

*Follow @asifaliquraishy*

# 5 Super Useful JS CheatSheet



save  
post



```
1 [1, 2, 3].length; // 3
2 [1, 2, 3].push(4); // [1, 2, 3, 4] *
3 [1, 2, 3].unshift(0); // [0, 1, 2, 3] *
4 [1, 2, 3].pop(); // [1, 2]
5 [1, 2, 3].shift(); // [2, 3] *
6 [1, 2, 3].at(2); // 3
7 [1, 2, 3].indexOf(3); // 2
8 [1, 2, 3].includes(3); // true
9 [1, 2, 3].map((num) => num * 2); // [2, 4, 6]
10 [1, 2, 3].filter((num) => num > 1); // [2, 3]
11 [1, 2, 3].every((num) => num > 0); // true
12 [1, 2, 3].some((num) => num > 2); // true
13 [1, 2, 3].fill(0); // [0, 0, 0]
14 [1, 2, 3].reduce((acc, num) => acc + num, 0); // 6
15 [1, 2, 3].concat([4, 5]); // [1, 2, 3, 4, 5]
16 [1, 2, 3].reverse(); // [3, 2, 1]
17 [3, 2, 1].sort(); // [1, 2, 3]
18 [1, 2, 3].join("-"); // "1-2-3"
19 [1, 2, 3].flat(); // [1, 2, 3]
20 [1, 2, 3].find((num) => num === 1); // 1
21 [1, 2, 3].findIndex((num) => num === 2); // 1
22 [1, 2, 3].toString(); // "1,2,3"
23 [1, 2, 3].toLocaleString(); // "1,2,3"
24 [1, 2, 3].slice(1, 2); // [2]
25 [1, 2, 3].splice(1, 1, "a"); // [1, 'a', 3]
26 Array.isArray([1, 2, 3]); // true
27 Array.from("123"); // ['1', '2', '3']
```



## 2.Array Methods



```
1 "JavaScript".length; //? 10
2 "JavaScript"[2]; //? 'v'
3 "JavaScript".charAt(2); //? 'v'
4 "JavaScript".charCodeAt(2); //? 118
5 "JavaScript".indexOf("S"); //? 4
6 "JavaScript".toLowerCase(); //? 'javascript'
7 "JavaScript".toUpperCase(); //? 'JAVASCRIPT'
8 "JavaScript".slice(2, 5); //? 'vaS'
9 "JavaScript".substring(2, 5); //? 'vaS'
10 "JavaScript".substr(2, 2); //? 'va'
11 "JavaScript".concat(" Dev"); //? 'JavaScript Dev'
12 "JavaScript Dev".split(" "); //? ['JavaScript', 'Dev']
13 "JavaScript Dev".includes("Dev"); //? true
14 "Java Dev".replace("Dev", "JS"); //? 'Java JS'
15 "Java Dev".replaceAll("Dev", "JS"); //? 'Java JS'
16 " JavaScript Dev ".trim(); //? 'JavaScript Dev'
17 " JavaScript Dev ".trimStart(); //? 'JavaScript Dev '
18 " JavaScript Dev ".trimEnd(); //? ' JavaScript Dev '
19 "Dev".padStart(10, "*"); //? '*****Dev'
20 "Dev".padEnd(10, "*"); //? 'Dev*****'
21 "JavaScript Dev".startsWith("Java"); //? true
22 "JavaScript Dev".endsWith("Dev"); //? true
23 "JavaScript Dev".repeat(3); //? 'JavaScript DevJavaScript DevJavaScript Dev'
24 "JavaScript Dev".indexOf("JavaScript"); //? 0
25 "JavaScript Dev".lastIndexOf("Dev"); //? 11
26 "JavaScript Dev".search("Dev"); //? 11
27 "JavaScript Dev".includes("Dev"); //? true
```





Tech\_sanku

# 3.Date Methods

## Initialization



```
1 const date = new Date();
2 //? 2023-04-25T19:27:35.363Z
3 new Date(1682452894553);
4 //? Sun Apr 25 2023 00:00:00 GMT+0200
5 new Date("2023-04-25");
6 //? Sun Apr 25 2023 00:00:00 GMT+0200
7 new Date("2023-04-25T01:10:00");
8 //? Sun Apr 25 2023 01:10:00 GMT+0200
9 new Date(2023, 3, 25, 1, 10, 0, 0);
10 //? Tue Apr 25 2023 01:10:00 GMT+0200
11 // year, month, day, hour, min, sec, misc
```



