Topic/Title: What We'll Make - Stylised Personal Site What are Favicons?

Keywords/Questions:

favicon

favicon.cc

.ico

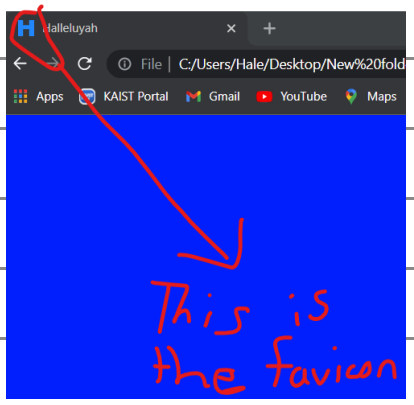
rel="icon"

href="facvicon.ico"

href="facvicon.ico?v=2"

Notes:

favicon:- the icon next to the tab title of your site. The icon thingy of your website head.



-have a file name ending of .ico instead of .png

-we can create our own favicon in favicon.cc website. Then we can download it for use.

-To add a favicon to our page, we need to download our favicon and place it in CSS - My Site and write the code below in the head part of our page:

- <link rel="icon" href="favicon.ico">

-we need to replace "stylesheet" with "icon" and modify the location of our page accordingly.

```
<head>
  <meta charset="utf-8">
  <title>Halleluyah</title>
  <link rel="stylesheet" href="css/styles.css">
  <link rel="icon" href="favicon.ico">
</head>
```

```
<link rel="favicon" href="favicon.ico?v=2">
```

-Try the above code if the left doesn't work. Add "?v=2" next to .ico

Summary:

favicon:- the icon thingy at the heading title.

-file name ends with .ico

-written as <link rel="icon" href="favicon.ico"> in the head part of our code.

-if an error is encountered we can write it as

-<link rel="icon" href="favicon.ico?v=2">

-we can make our own favicon in favicon.cc. We can build or paint our own favicon or import an image and edit it in favicon.cc

Topic/Title: HTML Divs

Keywords/Questions:

Ctrl+Pesticide

div

margin:0

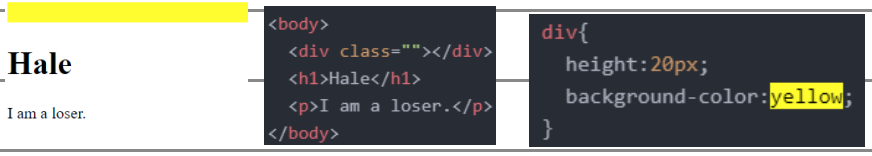
margin-top:0

Notes:

-Turning on "Pesticide for Chrome" and pressing Ctrl, and hovering over our website will tell us what each text we are hovering over is made from. A text will appear and tell us what element each text belongs to.

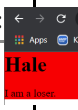
div: a content division element.

-It can be empty and have a height by the use of CSS for styling.



-It allows you to split up or divide your content into separate containers or boxes so that you can affect the layout of each box separately so that you can say that maybe these two things should be grouped together, and I'm going to structure it and style it separately from other content in my website.

-When using div, we notice that the margins at the top, left and right are white and empty. This is because there is a default margin for our body and h1 tags:



- To avoid these go to styles.ccs and set margin:0 in both

Summary:

-Hovering over our website while pressing Ctrl after turning on "Pesticide':

-will tell us the html element each part of our website is made of.

- ..... :

-It can be used to group contents and style them together.

-It can serve to style an empty area of a page in a box.

-we can remove white space while using div on our margins by using:

```
body{margin:0;}
h1{margin-top:0;}
```

Topic/Title: The Box Model of Website Styling

Keywords/Questions:

The box model

width:20%

width:10px

border:

border: solid;

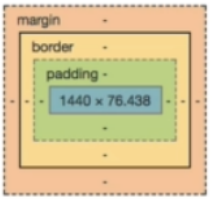
border:solid 10px;

border-width:

padding:

margin:

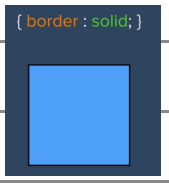
Notes:



-Every single HTML element on-screen is just treated as a box by our CSS.

- Height and width of this boxes can be specified mainly in two ways:

- {width:100%} -{height:30px}

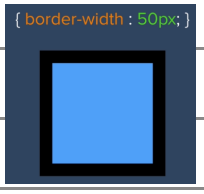


{border: solid;}

-creates by default a 3px wide border for our element.

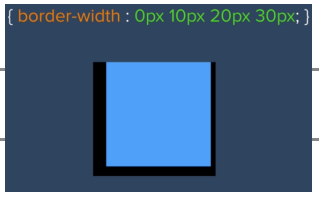
border-width:

-modifies the default border thickness.

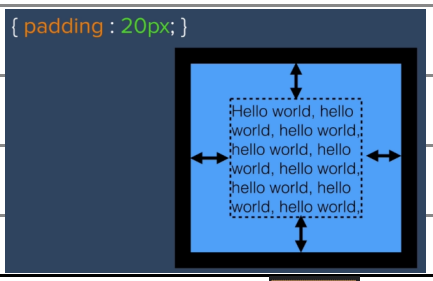


-We can also write it as `border:solid 50px;`

-we can specify the border thickness for each side. If we specify the border thickness for all sides, we kind of specify it starting from the top in a clockwise manner.



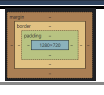
padding: -used to add a space between the border and the content inside it.



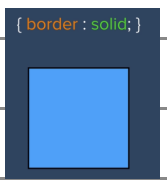
-It doesn't affect the background-color or the background-image. It just affects the content inside.

Summary:

The box model:- every HTML element is treated like a box.



height and width can be specified in px or in %.

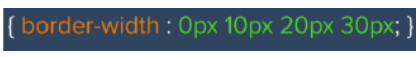


border :-specifies the border type. border:solid; has default thickness of 3px.

border-width: used to modify the thickness of the border.

```
border:solid;
border-width:50px;
```

- `border:solid 50px;` ==

-we can specify it as 

-1st no. for top and goes in a clockwise manner.

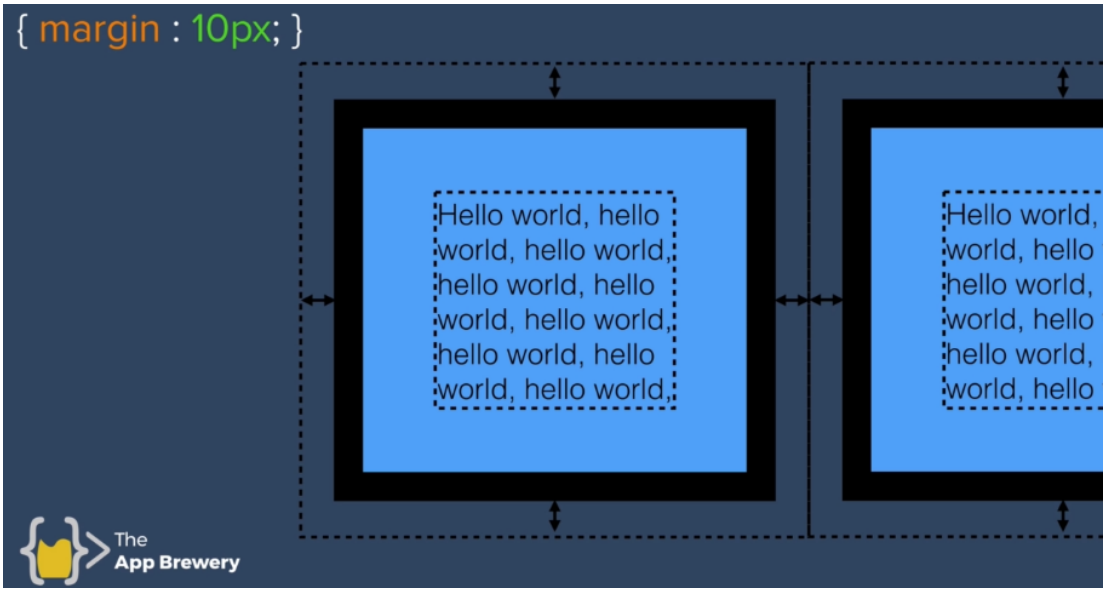
padding:- add space between border and content within

margin:- specify space between border and other element borders

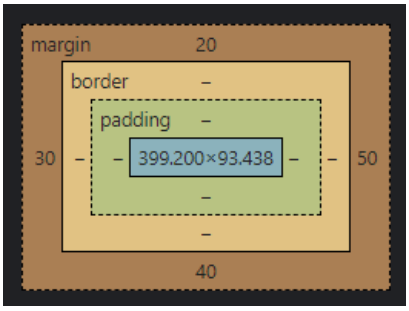
More Notes

-margin:

