

## Various methods of console function

The Console provides you with the ability to write, manage, and monitor JavaScript on demand. The console is an object which provides access to the browser debugging console

The console object provides us with several different methods, like :

**1) log()** :- Mainly used to log(print) the output to the console. We can put any type inside the log(), be it a string, array, object, boolean etc.

```
for e.g. console.log('abc');
          console.log(1);
          console.log(true);
          console.log(null);
          console.log(undefined);
          console.log([1, 2, 3, 4]); // array inside log
          console.log({a:1, b:2, c:3}); // object inside log
```

**2) error()** :- Used to log error message to the console. Useful in testing of code. By default the error message will be highlighted with red color.

```
for e.g.      console.error('This is a simple error');
```

**3) warn()** :- Used to log warning message to the console. By default the warning message will be highlighted with yellow color.

```
for e.g.      console.warn('This is a warning.');
```

**4) clear()** :- Used to clear the console. The console will be cleared, in case of Chrome a simple overlayed text will be printed like : 'Console was cleared' while in firefox no message is returned.

```
for e.g. console.clear();
```

**5) time() and timeEnd()** :- Whenever we want to know the amount of time spend by a block or a function, we can make use of the time() and timeEnd() methods provided by the javascript console object. They take a label which must be same, and the code inside can be anything( function, object, simple console).

```
for e.g.      console.time('abc');
              let fun = function(){
                  console.log('fun is running');
              }
```

```

let fun2 = function(){
    console.log('fun2 is running..');
}
fun(); // calling fun();
fun2(); // calling fun2();
console.timeEnd('abc');

```

Output: fun is running  
fun2 is running  
abc: 1ms

**6) table()** :- This method allows us to generate a table inside a console. The input must be an array or an object which will be shown as a table.

for e.g. `console.table({'a':1, 'b':2});`

**7) count()** :- This method is used to count the number that the function hit by this counting method.

for e.g. `for(let i=0;i<5;i++){  
 console.count(i);  
}`

Output:- 0 :1  
1 :1  
2 :1  
3 :1  
4 :1

**8) group() and groupEnd()** :- group() and groupEnd() methods of the console object allows us to group contents in a separate block, which will be indented. Just like the time() and the timeEnd() they also accepts label, again of same value.

for e.g. `console.group('simple');  
 console.warn('warning!');  
 console.error('error here');  
 console.log('vivi vini vici');  
 console.groupEnd('simple');  
 console.log('new section');`