save  
post

swipe  
to know





Tech\_sanku

# Conversions



```
1 date.toString();
2 //? Tue Apr 25 2023 21:27:35 GMT+0200 (Central European Summer Time)
3 date.toDateString();
4 //? Tue Apr 25 2023
5 date.toTimeString();
6 //? 21:27:35 GMT+0200 (Central European Summer Time)
7 date.toISOString();
8 //? 2023-04-25T19:27:35.363Z
9 date.toLocaleString();
10 //? 4/25/2023, 9:27:35 PM
11 date.toLocaleDateString();
12 //? 4/25/2023
13 date.toLocaleTimeString();
14 //? 9:27:35 PM
15 date.getTime();
16 //? 1682452895363
```



save  
post

swipe  
to know



Tech\_sanku

## Get Methods



```
1 date.getFullYear();           //? 2023
2 date.getMonth();             //? 3
3 date.getDate();              //? 25
4 date.getDay();               //? 2
5 date.getHours();             //? 21
6 date.getMinutes();           //? 27
7 date.getSeconds();           //? 35
8 date.getMilliseconds();      //? 363
9 date.getTime();              //? 1682452895363
10 date.getTimezoneOffset();    //? -120
```

## Set Methods



```
1 date.setFullYear(2023);      //? set year
2 date.setMonth(11);            //? set month
3 date.setDate(1);             //? set date
4 date.setHours(10);           //? set hours
5 date.setMinutes(20);          //? set minutes
6 date.setSeconds(20);          //? set seconds
7 date.setMilliseconds(20);     //? set milliseconds
8 date.setTime(1680153156131); //? set time (milliseconds since Jan 1, 1970)
```



save  
post

swipe  
to know



Tech\_sanku

# 4.DOM Methods

## Accessing Elements



```
1 document.getElementById("id"); //? find element by its id
2 document.getElementsByClassName("class"); //? find elements by class
3 document.getElementsByTagName("tag"); //? find elements by tag name
4 document.querySelector("selector"); //? find first element matching selector
5 document.querySelectorAll("selector"); //? find all elements matching selector
```

## Creating/Appending Elements



```
1 document.createElement("name"); //? create element node
2 document.createTextNode("text"); //? create text node
3 elem.appendChild(child); //? append child to element
4 elem.removeChild(child); //? remove child from element
5 elem.replaceChild(newChild, oldChild); //? replace child with new child
```

## Modifying Elements



```
1 elem.innerHTML = "<h2>outerHTML</h2>"; //? set HTML
2 elem.innerText = "inner text"; //? set inner text
3 elem.textContent = "text content"; //? set text content
4 elem.style.color = "blue"; //? set style
5 elem.outerHTML = "<p>Learn with <strong>Athreos</strong></p>"; //? replace HTML
```



save  
post

swipe  
to know





Tech\_sanku

## Accessing Parent, Children, Siblings



```
1 elem.parentElement; //? access parent element  
2 elem.children; //? access element children  
3 elem.firstElementChild; //? access first child  
4 elem.lastElementChild; //? access last child  
5 elem.nextElementSibling; //? access next sibling  
6 elem.previousElementSibling; //? access previous sibling
```

## Modifying Attributes



```
1 elem.getAttribute("attr"); //? get attribute value  
2 elem.setAttribute("attr", "value"); //? set attribute value  
3 elem.removeAttribute("attr"); //? remove attribute
```

## Modifying Element Classes



```
1 elem.classList.add("my-class"); //? add class  
2 elem.classList.remove("my-class"); //? remove class  
3 elem.classList.toggle("my-class"); //? toggle class  
4 elem.classList.contains("my-class"); //? check for class
```



save  
post

swipe  
to know



Tech\_sanku

# 5.DOM Events

## Event Listeners



```
1 document.addEventListener('click', (event) => {  
2     console.log('Click Event', event);  
3});  
4  
5 // unregister event listener  
6 document.removeEventListener('click', (event) => {  
7     console.log('Unregistered Event', event);  
8});
```

### Keyboard Events

key is pressed down

**keydown**

key is released

**keyup**

### Form Events

**blur**

element has lost focus

**change**

use modified value of <input>, <select>  
or <textarea>

**focus**

element has received focus

**select**

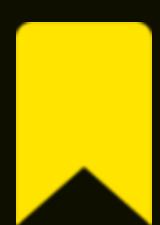
text has been selected in an element

**submit**

fires on <form> when submitted

**reset**

fires on form is reset



save  
post

swipe  
to know





## Mouse Events

**click** left mouse button click

**dblclick** left mouse button double click

**mousedown** pointing device button is pressed when inside element

**mouseup** mouse button released over an element

**mouseover** mouse button enter an element

**mousemove** mouse pointer moves over on elememt

## Window Events

**abort** resource was not fully loaded, but not due an error

**error** error event occurs if resource failed to load or can't be used

**load** document has finished loading

**unload** document has being unloaded

**scroll** document is scrolled

**resize** window is resized



save  
post

swipe  
to know