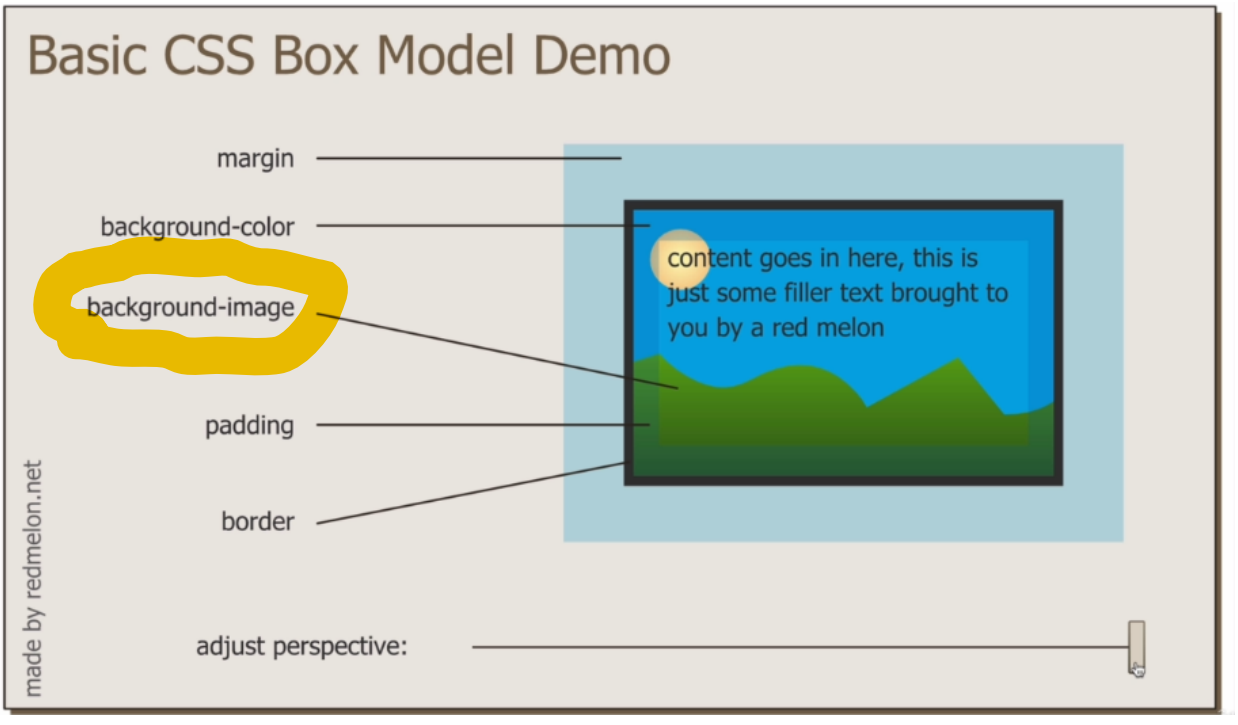
-used to specify spacing between the current element and any other elements on the screen. It pushes them away and makes them not stick to our element. We can use terms like margin, margin-top, ...

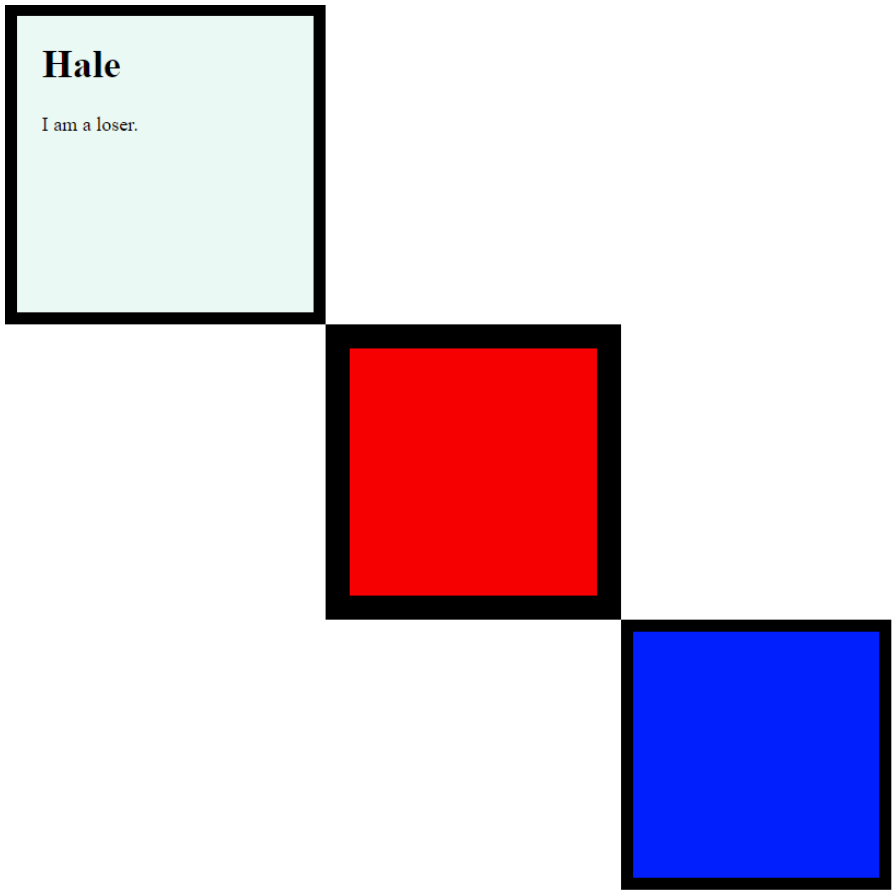


-The box model in the Styles section of Developer tools.



-We can edit the numbers for margin, border and padding by writing whatever number we want on the _ spaces inside the box. It is temporary and help us visualize what it would look like if we edit it.






```
Elements  Console  »
▼<body>
  ▼<div class="top-container" style="
      height: 200px;
      width: 200px;
      border: solid 10px;
      padding: 20px;
  ">
      <h1>Hale</h1>
      <p>I am a loser.</p>
    </div>
    <div class="middle-container" style="
        border: solid 20px;
        margin-left:260px;
    "> </div>
    <div class="bottom-container" style="
        border:solid 10px;
        margin-left:500px;
    "> </div>
  </body>
```

Topic/Title: CSS Display Property

Keywords/Questions:

display:block;

display:inline;

display:inline-block;

display:none;

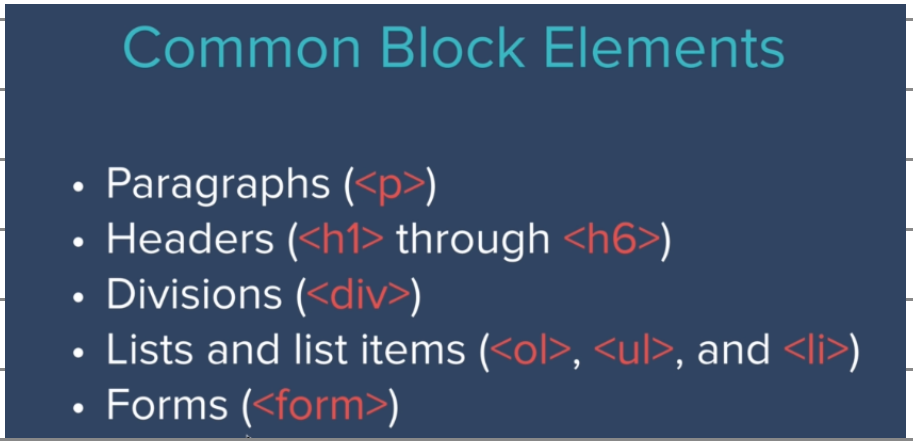
visibility:hidden;

text-decoration:underline

Notes:

Block elements:- are elements that take up the whole width of the screen.

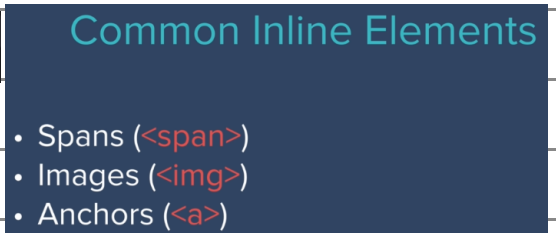
-They are coded in CSS by default as display:block;



text-decoration:underline; :- used to underline text.

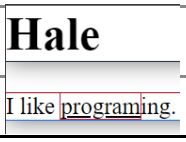
Inline elements:-take space only as much as they need. They don't block other elements from occurring in the same line. They are coded in CSS by default as display:inline;


```
span {
  background-color: blue;
}
```



...: commonly used to partition a text within a paragraph to style it without affecting the text's position.

