**Search**

* **Rem and Em**

**Both of them are scalable and relative units of size that will be computed to pixel px**

**Em: the unit is relative to the font size of its parent element**

**Rem: relative to the root font size of the html document (r in rem stand for root)**

**If we type font size 1 rep that mean it will be computed to 16px like root html size ,em inherits from it’s parent not the root html**

* **Type of positioning**

**Relative:**

**The element is positioned according to the normal flow of the document and then offset relative to itself based on the values top, right, bottom and left. The offset does not affect the position of any other element**

**Fixed:**

**The element is removed from the normal document flow, and no space is created for the element in the page layout.it is positioned relative to the initial containing block established by the viewport**

**Absolute:**

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

**Static:**

**The element is positioned according to the normal flow of the document. The top, right , bottom, left properties have no effect. This is the default value.**

**Sticky:**

**The element is positioned according to the normal flow of the document, and then offset relative to its nearest scrolling ancestor and containing block (nearest block- level ancestor), including table-related elements, based on the values of top, right ,bottom, and left. The offset doesn’t affect the position of any other elements**

* **For and while loops**

**Both of them serve the same idea of making loop in a code block for certain amount of times rather then calling doing it manually every time and the difference between them is the declaration**

**For:**

**For (initialization, condition, iteration){}**

The for loop is the most used repetition statement in almost all programming languages. The reason for its popularity is that it does three jobs at once.

It initializes the counter, tests the condition and increments/decrements the counter at the same time.

This is also the key difference which makes for loop more acceptable over the while and do while loop for most programmers.

**while:**

**while(condition){}**

**initialization**

**iteration**

in the while statement, first a counter is initialized which undergoes a test condition. If the condition evaluates to true, then the body of the loop gets executed. Hence, while loop runs until the condition becomes false.

* **Object methods**

**Object are usually created to represent entities of the real world like user, orders and so on and like in the real world a user can act like login and logout so in programing methods and makes object act**

**Example : sayhi(){ console.log(“hello”+ user)**

**Let user;**

**User.sayHi();**

* **Regular vs arrow function**

**A regular and arrow have the same functionality with some key difference on the arrow function side**

**An arrow function is 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 bind in the arrow function, you also can’t have duplicate named parameters in arrow function 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 the saving some space in the memory for the memory addresses**