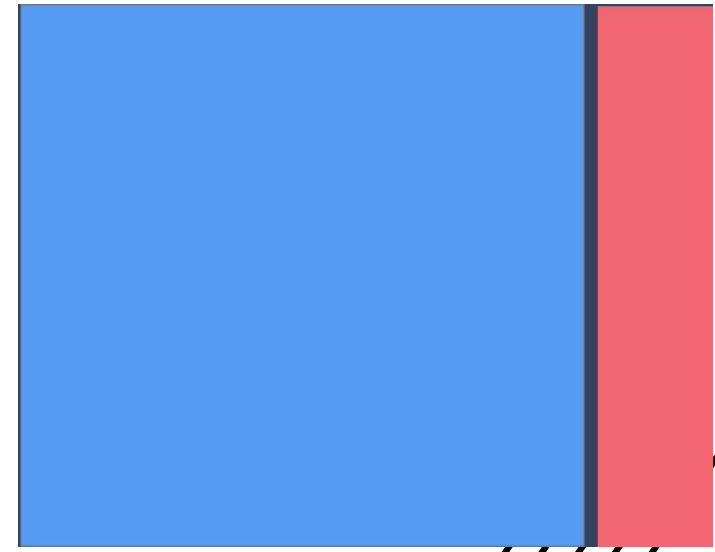
```
{ height : 600px; }
```



It is going to push down any other div or any other content in the website.

```
{ width : 600px; }
```

Same thing can happen with the width.



○ recap

You can specify this values as a static pixel or we can use percentages.

```
{ width : 600px; }
```

```
{ width : 100%; }
```

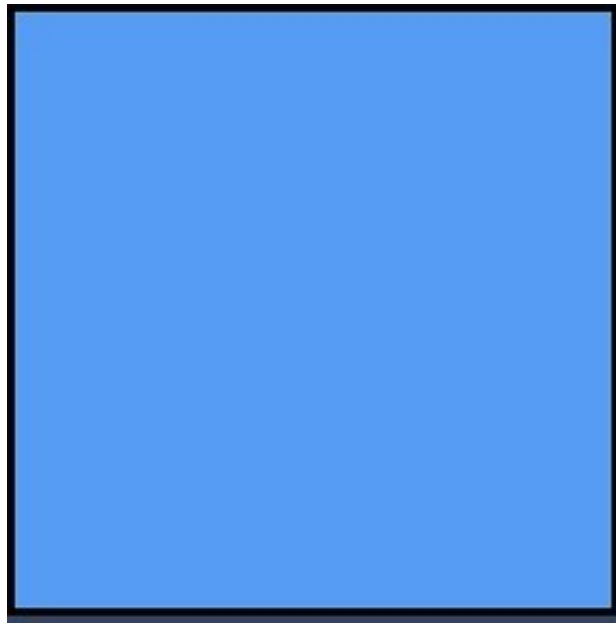
100 % of the view port – the view port is the screen that you are looking the web on





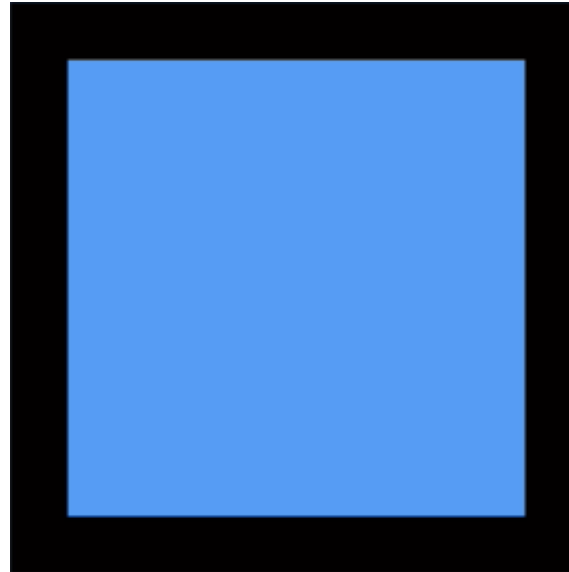
- In addition to the width and height. You can also specify the border.

```
{ border : solid; }
```