For example here we highlighted program




```
<div class="top-container">
  <h1>Hale</h1>
  <p>I like <span class="pro">program</span>ing.</p>
</div>
```

```
.pro{
  text-decoration:underline;
}
```

Summary:

display:block; :-have width independent of content -can't have another element on the same line -can set width

display:inline; :-have width dependent on content -can have another element on the same line -can't set width

-...: an inline text element.

display:inline-block; :-have width dependent on content -can have another element on the same line

display: none; :-makes the element disappear. -it will be like it has never existed. -the object will not occupy space.

visibility:hidden; makes the element invisible to the eye. But, we can see the space it occupies.

text-decoration:underline; :-used to underline text.

More Notes

-Another difference between block and inline elements is that ***we can specify the width for a block element,*** while ***we can't for an inline element.*** But, ***after specifying a width for a block element we can't add any other element on that same line.***

-We can change an inline element to a block element or vice versa by writing display:block; or display:inline; in styles.css. But, the above property still remains.

display:inline-block;

- -has best of both worlds: inline and block.
- -you are allowed to change the width(like block)
- -you are allowed to have other elements on the same line. (like inline)

display:none;

- -makes it like the element never existed in our website.
- -the element won't occupy any space(width) and won't be visible.
- -It can be used to hide elements in our website.

visibility:hidden;

- -makes the element disappear, but it occupies space.
- -makes that element disappear but it keeps its original position, and all the other elements still flow around it as if it's still there. It just can't be seen anymore.

Topic/Title: CSS Static and Relative Positioning

Keywords/Questions:

3 HTML Display rules

Static Positioning

Relative positioning

postion:relative

```
img {position:relative;
    left:50px;
}
```

Coordinates

Notes:

HTML element Rules of Display:

1.Content is everything

-the size of the content determines the size of the box. For

-an inline element: both height and width are based on it.

-a block element: the height is based on it.

2. Order comes from your code.

For example, if we change the position of the image tag, the location of the

image on our browser will change.

Summary: The 3 HTML rules of display:

1. Content is everything.

2. Order comes from your code.

3. Children sit on parents.

Positioning: -Static -Relative -Absolute -Fixed

Static positioning: original positioning set by default

Relative positioning: positioning relative to the default margins.

Coordinates:-left -top -right -bottom

More Notes

HTML element Rules of Display continued:

3. Children Sit On Parents.

-the first parent here is div then h1 then span. So, the span lies over the h1 tag and the h1 tag lies over the div tag.

CSS Positioning:

Static Positioning:

-All HTML elements are static in their position by default, and static just means go along with the HTML rules and keep to the default HTML flow, and that is what we see when we just have HTML without any CSS, or if we don't change this position property at all.

Relative Positioning:

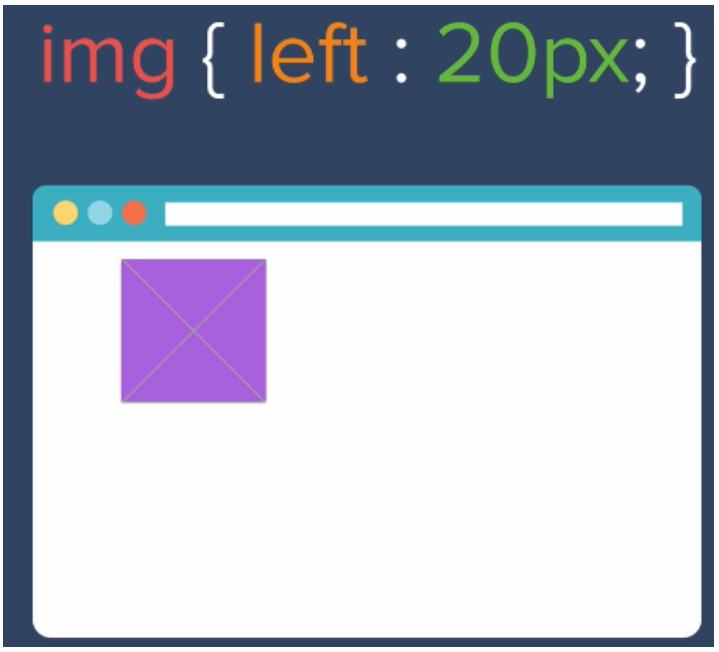
-this allows us to position the element that we select relative to how it would have been positioned had it been static. Don't forget to write position:relative

For example now we're saying give that image 30 pixels of space between its left edge and where the left edge used to be or give it a 30 pixels margin from the previous left edge of the img element.

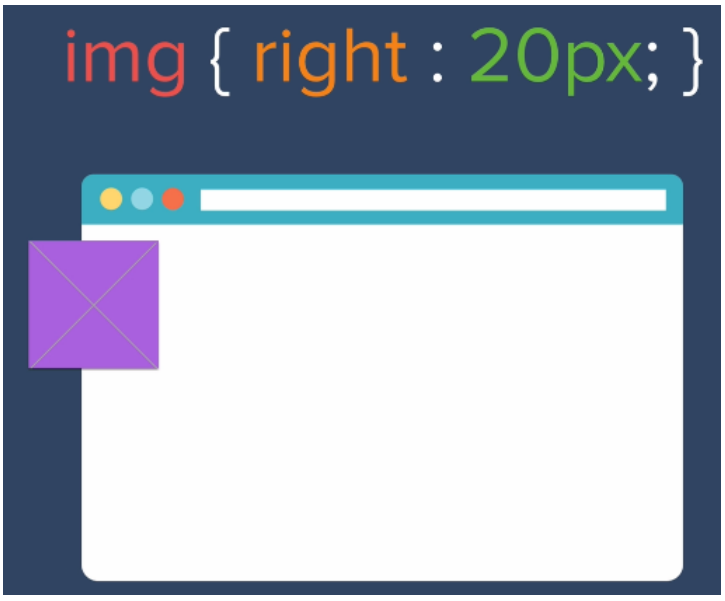
-We can set values for them in-order to determine how we wanna move our elements.

-This means it'll get moved down by 20 pixels.

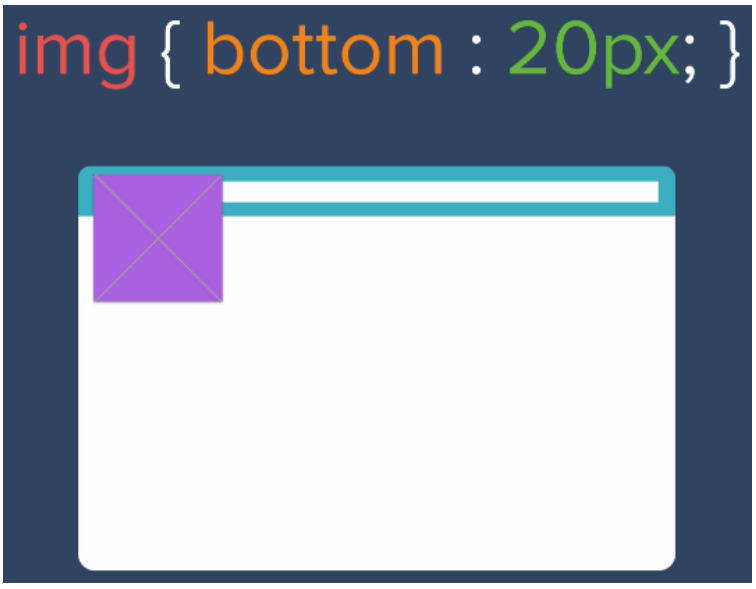
Relative Positioning Continued:



–it will get pushed to the right 20 pixels



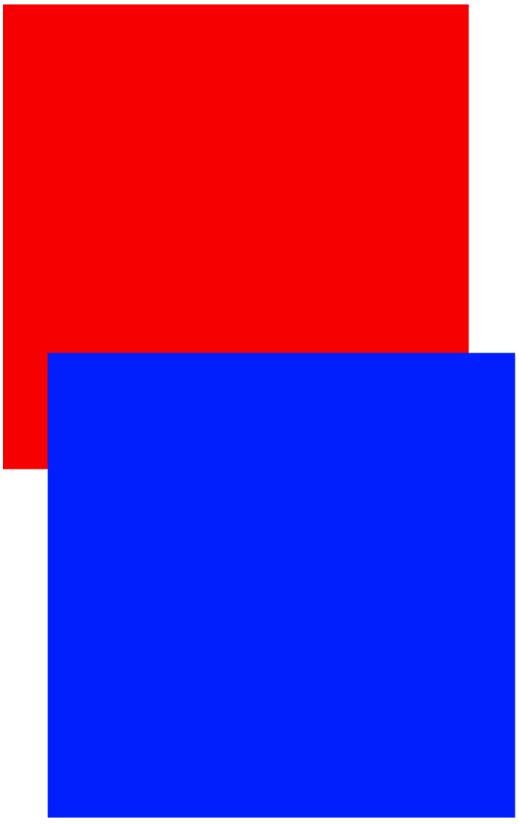
–it will get pushed to the left 20 pixels



–it will get pushed to the top 20 pixels

img {top:50px;}

—this means that we're taking the top of where that image used to be and we're giving it a 50 pixel margin from the top of our current image.



```
.middle-container{
  height: 200px;
  width: 200px;
  background-color: red;
}
.bottom-container{
  height: 200px;
  width: 200px;
  background-color: blue;
  position:relative;
  left:20px;
  bottom:50px;
}
```

—it just goes over the other box when we change its position.

