

JAVASCRIPT FUNCTIONS

INCLUDING ES6 CONCEPTS

FUNCTION TYPES

❑ FUNCTION DECLARATION

```
FUNCTION NAME () {
```

```
    STATEMENT;
```

```
}
```

❑ FUNCTION EXPRESSION

```
CONST NAME = FUNCTION () {
```

```
    STATEMENT;
```

```
}
```


FUNCTION TYPES

❑ ARROW FUNCTION

- `CONST NAME = () => {`

`STATEMENT;`

`}`

FUNCTION DECLARATION

- FUNCTION NAME (PARAMETER01, PARAMETER02) {
 STATEMENT;
}
NAME(ARGUMENT01, ARGUMENT02);

FUNCTION EXPRESSION

- `CONST NAME = FUNCTION (PARAMETER01, PARAMETER02) {`

`STATEMENT;`

`}`

`NAME(ARGUMENT01, ARGUMENT02);`

ARROW FUNCTION

- `CONST NAME = (PARAMETER01, PARAMETER02) => {`

`STATEMENT;`

`}`

`NAME(ARGUMENT01, ARGUMENT02);`

RETURNING VALUES

```
CONST ADD = (A, B) => {
```

```
  LET C = A + B;
```

```
  RETURN C;
```

```
}
```

```
LET SUM = ADD(2, 2);
```

```
CONSOLE.LOG(SUM)
```

RETURNING VALUES – EXAMPLE

```
275  const calcProd = function (products, tax) {  
276    let total = 0;  
277    for (let i = 0; i < products.length; i++) {  
278      total += products[i] + products[i] * tax;  
279    }  
280    return total;  
281  };  
282  
283  let productC = calcProd([20, 30, 40], 0.2);  
284  console.log(productC);  
285
```


SET TIME OUT

SETTIMEOUT(FUNCTIONNAME/ANONYMOUS FUNCTION, MILLISECOND)

```
//Set Time Out
setTimeout(() => {
  let employees = ["Faraz", "Aqib", "Shahab", "Ali"];
  console.log(employees);
}, 3000);
```

SET TIME OUT - EXAMPLE

SETTIMEOUT(FUNCTIONNAME/ANONYMOUS FUNCTION, MILLISECOND)

```
335  let box = document.querySelector("#box");
336
337  setTimeout(() => {
338      box.style.width = "500px";
339  }, 3000);
```


FOREACH METHOD - EXAMPLE

```
LET DAYS = [MON, TUES, WED, THURS, FRI];
```

```
DAYS.FOREACH(FUNCTION(DAY) {
```

```
    CONSOLE.LOG(DAY);
```

```
}
```

FOREACH METHOD - EXAMPLE

```
324 list = document.querySelector(".people");
325
326 html = ``;
327
328 const students = ["Aqib", "Aqib", "Aqib", "Aqib"];
329
330 students.forEach((person) => {
331     html += `<li style="color: red">${person}</li>`;
332 });
333
334 list.innerHTML = html;
```


CALLBACK FUNCTION

- ❑ ANY FUNCTION THAT IS PASSED AS AN ARGUMENT IS CALLED CALLBACK FUNCTION.
- ❑ A CALLBACK IS A FUNCTION THAT IS TO BE EXECUTED AFTER ANOTHER FUNCTION HAS FINISHED EXECUTING – HENCE THE NAME ‘CALL BACK’.

CALLBACK FUNCTION

- `CONST FUNC = FUNCTION(CALLBACKFUNC) {`
- `LET VALUE = 20;`
- `CALLBACKFUNC();`
- `}`
- `FUNC(FUNCTION(VALUE) {`
- `CONSOLE.LOG(VALUE);`
- `})`

CALLBACK FUNCTION

```
312  const mesgOne = (day, time) => {  
313    console.log(`Hello, Today is ${day}`);  
314    ⚡ time();  
315  };  
316  
317  const mesgTwo = (time) => {  
318    time = "Morning";  
319    console.log(`Good ${time}!`);  
320  };  
321  
322  mesgOne("Monday", mesgTwo);
```