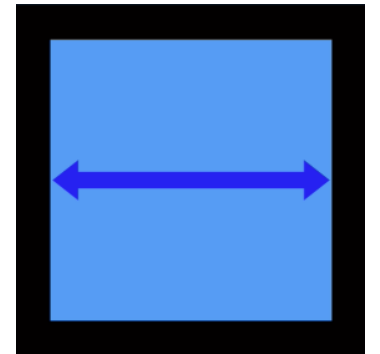


3px border

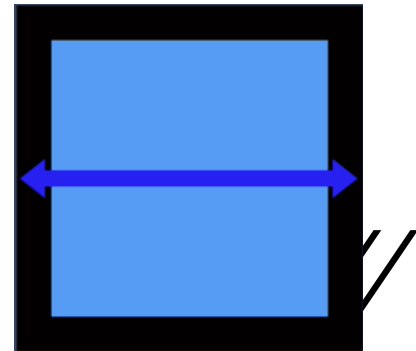
You can increase the border size.
Its 50px all around so now the
element is wider



300px width
the original
element



The entire
box is 400
px





- You can specify the border that you would like to add or remove. Ex:

```
{ border-top : 0px; }
```



Circle
shorthand:
you go around
in a circle. It
starts from the
top.

```
{ border-width : 0px 10px 20px 30px; }
```

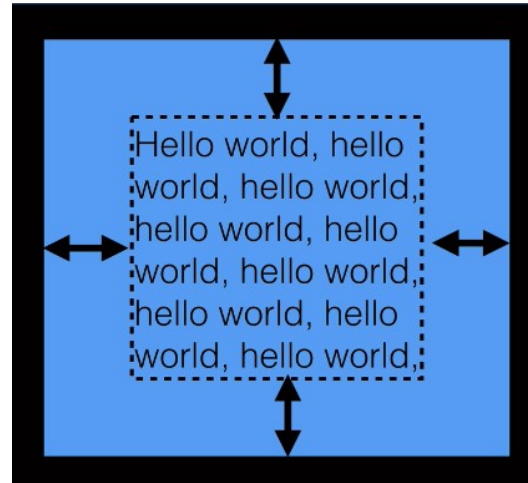


- Let's say we have some content in our div.



We do not like the way it looks. We might want a little bit of padding / space.

```
{ padding : 20px; }
```



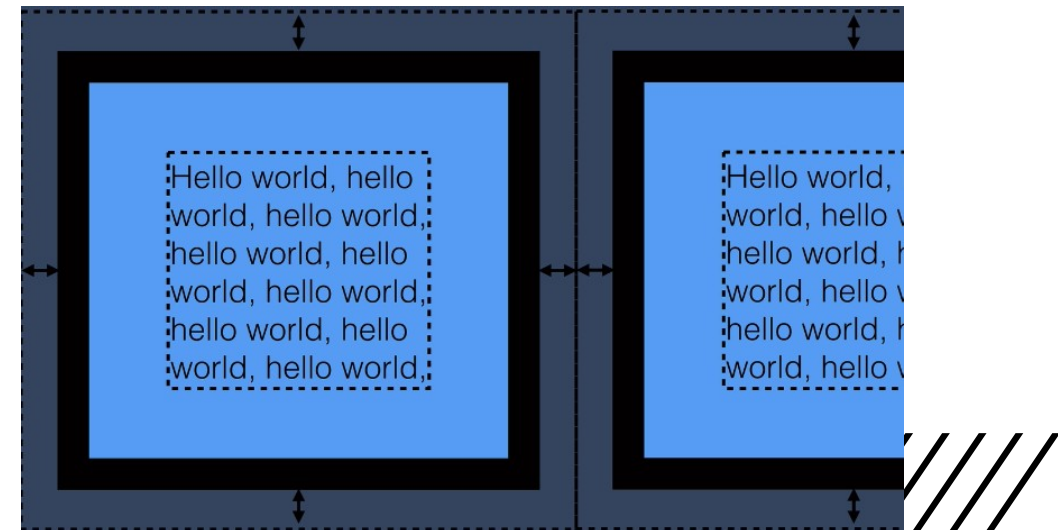
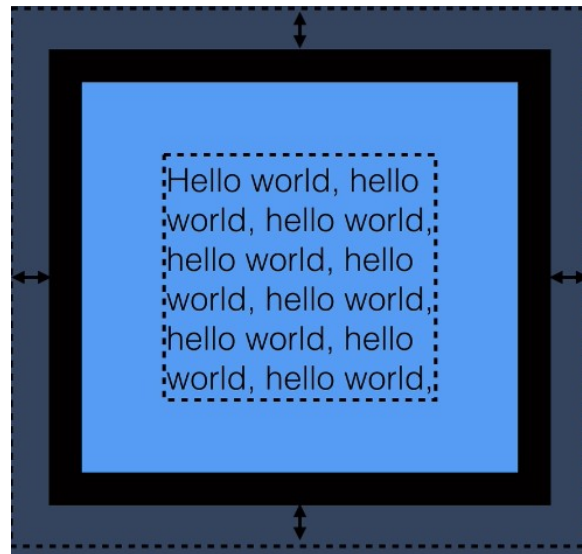
This says that there should be 20 pixels of space on all four edges between the text and the edge of the element.