HTML

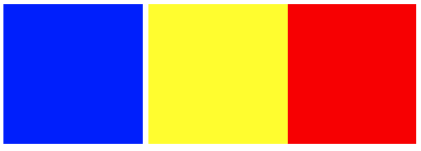
```
<body>
  <div class="red">
  </div>

  <div class="blue">
  </div>

  <div class="yellow">
  </div>

</body>
```

```
.red{
  height:100px;
  width:100px;
  background-color:red;
  display:inline-block;
  position:relative;
  left:200px;
}
.blue{
  height:100px;
  width:100px;
  background-color:blue;
  display:inline-block;
  position:relative;
  right:100px;
}
.yellow{
  height:100px;
  width:100px;
  background-color:yellow;
  display:inline-block;
  position:relative;
  right:100px;
}
```



—the reason why the spaces between our squares are inconsistent is because by making it an inline—block there's actually a little space that gets added in by the browser.

Topic/Title: Absolute Positioning

Keywords/Questions:

position:absolute;

position:relative;

.container{position:relative}

position:fixed;

Notes:

Relative Positioning

-means that you're adding a margin relative to where the element should have been.

Absolute Positioning:

-It is positioning relative to the parent element.

-It is usually relative to the border of the body.

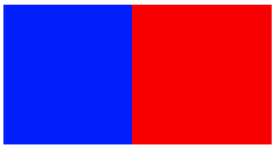
-When we use absolute positioning we are taking the object out of the natural flow of the document. It acts as if we have deleted the element. Other elements consider it as if it wasn't there.

HTML

```
<body>
  <div class="red"></div>
  <div class="blue"></div>
</body>
```

CSS

```
.red{
  height:100px;
  width:100px;
  background-color:red;
  position:absolute;
  left:100px;
}
.blue{
  height:100px;
  width:100px;
  background-color:blue;
```



Summary:

Positioning:

-static:- default positioning of our page.

-relative:-adding margin relative to where the element should have been.

-absolute:-adding a margin relative to the border of the parent element

-Combining absolute and relative:-we use a relative parent containing an absolute element. We position the element relative to the absolute parent.

-fixed: we create an element unaffected by scrolling.

More Notes

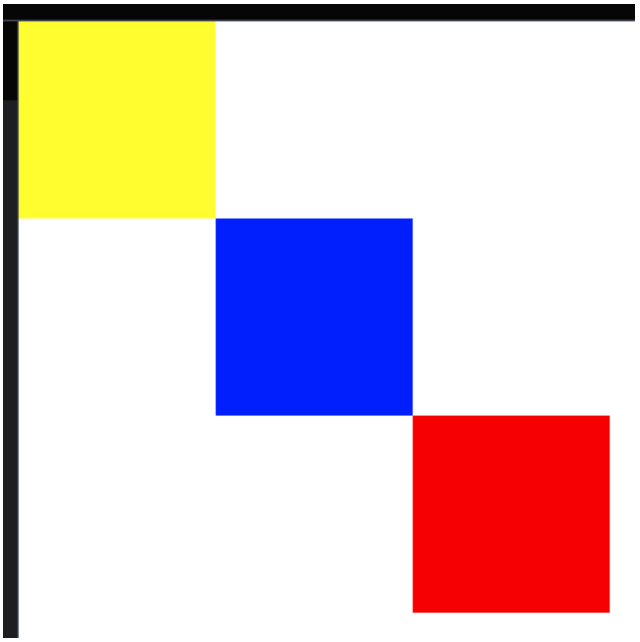
Absolute Positioning continued:

HTML

```
<body>
  <div class="red"></div>
  <div class="blue"></div>
  <div class="yellow"></div>
</body>
```

CSS

```
body{margin:0;}
.red{
  height:100px;width:100px;
  background-color:red;
  position:absolute;
  top:200px;
  left:200px;
}
.blue{
  height:100px;width:100px;
  background-color:blue;
  position:absolute;
  left:100px;
  top:100px;
}
.yellow{
  height:100px;width:100px;
  background-color:yellow;
  position:absolute;
}
```

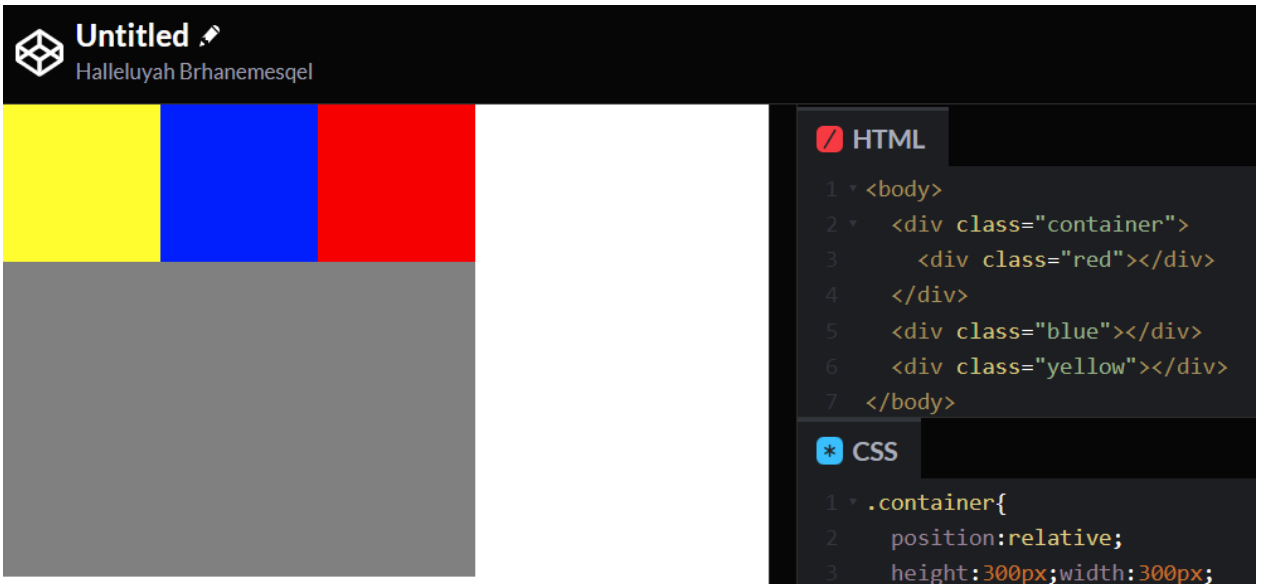


-Setting the margin of the body to 0 is important to avoid problems created by the margin or we can do what we did at the bottom for the yellow.

```
.red{
  height:100px;width:100px;
  background-color:red;
  position:absolute;
  top:200px;
  left:200px;
}
.blue{
  height:100px;width:100px;
  background-color:blue;
  position:absolute;
  left:100px;
  top:100px;
}
.yellow{
  height:100px;width:100px;
  background-color:yellow;
  position:absolute;
  left:0px;
  right:0px;
  top:0px;
}
```

Combining relative and absolute positioning:

– we can use containers to fine tune the position of our elements on screen by using a combination of absolute and relative positioning



HTML

```html
<body>
  <div class="container">
    <div class="red"></div>
  </div>
  <div class="blue"></div>
  <div class="yellow"></div>
</body>
```

CSS

```css
.container{
  position:relative;
  height:300px;width:300px;
  background-color:grey;
}
body{margin:0;}
.red{
  height:100px;width:100px;
  background-color:red;
  position:absolute;
  right:0px;
}
.blue{
  height:100px;width:100px;
  background-color:blue;
  position:absolute;
  left:100px;
  top:0px;
}
.yellow{
  height:100px;width:100px;
  background-color:yellow;
  position:absolute;
  left:0px;
  right:0px;
  top:0px;
}
```

