FSD Assignment – 9

Q1. Explain the .map() function in JavaScript and provide three examples with detailed explanations. Q2. Explain the .reduce() function in JavaScript and provide three examples with detailed explanations. Q3. Explain the .filter() function in JavaScript and provide three examples with detailed explanations.

Here’s a simple explanation of each function along with three of the easiest examples:

1: `.map()` Function

Theory: The `.map()` function in JavaScript is used to create a new array by applying a function to each element of an existing array. It’s commonly used when you want to transform each element in an array in the same way.

Example 1: Doubling Numbers

```
const numbers = [1, 2, 3];

const doubled = numbers.map(x => x * 2);

console.log(doubled); // Output: [2, 4, 6]
```

Explanation: This doubles each number in the `numbers` array.

Example 2: Adding 1 to Each Number

```
const numbers = [1, 2, 3];
const incremented = numbers.map(x => x + 1);

console.log(incremented); // Output: [2, 3, 4]
```

Explanation: Each number in the array is increased by 1.

Example 3: Converting to Strings

```
const numbers = [1, 2, 3];

const strings = numbers.map(x => x.toString());

console.log(strings); // Output: ["1", "2", "3"]
```

Explanation: Each number is converted to a string.

Q2: `.reduce()` Function

Theory: The `.reduce()` function in JavaScript is used to accumulate all the elements in an array into a single value. This could be a sum, a product, or any other operation that combines all the elements.

Example 1: Summing Numbers

```
const numbers = [1, 2, 3];

const sum = numbers.reduce((total, x) => total + x, 0);

console.log(sum); // Output: 6
```

Explanation: This adds up all the numbers in the array.

Example 2: Multiplying Numbers

```
const numbers = [1, 2, 3];

const product = numbers.reduce((total, x) => total * x, 1);

console.log(product); // Output: 6
```

Explanation: This multiplies all the numbers in the array.

Example 3: Concatenating Strings

```
const words = ["Hello", "World"];

const sentence = words.reduce((result, word) => result + " " + word);

console.log(sentence); // Output: "Hello World"
```

Explanation: This concatenates the strings in the array into a single sentence.

Q3: `.filter()` Function

Theory:

The `.filter()` function in JavaScript is used to create a new array that only contains elements that meet a certain condition. It’s helpful when you want to select specific elements from an array.

Example 1: Filtering Even Numbers

```
const numbers = [1, 2, 3, 4];
const even = numbers.filter(x => x % 2 === 0);

console.log(even); // Output: [2, 4]
```

Explanation: This keeps only the even numbers.

Example 2: Filtering Numbers Greater Than 2

```
const numbers = [1, 2, 3, 4];

const greaterThanTwo = numbers.filter(x => x > 2);

console.log(greaterThanTwo); // Output: [3, 4]
```

Explanation: This keeps only numbers greater than 2.

Example 3: Filtering Words with More Than 3 Letters

```
const words = ["hi", "hello", "hey"];

const longWords = words.filter(word => word.length > 3);
```

```
console.log(longWords); // Output: ["hello"]
```

Explanation: This keeps only words that have more than 3 letters.