Now our box is even bigger.

Final dimension: margin

Margin is a buffer zone if you will between the current element and others in the screen.

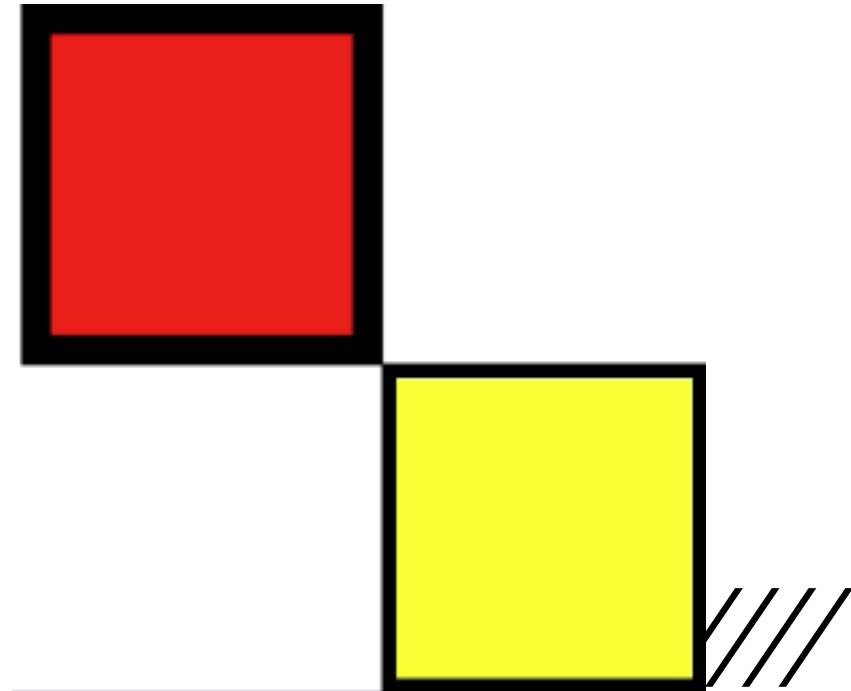
```
{ margin : 10px; }
```



○ Challenge

- Create three div with a class of top-container, middle-container and bottom-container.
- Give each div a height of 300px and width of 300px.
- Please give each of the div a different background color.
- 1st box: 10px border & 20px padding.
- 2nd box: 20px border.
- 3rd box: 10px border.

All boxes need to touch each other pixel perfect.



○ CSS Display property

Block: by default some element are block display. It takes up the whole width of the screen in the web page. It blocks out any element from the left or the right. Some common elements: h1, p, ul, ol, li, form, div. You can change the width, but it will not accept any element next to it anyway.

Hello

World

Inline: only take as much space as it needs to. Some common elements. Span, img and a. you can't change the width with inline element.

Inline-block: it acts like inline element, but you can change the width.

None: it gets rid of our element.

Programmer – how to style just part of the word. Use a tag called span. You can test this tag using text-decoration: underline;

Span is an inline.

Hello World

HTML

```
1 <p>Hello</p>
2 <p>World</p>
3 <span>Hello</span>
4 <span>World</span>
```

CSS

```
1 p {
2   background-color: red;
3 }
4
5 span {
6   background-color: blue;
7 }
```

```
p {
  background-color: red;
  width: 100px;
  display: inline;
}

span {
  display: block;
  background-color: blue;
  width: 100px;
}
```

CSS Static, relative, fixed and absolute positioning

Default rules how things are render on the screen.

- We have to realize that even without CSS, your HTML element already has predefined rules for how it should be display on your web page even if you do not have any CSS.

1. Content is everything: inline element only take up as much width and as much height as the content.

- block element even though it takes 100% of the width, the height it depends but the content of the element.



- 2. order comes from Code: the order of your element in your screen come from the html code.



```
<h1> </h1>
<p> </p>
<p> </p>
<p> </p>
<img>
```



```
<img>
<h1> </h1>
<p> </p>
<p> </p>
<p> </p>
```

The default layout order is determined by your code.