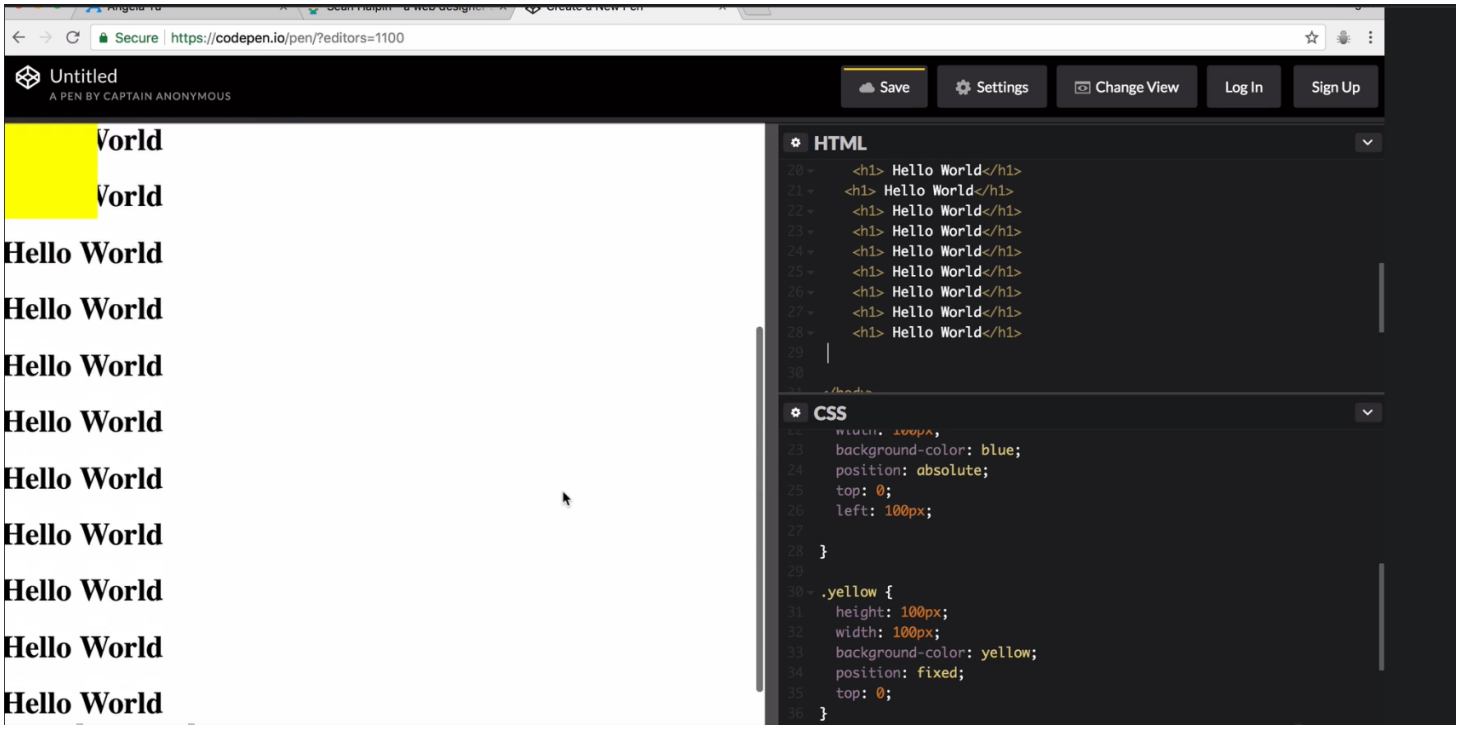
Fixed Positioning:

−It enables us to create an element unaffected by scrolling.

−This is really useful if you have a navigation bar that you want to be fixed so that it follows the user as they scroll through your web site or sometimes you might have a side bar that you want to stay fixed.

−For example if I change the above yellow's position from absolute to fixed and I say that it is fixed to the top then if I scroll through the web page it will stay in its current position.

−Written as position:fixed;



Topic/Title: The Dark Art of Centering Elements with CSS

Keywords/Questions:

text-align:center;

```
body{
  margin:0;
  text-align:center;
}

h1{
  width:10%;
  margin:0 auto;
}
```

margin: 0 auto;

Notes:

rcvr+_jge l8 ac l rcp9

+sqcb s l bcp AQQ g l r f c `mbw r_e mp _ n_pc l r am l r_g l cp bgt r_e,

c,e, `mbwyrcvr+_jge l8ac l rcp9{

+works well when their is no width specified or when it is an inline element. But if it is a block element and ***if there is a width specified***, then we should set the margin as margin:0 auto;

```
body{
  margin:0;
  text-align:center;
}

h1{
  width:10%;
  margin:0 auto;
}
```

Values of 0 shouldn't have units specified.

```
margin: 5%;                 /* All sides: 5% margin */

margin: 10px;               /* All sides: 10px margin */

margin: 1.6em 20px;         /* top and bottom: 1.6em margin */
                            /* left and right: 20px margin  */

margin: 10px 3% -1em;       /* top:            10px margin */
                            /* left and right: 3% margin   */
                            /* bottom:         -1em margin */

margin: 10px 3px 30px 5px;  /* top:    10px margin */
                            /* right:  3px margin  */
                            /* bottom: 30px margin */
                            /* left:   5px margin  */

margin: 2em auto;           /* top and bottom: 2em margin   */
                            /* Box is horizontally centered */

margin: auto;               /* top and bottom: 0 margin     */
                            /* Box is horizontally centered */
```

Summary: text-align:center;

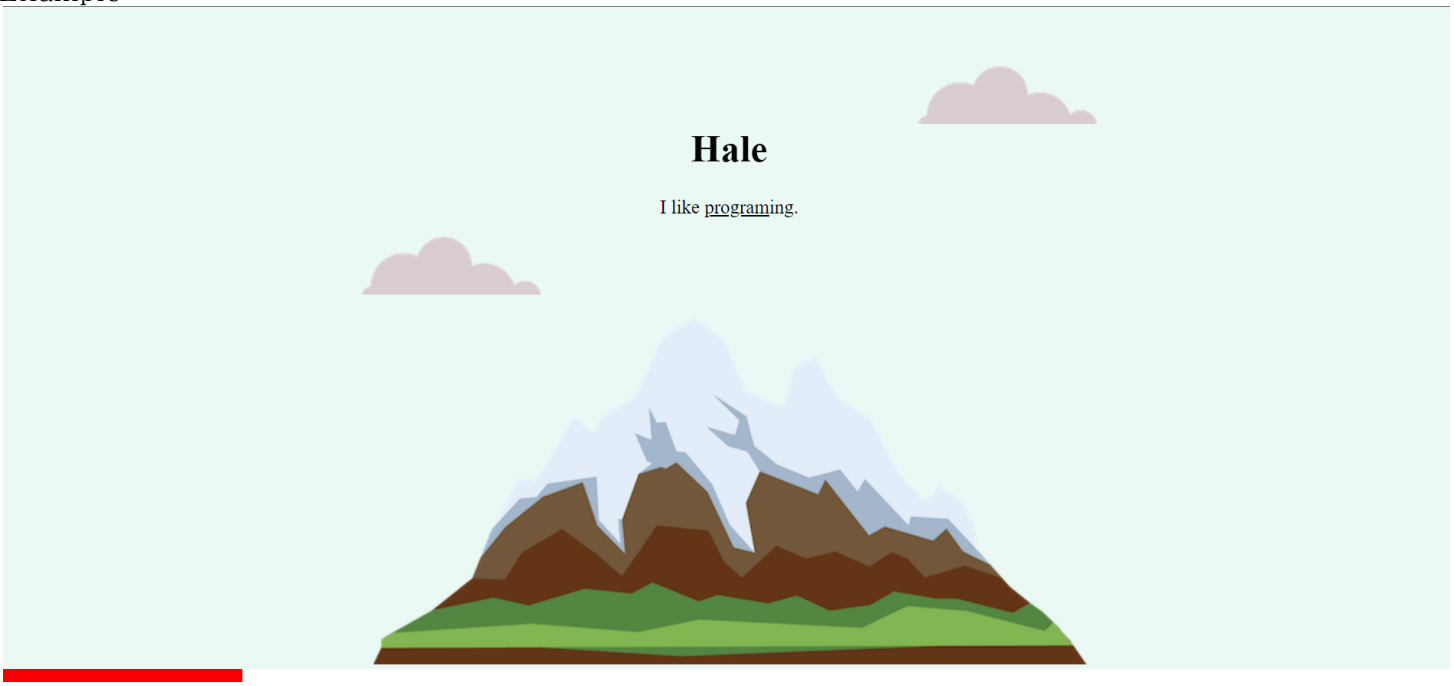
-used to center text, places it in the center and is written inside the parent element.

