

• github : divyaranjan10



DATE \_\_\_\_\_

## Chap 18: Higher-order functions ft. Functional Programming

=> A function which takes another function as an ~~or~~ argument or returns a function from it, is called Higher Order function.

```
=> function x () {  
    console.log("x called");  
}
```

└──────────> Higher Order function

```
function y (x) {  
    console  
    x();  
}
```

└──────────> Callback function

• Optimized way of writing the code using Higher-order function:

• Calculate area and circumference of circle:

```
const radiuses = [3, 4, 5, 6];
```

```
const area = function (radius) {
```

```
    return Math.PI * radius * radius;
```

```
}
```





DATE \_\_\_\_\_

```
const circum = function (radii) {  
  return 2 * Math.PI * radii;  
}
```

```
const calculate = function (radii, formula) {  
  let output = [];  
  for (let i = 0; i < radii.length; i++) {  
    output.push(formula(radii[i]));  
  }  
  return output;  
}
```

```
console.log(calculate(radii, area));  
console.log(calculate(radii, circum));
```

\* In this code example, we get to know about the beauty of functional programming and the power of its reusability \*