

```

1 /***** 12_FUNCTIONS *****/
2
3
4 /**
5  * FUNCTIONS:
6  * --> A JavaScript function is a block of code designed to perform a
particular task.
7  * --> A JavaScript function is executed when "something" invokes it (calls
it).
8  * --> Three Ways To Create Functions:
9  *         --> 1. Function Keyword
10 *         --> 2. Function Expression
11 *         --> 3. Arrow Function (ES6)
12 */
13
14 // FUNCTION DECLARATION
15 function greet() {
16     return 'Hello';
17 }
18
19 // FUNCTION CALL
20 console.log(greet());
21
22 /**
23  * FUNCTION WITH PARAMETERS
24  */
25 function add(x, y) {
26     return x + y;
27 }
28 console.log(add(55, 65));
29
30
31 /**
32  * FUNCTION WITH DEFAULT PARAMETERS (ES5)
33  */
34 function square(x) {
35     if (typeof x === 'undefined') {
36         x = 1;
37     }
38     return x * x;
39 }
40 console.log(square());
41
42 /**
43  * FUNCTION WITH DEFAULT PARAMETERS (ES6)
44  */
45 function mult(x = 1, y = 1) {
46     return x * y;
47 }
48 console.log(mult());
49
50
51 /**
52  * FUNCTION EXPRESSION
53  */
54 const message = function(name = 'Anonymous') {
55     return `Hello, ${name}`;
56 }
57 console.log(message('Abhishek Baghel'));
58

```

```
59 /**
60  * IMMEDIATELY INVOKABLE FUNCTION EXPRESSIONS - IIFEs
61  * --> Used In 'Module Design Pattern'
62  */
63 (function() {
64     console.log('IIFE Ran....');
65 })();
66
67 /**
68  * IIFEs - with parameters
69  */
70 (function(name) {
71     console.log(`Hello, ${name} --> You Ran IIFE`);
72 })(`Abhishek Baghel`);
73
74 /**
75  * PROPERTY METHODS:
76  * --> When a function is declared inside an object then it is called
77  * 'PROPERTY METHODS'
78  */
79 const notes = {
80     // ES5 Style
81     add: function() {
82         console.log('Add Note');
83     },
84     // ES6 Style
85     remove() {
86         console.log('Note Delete')
87     }
88 };
89
90 notes.edit = function() {
91     console.log('Note Edit');
92 }
93
94 notes.add();
95 notes.remove();
96 notes.edit();
```