-If there is a width specified for an element, we need to use margin:0 auto;

```
body{
  margin:0;
  text-align:center;
}

h1{
  width:10%;
  margin:0 auto;
}
```

Example



index.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Halleluyah</title>
    <link rel="stylesheet" href="css/styles.css">
    <link rel="icon" href="favicon.ico">
  </head>
  <body>
    <div class="top-container">
      <img class="top-cloud" src="images/cloud.png" alt="cloud-img">
      <h1>Hale</h1>
      <p>I like <span class="pro">program</span>ing.</p>
      <img class="bottom-cloud" src="images/cloud.png" alt="cloud-img">
      <img src="images/mountain.png" alt="mountain-img">
    </div>
    <div class="middle-container">

    </div>
    <div class="bottom-container">

    </div>
  </body>
</html>
```

styles.css

```
body{
  margin:0;
  text-align:center;
}
h1{
  margin-top:0;
}
.top-container{
  background-color:#EAF9F5;
  position:relative;
  padding-top:100px;
}
.middle-container{
  height: 200px;
  width: 200px;
  background-color: red;
}
.bottom-container{
  height: 200px;
  width: 200px;
  background-color: blue;
  position:relative;
  left:20px;
  bottom:50px;
}
.pro{
  text-decoration:underline;
}
.top-cloud{
  position:absolute;
  right:300px;
  top:50px;
}
.bottom-cloud{
  position:absolute;
  left:300px;
}
```

Placing the clouds like the above:

We made the top-container(it contains all the images and the h1 and p tag) have a relative position. We then added a padding at the top of the top-container. We then made the top-cloud have an absolute position and have a margin of 300px right and 50px top relative to the top-container. We also made the bottom-cloud have a margin of 300px left relative to the top container.

Topic/Title: Font Styling in Our Personal Site

Keywords/Questions:

font-family: sans;

font-family: verdana, sans-serif;

websafe fonts

cssfontstack.com

fonts.google.com

Sacramento

Merriweather

Notes:

font-family :-used to define the font style for our website.

-written under the styling category of a body or a parent container.

-There are two commonly used text families:

- serif :- It is the default for most browsers.

-It is by default "Times New Roman"

- san-serif:- It is by default "arial". It includes "verdana" and "helvetica".

`font-family:sans-serif;` -gives us an aerial font type.

-For example, if we want to specify a specific type like verdana.

We write `font-family: verdana, sans-serif;`

-this code asks the browser to try and render all the text in the verdana font. But if the browser or operating sys. that the user's using doesn't have this font installed, then it will default to whatever is the sans serif font that is installed on their system.

- monospace, cursive and fantasy are other font families we don't use much.

- go to w3schools and search font-family when you want to edit the font.

-websafe fonts:-a set of font families where there is a maximum chance that operating systems will be able to render it correctly. But there is no 100% guarantee.

-serifs:- Georgia, Times New Roman -sans-serif: Arial, Verdana

Summary:

font-family:- used to choose the type of font.

-There are two main types: serif & sans-serif. The default one is serif.

-The others are monotone, cursive and fantasy.

websafe fonts:- fonts that have a high likely hood of being rendered.

cssfontstack.com :- tells us the chance of each font being rendered on windows and mac. It also lets us copy and help us have close alternatives if our web page isn't rendering the font we want.

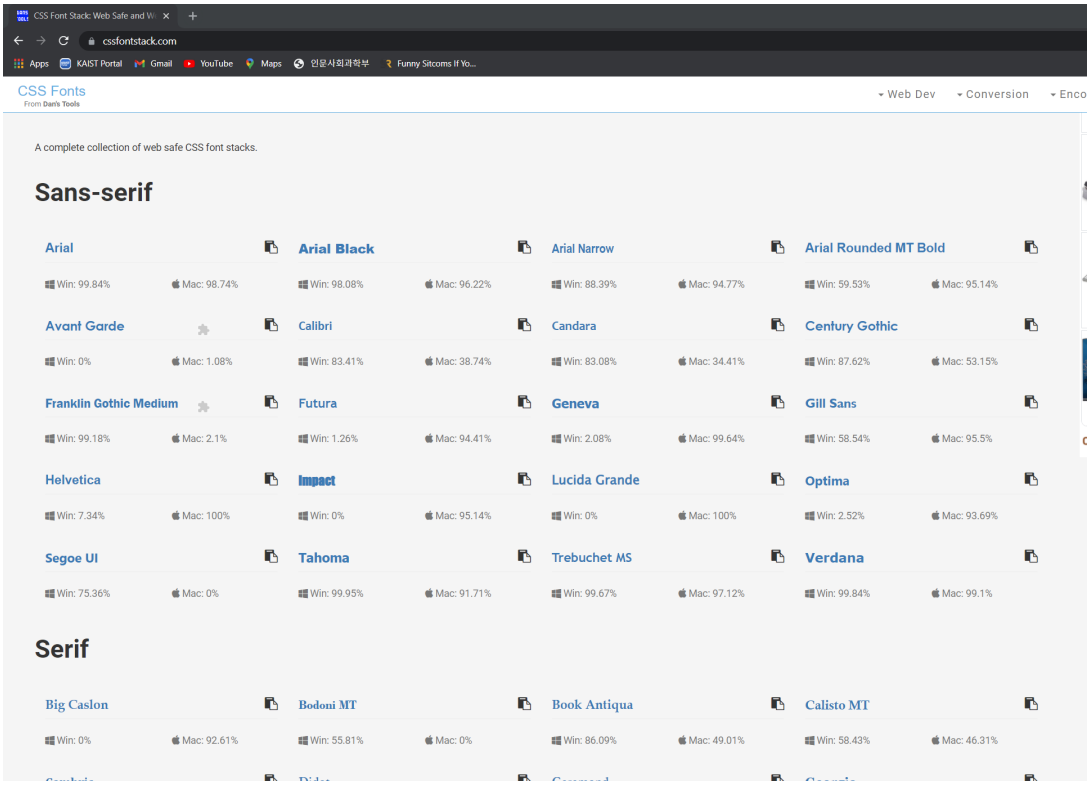
fonts.google.com:- using this font is known as ***font embedding***.

-We will have a constant styling across different websites.

More Notes

cssfontstack.com:

- It will tell us the chance of each font being displayed based on our operating system.



- If we touch the copy button ( ) and paste it in our atom CSS body styling part, we can get that particular font style with a set of font fall backs that are rendered if the particular font style couldn't be rendered. For example, if we copy and paste Helvetica we will get:

