|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **for** | **for…of** | **for…in** | **forEach** |
| **Usage** | General purpose loop | Iterating over iterable objects (arrays, strings, maps) | Iterating over object properties or array indices | Iterating over array elements |
| **Syntax** | for (let i = 0; i < length; i++) {} | for (let value of iterable) {  } | for (let key in object) {  } | array.forEach ((value) => {}); |
| **Access** | Accesses via index | Directly accesses values | Accesses property names/keys | Directly accesses values |
| **Break/Continue** | Supports **break** and **continue** | Supports **break** and **continue** | Supports **break** and **continue** | Does not support **break** and **continue** |
| **Return Value** | N/A | N/A | N/A | undefined |
| **Function Context** | Does not require a function | Does not require a function | Does not require a function | Requires a callback function |
| **Use Case** | Best for flexible and controlled iterations | Best for iterables like arrays, strings, maps | Best for iterating over object properties | Best for arrays, applying a function to each element |
| **Array Specific** | Works with any iterable if indexed manually | Works with any iterable object | Not recommended for arrays (iterates over indices) | Only works with arrays |

|  |  |
| --- | --- |
| for  let array = [10, 20, 30];  for (let i = 0; i < array.length; i++) {    console.log(array[i]);  }  // Output: 10, 20, 30 | forEach  let array = [10, 20, 30];  array.forEach((value) => {    console.log(value);  });  // Output: 10, 20, 30 |
| for…of  let array = [10, 20, 30];  for (let value of array) {    console.log(value);  }  // Output: 10, 20, 30 | for…in  let object = { a: 1, b: 2, c: 3 };  for (let key in object) {    console.log(key, object[key]);  }  // Output: a 1, b 2, c 3 |

|  |  |  |
| --- | --- | --- |
| **Feature** | **every** | **some** |
| **Usage** | Checks if **all elements** meet a condition | Checks if **at least one element** meets a condition |
| **Syntax** | array.every(callback); | array.some(callback); |
| **Break/Continue** | Stops checking as soon as a **false** is found | Stops checking as soon as a **true** is found |
| **Return Value** | **true** if **all elements** meet the condition, otherwise **false** | **true** if **at least one element** meets the condition, otherwise **false** |
| **Function Context** | Requires a callback function | |
| **Use Case** | Ensuring all items in a collection meet a certain criteria | Checking for the presence of at least one item that meets a criteria |

|  |  |
| --- | --- |
| // every  const users = [    { name: "Alice", isActive: true },    { name: "Bob", isActive: true },    { name: "Charlie", isActive: true },  ];  const allActive = users.every((user) => user.isActive);  console.log(allActive);  // Output: true  const products = [    { name: "Laptop", stock: 5 },    { name: "Phone", stock: 0 },    { name: "Tablet", stock: 10 },  ];  const allInStock = products.every((product) => product.stock > 0);  console.log(allInStock);  // Output: false | // some  const users = [    { name: "Alice", age: 25 },    { name: "Bob", age: 17 },    { name: "Charlie", age: 22 },  ];  const someUnderage = users.some((user) => user.age < 18);  console.log(someUnderage);  // Output: true  const products = [    { name: "Laptop", price: 899 },    { name: "Phone", price: 599 },    { name: "Headphones", price: 29 },  ];  const someAffordable = products.some((product) => product.price < 20);  console.log(someAffordable);  // Output: false |

const users = [

  { name: "Alice", age: 25 },

  { name: "Bob", age: 30 },

  { name: "Charlie", age: 35 },

];

function checkAll(array, condition) {

  return array.every(condition);

}

function checkSome(array, condition) {

  return array.some(condition);

}

// التحقق من أن جميع المستخدمين أكبر من 20 سنة

const allAdults = checkAll(users, (user) => user.age > 20);

console.log(allAdults); // true

// التحقق من أن واحدًا على الأقل من المستخدمين أكبر من 30 سنة

const hasSenior = checkSome(users, (user) => user.age > 30);

console.log(hasSenior); // true