Filtering even numbers:

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const evenNumbers = numbers.filter((num) => num % 2 === 0);

console.log(evenNumbers); // Output: [2, 4, 6, 8, 10]

Filtering strings with a specific length:

const fruits = ['apple', 'banana', 'kiwi', 'orange', 'grape'];

const fruitsWithLengthFive = fruits.filter((fruit) => fruit.length === 5);

console.log(fruitsWithLengthFive); // Output: ['apple', 'grape']

Filtering objects with a certain property value:

const products = [

  { name: 'Laptop', price: 1000 },

  { name: 'Smartphone', price: 500 },

  { name: 'Tablet', price: 800 },

  { name: 'Monitor', price: 300 },

];

const affordableProducts = products.filter((product) => product.price <= 800);

console.log(affordableProducts);

// Output: [{ name: 'Smartphone', price: 500 }, { name: 'Tablet', price: 800 }, { name: 'Monitor', price: 300 }]

Filtering out null and undefined values from an array:

const mixedValues = [10, 'hello', null, 25, undefined, 'world'];

const filteredValues = mixedValues.filter((value) => value !== null && value !== undefined);

console.log(filteredValues); // Output: [10, 'hello', 25, 'world']

Filtering numbers greater than a specific threshold:

const scores = [80, 65, 90, 50, 75, 95];

const passedScores = scores.filter((score) => score >= 70);

console.log(passedScores); // Output: [80, 90, 75, 95]

Filtering elements based on their data type:

const mixedData = [10, 'hello', true, null, 25, undefined, false];

const numbersOnly = mixedData.filter((value) => typeof value === 'number');

console.log(numbersOnly); // Output: [10, 25]

Filtering out duplicates from an array:

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

const uniqueValues = duplicates.filter((value, index, array) => array.indexOf(value) === index);

console.log(uniqueValues); // Output: [1, 2, 3, 4, 5]

Filtering objects based on multiple criteria:

const students = [

  { name: 'Alice', age: 22, grade: 'A' },

  { name: 'Bob', age: 19, grade: 'B' },

  { name: 'Charlie', age: 20, grade: 'A' },

  { name: 'Diana', age: 21, grade: 'A' },

];

const goodStudents = students.filter((student) => student.age >= 20 && student.grade === 'A');

console.log(goodStudents);

// Output: [{ name: 'Alice', age: 22, grade: 'A' }, { name: 'Diana', age: 21, grade: 'A' }]

Filtering out empty strings from an array:

const words = ['apple', '', 'banana', '', 'kiwi', ''];

const nonEmptyWords = words.filter((word) => word !== '');

console.log(nonEmptyWords); // Output: ['apple', 'banana', 'kiwi']

Filtering elements that satisfy a custom condition:

const temperatures = [10, 25, 30, 18, 15];

const comfortableTemps = temperatures.filter((temp) => temp >= 15 && temp <= 25);

console.log(comfortableTemps); // Output: [25, 18, 15]

Examples for map() method:

Mapping an array of numbers to their squares:

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

const squares = numbers.map((num) => num \* num);

console.log(squares); // Output: [1, 4, 9, 16, 25]

Mapping an array of strings to their lengths:

const words = ['apple', 'banana', 'kiwi', 'orange', 'grape'];

const wordLengths = words.map((word) => word.length);

console.log(wordLengths); // Output: [5, 6, 4, 6, 5]

Mapping an array of objects to extract specific properties:

const products = [

  { name: 'Laptop', price: 1000 },

  { name: 'Smartphone', price: 500 },

  { name: 'Tablet', price: 800 },

  { name: 'Monitor', price: 300 },

];

const productNames = products.map((product) => product.name);

console.log(productNames); // Output: ['Laptop', 'Smartphone', 'Tablet', 'Monitor']

Mapping an array of temperatures from Celsius to Fahrenheit:

const celsiusTemps = [0, 10, 20, 30, 40];

const fahrenheitTemps = celsiusTemps.map((tempCelsius) => (tempCelsius \* 9/5) + 32);

console.log(fahrenheitTemps); // Output: [32, 50, 68, 86, 104]

Mapping an array of strings to uppercase:

const fruits = ['apple', 'banana', 'kiwi', 'orange', 'grape'];

const capitalizedFruits = fruits.map((fruit) => fruit.toUpperCase());

console.log(capitalizedFruits); // Output: ['APPLE', 'BANANA', 'KIWI', 'ORANGE', 'GRAPE']

Mapping an array of numbers to strings with custom formatting:

const numbers = [100, 42, 789, 5, 300];

const formattedNumbers = numbers.map((num) => `Number: ${num}`);

console.log(formattedNumbers);

// Output: ['Number: 100', 'Number: 42', 'Number: 789', 'Number: 5', 'Number: 300']

Mapping an array of objects to a new shape:

const students = [

  { name: 'Alice', age: 22 },

  { name: 'Bob', age: 19 },

  { name: 'Charlie', age: 20 },

];

const studentSummaries = students.map((student) => ({

  fullName: student.name,

  status: student.age >= 20 ? 'Adult' : 'Teenager'

}));

console.log(studentSummaries);

/\* Output:

   [

     { fullName: 'Alice', status: 'Adult' },

     { fullName: 'Bob', status: 'Teenager' },

     { fullName: 'Charlie', status: 'Teenager' }

   ]

\*/

Mapping an array of boolean values to their negations:

const boolValues = [true, false, true, true, false];

const negatedValues = boolValues.map((value) => !value);

console.log(negatedValues); // Output: [false, true, false, false, true]

Mapping an array of numbers to strings with different units:

const distancesInMeters = [1000, 500, 200, 100];

const distancesInKilometers = distancesInMeters.map((distance) => `${distance / 1000} km`);

console.log(distancesInKilometers); // Output: ['1 km', '0.5 km', '0.2 km', '0.1 km']

Mapping an array of strings to their first characters:

const words = ['apple', 'banana', 'kiwi', 'orange', 'grape'];

const firstCharacters = words.map((word) => word.charAt(0));

console.log(firstCharacters); // Output: ['a', 'b', 'k', 'o', 'g']