One-Liners

JavaScript

Check if a string is an anagram of another string

```
const isAnagram = (str1, str2) =>
    [...str1.toLowerCase()].sort().join('')
                        === [...str2.toLowerCase()].sort().join('');

console.log(isAnagram('listen', 'silent'));
// Output: true
```

Merge two arrays and remove duplicates

```
const mergeArrays = (arr1, arr2) => [...new Set([...arr1, ...arr2])];

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

Generate an array of numbers from 1 to n

```
const generateArray = n => [...Array(n).keys()].map(i => i + 1);

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

Shuffle an array

```
const shuffleArray = arr => arr.sort(() => Math.random() - 0.5);

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

Find the maximum value in an array

```
const maxNumber = arr => Math.max(...arr);

console.log(maxNumber([ 26,9,45,18,37,12 ]))
// Output: 45
```

Get the current date in the format "YYYY-MM-DD"

```
const currentDate = () => new Date().toISOString().split('T')[0];

console.log(currentDate());
// Output: "2023-06-17"
```

Check if an object is empty

```
const isObjectEmpty = obj => Object.keys(obj).length === 0;

console.log(isObjectEmpty({})); // Output: true
```

Group an array of objects by a specific property

```
const groupByProperty = (arr, prop) => arr.reduce((result, obj) =>
    (result[obj[prop]] = [...result[obj[prop]] || [], obj], result), {});

const objArray = [{ id: 1, name: 'John' },{ id: 2, name:  'Jane'},
                  { id: 3, name: 'John'}]

console.log(groupByProperty(obj, 'name'));
// Output: { John: [ { id: 1, name: 'John' }, { id: 3, name: 'John' } ],
//           Jane: [ { id: 2, name: 'Jane' } ]}
```

Check if all elements in an array satisfy a condition

```javascript
const allElementsSatisfy = (arr,condition) => arr.every(condition)

console.log(allElementsSatisfy([2,4,6,8,10], num => num % 2 === 0))
// Output: true
```

Generate a range of numbers from start to end with a step

```javascript
const range = (start, end, step = 1) =>
   [...Array( Math.floor((end - start) / step) + 1)]
         .map((_, i) => start + (i * step));

console.log(range(1, 10, 2));
// Output: [1, 3, 5, 7, 9]
```

Find the average of an array of numbers

```
const calculateAverage = arr =>
          arr.reduce((sum, num) => sum + num, 0) / arr.length;

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

Deep clone an object

```
const deepClone = obj => JSON.parse(JSON.stringify(obj));

const originalObject = { name: 'Imtiyaz', age: 22 };
const clonedObject = deepClone(originalObject);
clonedObject.name = 'CodeClash';

console.log(originalObject); // Output: { name: 'Imtiyaz', age: 22 }
console.log(clonedObject); // Output: { name: 'CodeClash', age: 22 }
```

Conclusion

- JavaScript one-liners provide concise and efficient solutions for various tasks.
- They can be impressive but remember to prioritize code readability.
- Understanding and using these one-liners can enhance your JavaScript coding skills.
- As always, I hope you enjoyed the post and learned something new.
- If you have any queries then let me know in the comment box.