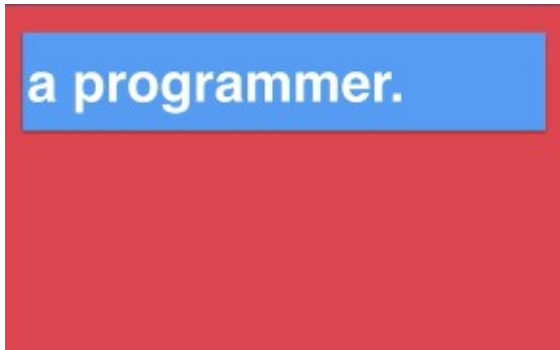


○ 3. Children sit on Parents:



div

```
<div>  
</div>
```



```
<div>  
<h1>a programmer</h1>  
</div>
```

The h1 is going to be on top of the div. it is more towards the viewer and away from the screen.

You can also set a css property which is the position property.





Static position

- Goes along with the html rules and keep to the default HTML flow. Ex: that's what you see when you do not have any css in the page. All elements are static by default.



○ Relative position

Position the element that we select relative to how it would have positioned had it been static.



```
img {  
  position: relative  
  left: 30px;  
}
```



Two important things to remember:

When you move an element with relative position, it does not affect the position of other elements.

We change the coordinate property
From the current image property. //

It push the image 30px space from the left of the previous position.

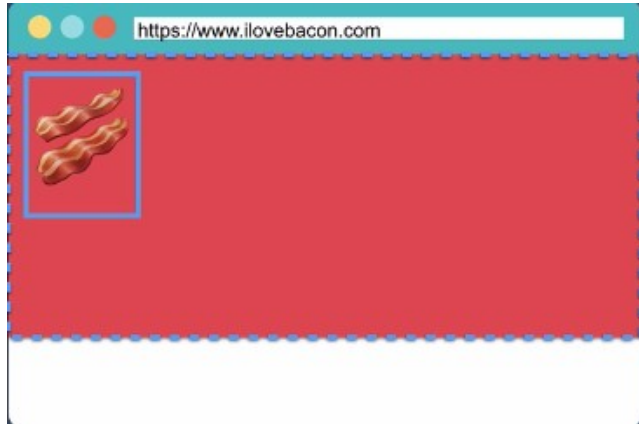
There are 4 coordinates properties:

1. Top
2. Bottom
3. Left
4. right

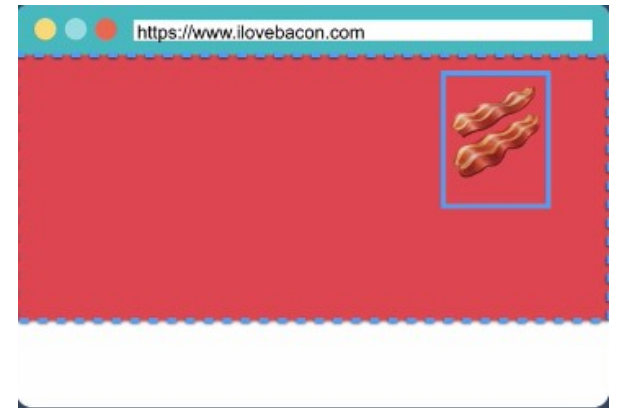


Absolute position

Position the element relative to its parent. Ex: the parent in the example is the red div.



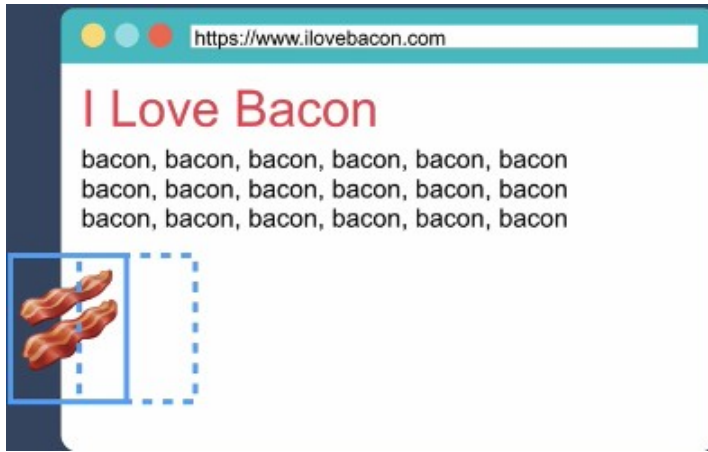
```
div {  
  position: relative  
}  
img {  
  position: absolute  
  right: 30px;  
}
```



It shift to the right of the screen.



