Name: Tushar Bharti
Reg: 12015881

PEP DOCS-2

Q1. Describe higher order functions ?

Ans: Higher Orders Functions are functions that perform operations on other functions.

For example, we have two functions , func A and func B. Now if func A is taking func B in its arguments or func A is returning func B as a result, then we can say that func A is a higher order function.

Like this:

```
const nums = [1, 2, 3, 4, 5];

nums.forEach((number) => console.log(number + 1));
```

HIGHER ORDER FUNCTION:

1. REDUCE

const nums = [1, 2, 3, 4, 5];

const totalValue = nums.reduce((sum, number) => sum + number);

console.log(totalValue);

- Sum is the prefix sum of the elements whereas,number is the current element.
- The higher order function reduce() expects two parameters in the anonymous function within.
- The first parameter is an accumulator and the second parameter is an element from the numbers array.

2. MAP

Lets say we want to add one to each element of the array and put it inside a new array. We can use map() function to do this.

```
const nums = [1, 2, 3, 4, 5];

const newArray = nums.map((number) => number + 1);

console.log(nums);
```

map() function creates a new array and push values inside of it after adding one.

3. FILTER

If I wanted to create a new array that only has the odd numbers extracted from the original array, I could do the following:

```
const nums = [1, 2, 3, 4, 5];

const oddArray = numbers.filter((number) =>
number % 2 !== 0);

console.log(oddArray);
```

Filter() will create a new array. The higher-order function will return each element that meets the condition set within the anonymous function it receives.