```
font-family: Helvetica Neue, Helvetica, Arial, sans-serif;
```

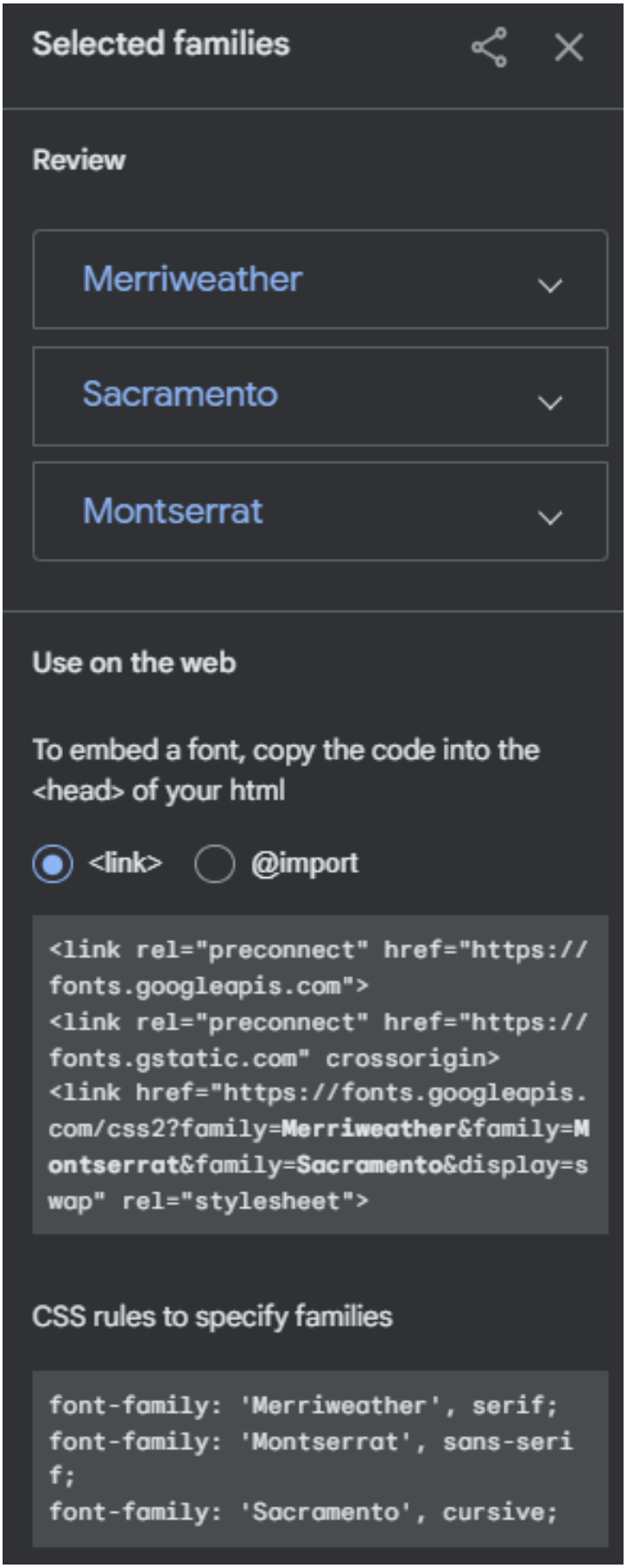
  - we will first try to render Helvetica Neue, if it couldn't be rendered then Helvetica, ... then Arial, ... then atlast sans-serif.

- If the copy button doesn't work, you can touch the font and copy the font family part from the right of the page.

fonts.google.com

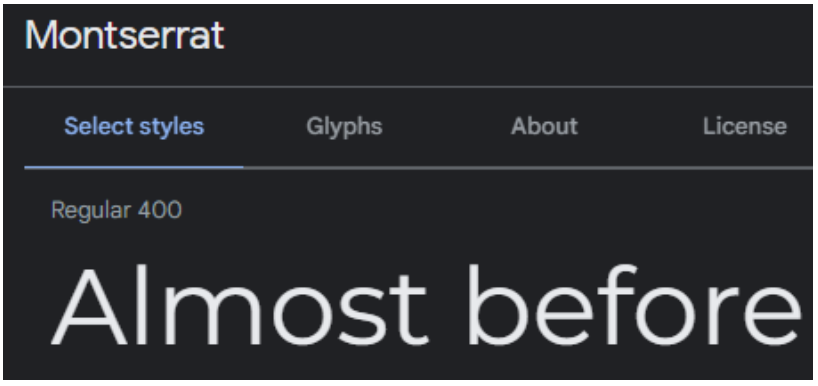
- enables us to have a constant styling across different browsers and operating systems.
- it ensures that everybody has the same viewing experience with regards to the fonts on the website. This system is called ***font embedding***.
- we choose the font styles we want. We then copy the link part and add it in the body part of our HTML code. Then, we can apply the font family we have downloaded in the CSS styling part of each element we want to style in a particular way.

fonts.google.com continued:

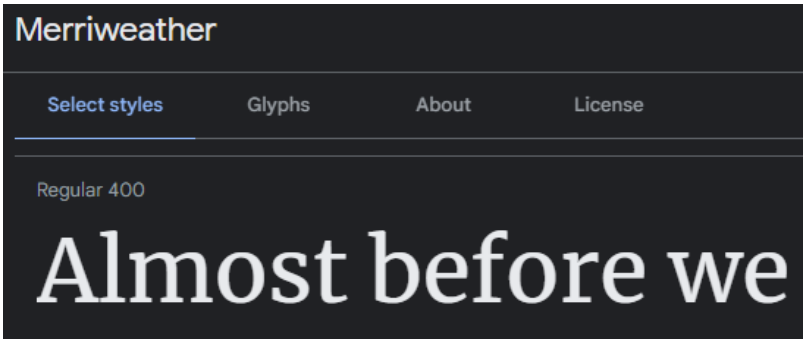


-We use Sacramento for our h1 tag. Sacramento is a bit unreadable So it can't be used for the body.

-We use Montserrat for our h2 tag,



-Then, we use Merriweather for our body. It is very commonly used in different websites.



Topic/Title: Adding Content to Our Website

Keywords/Questions:

lorem ipsum

loremipsumgenerator.com

flaticon.com

giphy.com

height:

width:

Notes:

Lorem ipsum:- it is used to place text paragraphs in a website before you have any content.

loremipsumgenerator.com:- it can be used to generate paragraphs.

-it is useful when you are creating websites and haven't got the content yet.

-the website also tells us the history of lorem ipsum.

flaticon.com:-used to find icon like images for our website that represent our various skills.

- If you find an icon you want you can download it as png and use it or u can just copy the source address and use it in your website within an img tag.

giphy.com:- you can search for animated gifs you want to include on your web site.

-giphy.com>Stickers: -If you click the stickers tab you will get transparent gifs.

- you can copy the link address and include it wherever in your website just using an <img> tag

You can modify the size of your image by getting in styles.css and using one of these 2 CSS properties: -height -width

```
<a class="hale" href="mailto:hale2024@kaist.ac.kr">CONTACT ME</a>
```

-when you touch the contact me link on this website. It will prompt you to send an email to hale2024@kaist.ac.kr using the desktop mail app. CONTACT ME

Summary:

lorem ipsum:- a placeholder text for empty paragraphs in websites.

loremimpsumgenerator.com:- generates lorem ipsum paragraphs for us.

-you can also read about the history of lorem ipsum

flaticon.com :-get images for your website. You place the images within an <img> tag.

giphy.com :- get animated gifs are to add to your website. You place within an <img> tag.

To edit the size of your image, you can use the -height and -weight CSS properties.

```
<a class="hale" href="mailto:hale2024@kaist.ac.kr">CONTACT ME</a>
```

-opens the mail app on desktop and prompts an email to hale2024@kaist.ac.kr.

Topic/Title: CSS Sizing

Keywords/Questions:

Static size

Dynamic size

px

em

%

rem

1rem=1 em

1em=16px=100%

Notes:

font-size:- used to change the size of our font.

```
h1{
  font-size:90px;
  margin-top:0;
  font-family: 'Sacramento', cursive;
}
```

-Using pixels: The font-size is statically sized. The size would remain constant even though we change the font size inside the browser.

```
h1{
  font-size:100%;
  margin-top:0;
  font-family: 'Sacramento', cursive;
}
```

-Using percentages: 100%=16px. font-size changes when we change the font size inside the browser(dynamically sized)

Changing text font-size inside Chrome:- Go to Chrome Settings>Appearance>Font size

```
font-size: 2em;
```

-1em=16px=100%

-it is dynamically sized. Its font changes when we change the font-size in chrome.

-But all of them whether em, % or px scale up when we zoom in.

```
body{
  margin:0;
  text-align:center;
  font-family: 'Merriweather', serif;
  font-size:2em;
}
h1{
  font-size:5em;
  margin-top:0;
  font-family: 'Sacramento', cursive;
}
```

-The font size for the h1 tag will be 2em +5em=7em. This property also works when using % to express our font-size.

-But when using px the font-size specified under the h1 tag will be applied alone for the h1 tag. This is because px is a static size.

Summary:

Static sizes: -px -the font size would remain the same eventhough we change it inside the chrome settings

-it is specifically assigned and doesn't sum up with parent font sizes.

Dynamic sizes:-em & %- the font size will change when we change the chrome settings.

-The font size for the h1 tag will be 2em+5em=7em. If the body tag CSS property for font size is 2em and the h1 tag's is 5em. This might result in errors due to forgeting.

rem:-best of both worlds. It changes when we change settings in chrome and doesn't add up with parent sizes.

1rem=1em=16px=100%

We can use all of them for specifying other sizes

More Notes

-rem: used to set a fixed size that works like that of px irregardless of the parent's font size. It prevents the em of the body to add with the em of the h1 tag. The em of the h1 tag will just be the em of the h1 tag.

-rem stands for root em. It means em set relative to the root. The "r" stands for root. 1rem=1em