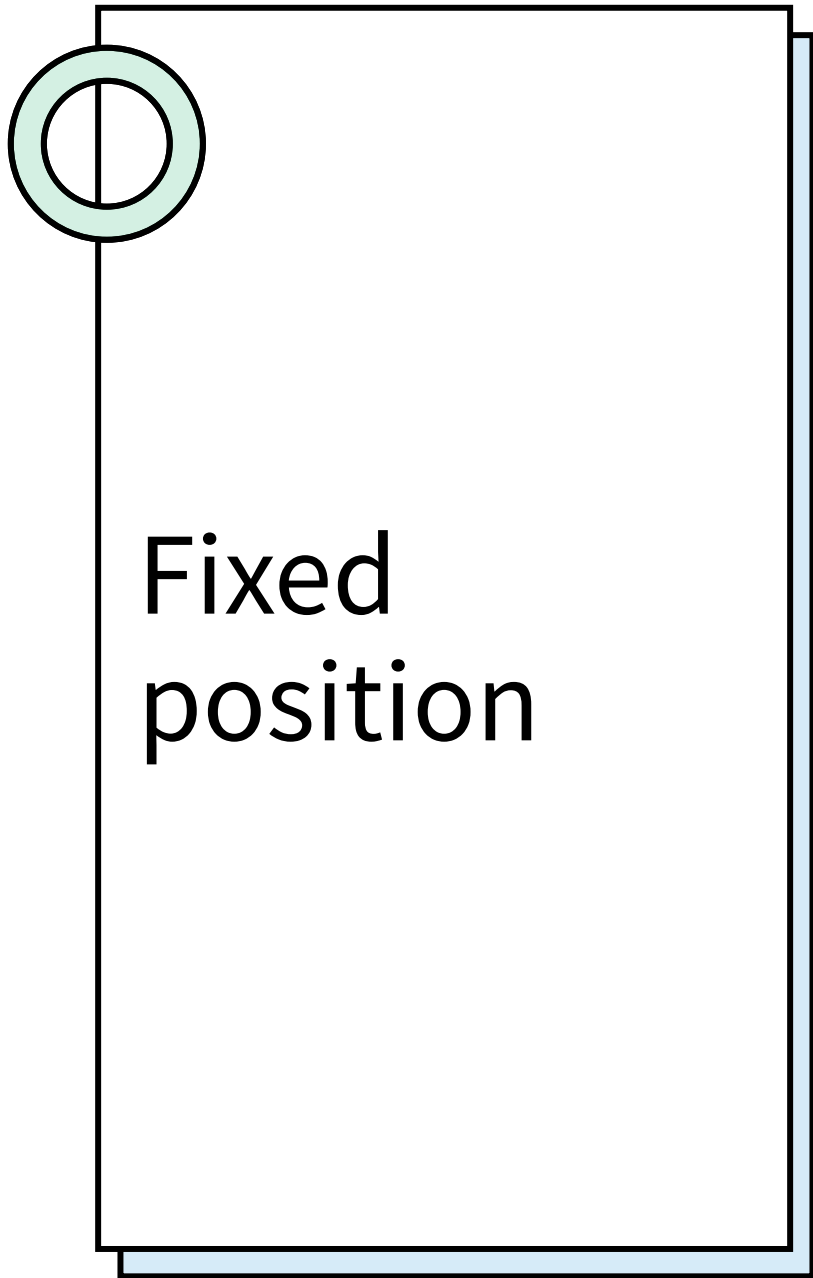
○ Relative vs Absolute



Relative positioning means that you are adding a margin relative to where the element should have been.

Absolute position means that you are adding a margin to its parent element.





Element will
stay in their
current
position.

It is useful for a
navbar or a
sidebar.

○ Font Styling

- Two big font family: sans-serif and serif.
 - Depending on the browser by default all the elements are serif family.
- How to know if you would like to use a different type a font.
 - Google font.

```
head>
<meta charset="utf-8">
<title>Angela Yu</title>
<link rel="stylesheet" href="css/styles.css">
<link rel="icon" href="favicon.ico">
<link href="https://fonts.googleapis.com/css?family=Merriweather|Montserrat|Sacramento" rel="stylesheet">
/head>
```

```
• body {
  margin: 0;
  text-align: center;
  font-family: verdana, sans-serif;
}
```

It set the font family to verdana but if the operating system does not have the font install, it will go to sans-serif by default.

```
• body {
  margin: 0;
  text-align: center;
  font-family: 'Merriweather', serif;
}
```





CSS Sizing

Font-size: 90px

Increase the size of the element. If the developer uses a %, it will make the size dynamic. If the user change the size of the browser, the size of the element will scale with it.

Font-size: 2em

1 em = 16px = 100% so if you specify 90 in pixels, it would be 90px but 5.625em

Why use one over the other one?

em would scale up and px does not. People with visual problem would have a better experience if the website is using em.

When you use em and % the value is inherited.

Font-size: 2rem

It would ignore the parent settings and keep its own.

