

JS NUGGETS

LIFE IN JAVASCRIPT

IIFE - **I**mmediately **I**nvoked **F**unction **E**xpression is a way to execute functions immediately, as soon as they are created. This is a good way to protect the scope of your function and the variables within it.

```
(function () {  
    // write your js code here  
})();
```

Named **IIF** with parameters

```
(function showCurrentDate(date) {  
    console.log('I m from IIFE', date)  
})(new Date());
```

WHY IIFE IS SO USEFUL? 🤔

IIF is the most better way to avoid,

- Polluting global variables
- Overwrite global scope.

```
var name = '🌙';  
(function () {  
    var name = '★';  
    console.log("Hello "+name);  
})();  
console.log("Hello "+name);
```

OUTPUT:

```
Hello ★  
Hello 🌙
```

Another use case for an IIFE is, it can be used as a closure

```
const generateUniqueId = (function() {
  var counter = 0;
  return function() {
    ++counter;
    return `JSN_${counter}`;
  };
})();
console.log(generateUniqueId()); // JSN_1
console.log(generateUniqueId()); // JSN_2
console.log(generateUniqueId()); // JSN_3
```

The *counter* variable is inaccessible from outside of the IIFE. Except for the function that's being returned, nobody can read or modify the *counter* variable.

Note: In ES6, we can just use curly braces ({}) and use **const** or **let** to get the same effect.