

Rem and Em

Both Rem and Em are values that indicates the size of something in CSS.

For example font-size = 1rem or 1em they both mean that the

Font size that will be computed will have the value 16px but there is a difference as rem is independent of parents elements but rather the root HTML sizing, meaning that if we type

font-size = 1rem that means that our margin is going to be 16px in size just like our root HTML size while,

Em inherits from it's parent component not the root HTML

CSS Positioning

we have five types of positioning in CSS .

Relative , Fixed , Absolute , Static and Sticky .

Position : relative

makes the component's position relative to the component it's in while maintaining the space it had in the original document flow as it's not removed from there.

Position : Fixed

makes the component's position relative to the initial containing block where it was created selected by the viewport except if the container has a transform , filter or perspective features enabled , it will not have space created for it in the document flow

Position : Absolute

The element is removed from the normal document flow, and no space is created for the element in the page layout and It is positioned relative to its closest positioned parent, if there is a parent otherwise, it is placed relative to the initial containing block. Its final position is determined by the values of top, right, bottom, and left.

Position : Static

Default Position value , the element is positioned according to the normal flow of the document. The top, right, bottom, left, and z-index properties have no effect.

Position : Sticky

The element is positioned according to the normal flow of the document, and then offsets relative to its nearest scrolling Container

For and while loops

Both of these keywords are used to loop a certain code block a certain amount of times as in automate a process rather than doing it manually every time,

technically speaking both loops have no difference as they are inter-replaceable meaning I can make a for loop logic using while loop code and vice versa , difference though is in the structure .

a for loop is declared in the following way

➔ `for (initialization , condition , iteration (inc/dec)) {}`

meaning that we have to know the number of iterations we are going to make just so our condition would be valid , for loops are used mostly in running through arrays of data automating the data fetching process

while a while loop is declared in the following way

➔ `while (condition) {}`

while is used generally without a specific usage as it solely dependent on how are you going to structure it , it can fail without doing a single iteration or working like a for loop that has a counter or just wait for a Boolean condition to be true/false

Object methods

They are methods that are defined in the object body and can be accessed by instantiating the object and accessing it's methods through it

Regular vs arrow Functions

A regular function has the same functionality as the arrow function with just some key differences on the arrow function side.

An arrow function is basically used to lower the amount of code written to do the same function as the regular one and it was introduced in ES6 , the (this) identifier no longer binds in the arrow function , you also can't have duplicate named parameters in arrow functions along with constructing the function using the new keyword , you can only call them

Object vs Instance

Instancing means making a virtual copy of something, making an object is the making an instance of the class it was derived from and saving some space in the memory for the memory addresses