1 )

|  |  |
| --- | --- |
| **REM** | **EM** |
| **REM** is relative to the root (HTML) font size, so if you wish to scale the element’s size based on the root size, no matter what the parent size is, use REM. If you’ve used EM and are finding sizing issues due to lots of nested elements, REM will probably be the better choice. | **EM** is relative to the parent element’s font size, so if you wish to scale the element’s size based on its parent’s size, use EM. |

2) **CSS Position :**

a) What is css position ?

The CSS position property defines the position of an element in a document .

**b)Type of css position :**

* Static : The element is positioned according to the normal flow of the document. The left, right, top, bottom and z-index
* Relative : Elements in relative position Remain in the normal flow of the document. But, unlike static elements, the left, right, top, bottom and z-index properties affect the position of the element. An offset, based on the values of left, right, top and bottom properties, is applied to the element relative to itself.

**Absolute** : Elements with absolute position are positioned relative to their parent elements. In this case, the element is removed from the normal document flow. The other elements will behave as if that element is not in the document. No space is created for the element in the page layout. The values of left, top, bottom and right determine the final position of the element.

* Fixed : Fixed position elements are similar to absolutely positioned elements. They are also removed from the normal flow of the document. But unlike absolutely positioned element, they are always positioned relative to the html element.
* Sticky :sticky position is a mix between the fixed and relative It acts like a relatively positioned element until a certain scroll point and then it acts like a fixed element. Have no fear if you don't understand what this means, the example will help you to understand it better.

**3) for loop and while loop**

**Key Differences Between For Loop and While Loop**

|  |  |
| --- | --- |
| **For Loop** | **While Loop** |
| It is used when the number of iterations is known. | It is used when the number of iterations is not known. |
| In case of no condition, the loop is repeated infinite times. | In case of no condition, an error will be shown. |
| Initialization is not repeated. | Initialization is repeated if carried out during the stage of checking. |
| Statement of Iteration is written after running. | Can be written at any place. |
| Initialization can be in or out of the loop | Initialization is always out of the loop. |
| The nature of the increment is simple. | The nature of the increment is complex. |
| Used when initialization is simple. | Used when initialization is complex. |

4) object methods :

JavaScript methods are actions that can be performed on objects.

A JavaScript **method** is a property containing a **function definition**

Methods are functions stored as object properties

## **Accessing Object Methods**

You access an object method with the following syntax:

*objectName.methodName()*

5 ) regular vs arrow function js :

Understanding the differences between regular and arrow functions helps choose the right syntax for specific needs.

In regular function, this changes according to the way that function is invoked.

The value of this inside an arrow function remains the same throughout the lifecycle of the function and is always bound to the value of this in the closest non-arrow parent function which means No matter how or where being executed, this value inside of an arrow function always equals this value from the outer function.

6) **objects vs instance oop :**

Instance: instance means just creating a reference(copy).

object: means when memory location is associated with the object (is a run-time entity of the class) by using the new operator.

In simple words, Instance refers to the copy of the object at a particular time whereas object refers to the memory address of the class.