



@parth.webdev

Swipe →



JavaScript

# 5 PRO JavaScript Tricks you should know



Follow For More  
Web Development | Coding



# 1 Use Ternary Operator

## Noob:

```
let thirsty = false;
let drink;

if (thirsty === true) {
  drink = 'yes';
} else {
  drink = 'no';
}
```

## Pro:

```
let thirsty = false;
let drink = thirsty ? 'yes' : 'no';
```



## 2 Populating an array

Noob:

```
for (let i = 0; i < arraySize; i++) {  
    filledArray[i] = 'hello';  
}
```

Pro:

```
let filledArray = Array.from({length:10}, () => 'hello')
```

OUTPUT

```
['hello', 'hello', 'hello', 'hello', 'hello', 'hello', 'hello', 'hello', 'hello', 'hello']
```



@parth.webdev

### 3 Filter Unique Values

The Set object type was introduced in ES6, and along with ..., the 'spread' operator, we can use it to create a new array with only the unique values.

```
const arrayWithDuplicates = [1, 2, 3, 2, 4, 1, 5, 6, 3];  
  
const uniqueArray = [...new Set(arrayWithDuplicates)];  
  
console.log(uniqueArray); // Output: [1, 2, 3, 4, 5, 6]
```





## 4 Convert Any Value to Boolean

### 1. Converting truthy values to true and falsy values to false

```
Boolean(true);      // Output: true
Boolean(1);          // Output: true
Boolean("hello");    // Output: true
Boolean([]);         // Output: true

Boolean(false);      // Output: false
Boolean(0);          // Output: false
Boolean("");         // Output: false
Boolean(undefined); // Output: false
```

### 2. Using double negation !! to achieve the same result

```
console.log(!!true);      // Output: true
console.log(!!1);         // Output: true
console.log(!!"hello");   // Output: true

console.log(!!false);     // Output: false
console.log(!!"");        // Output: false
console.log(!!null);      // Output: false
```



## 5 Objects to Arrays

1. **Object.keys()**: This method gets all the keys from an object and puts them into an array.
2. **Object.values()**: This method retrieves all the values from an object and puts them into an array.
3. **Object.entries()**: This method combines both keys and values as pairs, creating arrays of pairs from an object's properties.

```
const obj = {
  key1: 'value1',
  key2: 'value2',
  key3: 'value3'
};

const keysArray = Object.keys(obj);
const valuesArray = Object.values(obj);
const entriesArray = Object.entries(obj);

console.log(keysArray);    // Output: ['key1', 'key2', 'key3']
console.log(valuesArray);  // Output: ['value1', 'value2', 'value3']
console.log(entriesArray);
// Output: [['key1', 'value1'], ['key2', 'value2'], ['key3', 'value3']]
```

