Date __/__/____

Functions

Block of code that performs a specific ~~code~~ task, can be invoked whenever needed.

↳ call

To create a function :-

1.> Function defination

parametres / input

```
function functionName () {
    // do some work
}
```

in function defination we tell our function ki tumhe Karna kya hai.

Function saves us from redundancy.

2.> Function call

Parametres that needs to be passed to this function.

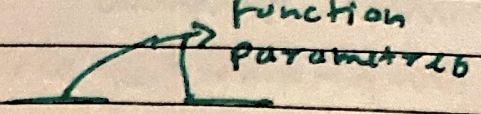
```
functionName ( );
```

↳ This syntax invokes the function when called.

↳ The value passed in function call is termed as ~~parameters~~ Arguments & The value which is recieved in function defination is called Parametres.

Date __/__/____

For eg:-




```
function addFunction(num1, num2) {
// Parameters num1 & num2 is a local variable
for this function & will be put in use only inside
this function block.

// it cannot be accessed globally.

let sum = num1 + num2;
console.log("Before return");
return sum;  // used to return some value from
console.log("After return");      the function.
```

↳ this particular line will never be executed as function has already returned some value.

☆ only one thing at a time can be returned from a particular function.

```
};

// Function call to invoke the add function
let sum = addFunction(1, 2);
                      ↳ Arguments.
console.log(sum);
        ↳ this line will print the value that
is returned from the addFunction.
```

Date __/__/____

Arrow function :-

compact way of writting a function

Syntax :-

```
const functionName = (param1, param2, ....)
    => {
        // do some work
};
```

Foreach Loop in Arrays :-

arr.forEach (callBackFunction)

callbackFunction : Here, it is a function to execute for each element in the array.

* A callback is a function passed as an argument to another function.

```
arr.forEach ((val) => {
        console.log(val);
});
```

For eg :-

```
arr = ["Mumbai", "Pune", "Bhopal"];

arr.forEach ((val) => {
        console.log(val)
});
```

Date __/__/____

// For each is a higher order function / method which takes a function as a parameter or returns a function.

Map method.

→ creates a new array with the results of some operations. The value it's callback returns are used to form a new array.

Syntax

Array-name.map (callback Fxn (value, index, array))

```
let nums = [1, 2, 3, 4, 5, 6, 7];
let newNums = nums.map (val) => {
    return val*2
});

console.log (newNumsArray);
```

That particular code we generate an new newNumsArray = [2, 4, 6, 8, 10, 12, 14];