

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:

```
const words = ["hi", "bye", "thanks"];

const excitedWords = words.map(word => word + "!");

console.log(excitedWords); // Output: ["hi!", "bye!", "thanks!"]
```

Output: ["hi!", "bye!", "thanks!"]

Explanation: This adds an exclamation mark to each word to make them more exciting.

Example 2:

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

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

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

Explanation: This doubles the value of each number in the array.

Example 3:

```
const names = ["Alice", "Bob", "Charlie"];

const greetings = names.map(name => "Hello " + name);

console.log(greetings); // Output: ["Hello Alice", "Hello Bob", "Hello Charlie"]
```

Explanation: This adds "Hello" before each name to greet them.

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:

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

const totalApples = apples.reduce((total, num) => total + num, 0);

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

Explanation: This adds up all the apples to find the total.

Example 2:

```
const prices = [10, 20, 15];

const totalCost = prices.reduce((total, price) => total + price, 0);

console.log(totalCost); // Output: 45
```

Explanation: This sums up the prices to find the total cost.

Example 3: Concatenating Strings

```
const items = ["milk", "bread", "butter"];

const shoppingList = items.reduce((list, item) => list + ", " + item);

console.log(shoppingList); // Output: "milk, bread, butter"
```

Explanation: This combines all items into a single shopping list 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:

```
const apples = ["red", "green", "yellow", "green"];

const greenApples = apples.filter(apple => apple === "green");

console.log(greenApples); // Output: ["green", "green"]
```

Explanation: This keeps only the green apples from the list.

Example 2:

```
const items = ["banana", "apple", "bread", "carrot"];

const bItems = items.filter(item => item.startsWith("b"));

console.log(bItems); // Output: ["banana", "bread"]
```

Explanation: This keeps only the items that start with the letter "B".

Example 3:

```
const numbers = [5, 12, 8, 20];

const greaterThanTen = numbers.filter(num => num > 10);

console.log(greaterThanTen); // Output: [12, 20]
```

Explanation: This filters out numbers greater than 10.