<script>

var v = [10, 20, 30, 5, 15, 25];

var m = v.map(x => x \* 5); //x assigned to x \* 5

for(i of m){

document.writeln(i + “<br>”);

}

</script>

**Functional Components**

|  |  |
| --- | --- |
| **Functional Components / Tags** | **Class Components / Tags** |
| It is easy to implement since it takes lesser coding than Class Tags. | It is not simple as Functional Components. |
| When less amount of data need to be handled and less functionalities are there then go for Functional Tags. | When lot of functionalities and data is involved then go for Class Tags. |
| Does not have built-in state. | Uses state. |
| Does not get all benefits of OOPS. | Gets all benefits of OOPS.   * Inheritance * Polymorphism * Encapsulation/Abstraction |
| Has just return. | Has a render function with return inside it. |
| No life cycle phases. Because it doesn’t have state. | Has life cycle. |
| Does not have this operator. | Has this operator. |

See tutorials point dont study this table