

Higher-Order Function ft. functional programming

HOF

A function which takes a function as -
a argument or ~~argument~~ return a -
function is called HOF

example

→ x is call back function

```
function x() {  
  console.log("Namaste");  
}
```

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

→ y is HOF

example

- ① DRY Principle
- ② Optimize our code

```
const area = function (radius) {  
  return 2 Math.PI * radius * radius;  
};
```

```
const circumference = function (radius) {  
  return 2 * Math.PI * radius;  
};
```

```
const diameter = function (radius) {  
  return 2 * radius;  
};
```



```

Array.prototype.calculate = function (logic) {
  const output = [];
  for (let i = 0; i < this.length; i++) {
    output.push(logic(this[i]));
  }
  return output;
};

```

```

console.log(radius.map(area));
console.log(radius.calculate(area));

```

Functional programming deal with

- ① Pure function
- ② Composition of function
- ③ Reusability
- ④ modularity

example

① DRY Principle

② optimise our code

equivalent