```
body{
  margin:0;
  text-align:center;
  font-family: 'Merriweather', serif;
  font-size:200%;
}
h1{
  font-size:5.625em;
  margin-top:0;
  font-family: 'Sacramento', cursive;
}
```

-For example here the font-size of the h1 tag is 5.625em.

-The beauty of the root em is that it does not get affected by upstream size changes, and it means that it's easier to debug and it's less likely that something will go wrong.

-rem is is the most adaptable and also the most reliable and least error-prone. It has the best of both worlds:

-will change size when changing font size in Settings>Appearance>Font-size. like em & %

-it is specifically assigned and doesn't sum up with parent font sizes like px.

-We can use rem, em, % and px when specifying other sizes like margin, padding, etc.

Topic/Title: Font Properties Challenge 1 - Change the Colours Font
Properties Challenge 2 - Change the Font Weight Font
Properties Challenge 3 - Change the Line Height
CSS Font Property Challenge Solutions

Keywords/Questions:

color:#66BFBF;

color:blue;

font-weight:normal;

font-weight:400;

font-weight:bold;

font-weight:700;

line-height:2;

line-height:30%

line-height:2em;

line-height:normal; ==

line-height:1.2;

Notes:

-Changing Color of text within a tag. We usually specify it using a # followed by a hexacode or just using the name of the color:

-Selector{color:#66BFBF;} --Selector{color:blue;}

-Changing the boldness of the text within a tag. We can set it using a number or keywords. The number can be any value between 1 and 1000 with 1 being the lightest and 1000 being the boldest. The 2 most common keywords with their respective numbers are:

-normal=400 `font-weight: normal;` **-bold=700** `font-weight: 700;`

Lincoln's Inn Hall. Implacable November weather. As much mud in the streets as if the waters had but newly retired from

in Lincoln's Inn Hall. Implacable November weather. As much mud in the streets as if the waters had but newly

-Changing the gap in between two lines of text within a tag. We can set it using:

-a number: line-height= the element's font size × the number. ***Preferred method***

```
line-height: 3.5;
```
-line-height= element's font size × 3.5

-a length:(em,px,..)-specify a particular space

```
line-height: 3em;
```

-a percentage: line-height=a specific percentage of the font size

```
line-height: 34%;
```
-line-height=0.34×element' font size

-normal:-depends browser/fontfamily but usually=1.2(a number).***line-height:normal;***

Summary: color:- a CSS property used to change color of a text. -usually specified using #hexacode or a color name.

line-weight: a CSS property used to modify text boldness

-usually specified as a number from 1 to 1000 or normal & bold keywords -normal=400 -bold=700

line-height: a CSS property used to specify distance in between lines. Usually specified using:

-a number: line height=font size × the number

-a percentage: line height=font size×(percentage/100)

-a length: line height=the length. specified in px, em,...

-the normal key word: line-height=normal usually is the same as line-height:1.2.

Topic/Title: CSS Float and Clear

Keywords/Questions:

File>Settings>Editor>Show Indent Guide

text-align:left;

width:25%

float:left;

clear:left;

float:right;

clear:right;

float:inherit;

float:none;

Notes:

Go to File>Settings>Editor. Then, tick the **Show Indent Guide** option.

-It creates lines which show you the opening and closing tag of elements.

-It especially useful for seeing the opening and closing tag of elements.

text-align:left;

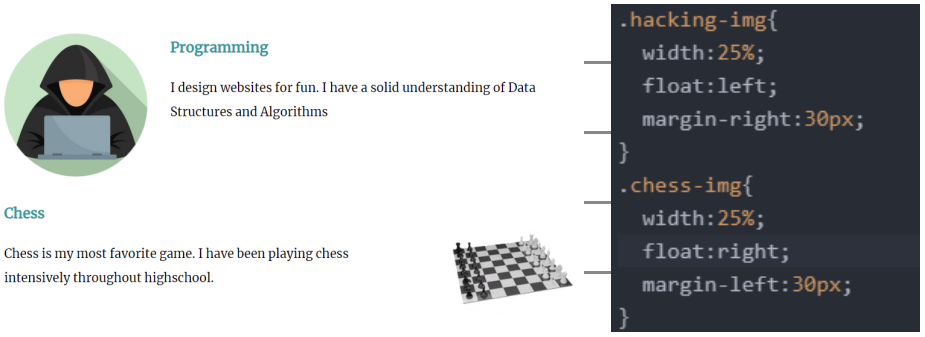
- aligns text or image to the left of the container.

- text alignment property in a child will override one in the parent.

width:25%;

- the width would be set relative to the parent element: div or body or ....

float property-used for wrapping text around a certain element e.g. let an image float left to the text in a container. By default the float property for all elements is float:none;



float:left;- element floats to the left of its container.

float:right; :-element floats to the right of its container.

float:inherit:- element inherits the float value of its parent.

Summary:

Open Atom>File>Settings>Editor>Show Indent Guide option :-draws a line between the start&end of an element.

text-align:left; :- aligns the element to the left of the container.

width:- css property to set the width of a block or inline-block element. If specified in %, its relative to its container.

float:-used for wrapping elements around certain elements. We use terms like

-float:left; -float:right; -float:inherit; -float:none;

clear: serves as an anti-float. It prevents the effect of float from being applied on elements. We can use terms like

-clear:left; -clear:right;

More Notes

-How do we prevent the effect of float from being applied on some of the elements?

-We can use the **clear** property.

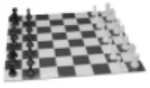
-clear: serves as an anti-float.

-we apply the clear property to a part of text we don't want it to be wrapped next to the element we floated.

Programming

I design websites for fun. I have a solid understanding of Data Structures and Algorithms. I do coding interview problems in my free time, and enjoy figuring out new ways to solve problems.

Chess



Chess is my most favorite game. I have been playing chess intensively throughout highschool.

```
.programming-img{
  width:25%;
  float:left;
  margin-right:30px;
}
.programming-text{
  clear:left;
}
.chess-img{
  width:25%;
  float:right;
  margin-left:30px;
}
.chess-text{
  clear:right;
}
```

-we prevented the programming and chess description texts from wrapping the images.

-Angela recommends to only use float when really, really necessary and only use it for what it's meant, which is wrapping text around a certain element and don't use it for positioning. Instead, use what we learned about positioning like relative or absolute positioning or using the margin and the padding for the layout and the positioning of your website, and to leave the float property only for the cases of wrapping text.

Topic/Title: CSS Challenge
Get More Practice HTML and CSS
Tip from Angela - Nothing Easy is Worth Doing!

Keywords/Questions:

css3buttongenerator.com

border-color:

border-width:

border-bottom:none;

border:solid 6px blue;

text-decoration:none;

Section 61 challenges

Notes:

css3buttongenerator.com

-you can edit and copy the code to the css part and add the class ***btn*** to one of your <a> tags.

border-color: used to change the color of our border

border-style: used to change the style of our border

border-style: dotted none dotted;

border-width: used to specify the thickness of the border

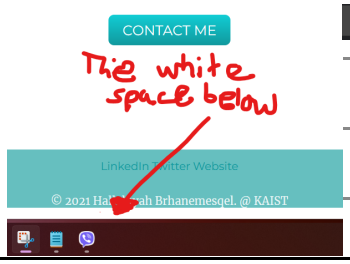
border: dotted #EAF6F6 6px; -order doesn't matter -same as border:#EAF6F6 dotted 6px;

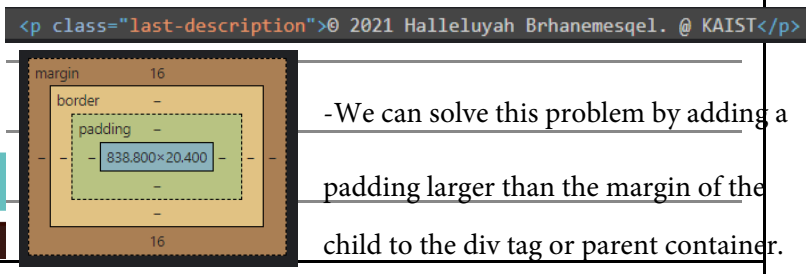
- a shorthand property for border-style, border-width and border-color.

border-bottom:none;

-means the border box won't have a bottom or it won't horizontally have a second line.

If there is a parent tag(let us say it is a tag) and if there is a child tag(let us say it is a <p> tag), if the <p> tags margin is more than the space in the div tag. The margin would be outside the div tag(parent container). So, there would be a white space outside the container.





-We can solve this problem by adding a padding larger than the margin of the child to the div tag or parent container.

Summary:

-css3buttongenerator.com :-used to create a button.

-border-color: changes the color of the border

-border-width: changes the thickness of the border

-border-style: choosing the style of the border. i.e solid, dotted, ...

-border: dotted #EAF6F6 6px; :- a shorthand property for border-style, border-width and border-color.

-border-bottom:none; :-means the border will be a single horizontal line. removes the bottom part.

-text-decoration:none; :- removes the underline from elements by default underlined like anchor tags(<a>)

-Section 61 challenges.

More Notes

text-decoration:none;

-it removes an underline from elements underlined by default like <a>(anchor) tags.

https://www.frontendmentor.io/challenges/space-tourism-multipage-website-gRWj1URZ3

-website to get more practice on html and css by building stuff.