



learning.atheros.ai - Modern Frontend Developer (Link in BIO)



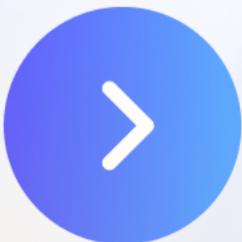
55% OFF now

Learn JavaScript

# 5 Super Useful JS Cheatsheets



David Mráz  
@davidm\_ai





# Array Methods Cheatsheet

```
● ● ● ⏪ ⏹ Array Methods Cheatsheet ⏷ + ⏺  
[1,2,3].length; //? 3  
[1,2,3].push(4); //? [1,2,3,4] *  
[1,2,3].unshift(0); //? [0,1,2,3] *  
[1,2,3].pop(); //? [1,2] *  
[1,2,3].shift(); //? [2,3] *  
[1,2,3].at(2); //? 3  
[1,2,3].indexof(3); //? 2  
[1,2,3].includes(3); //? true  
[1,2,3].map((num) => Math.pow(num, 2)); //? [1, 4, 9]  
[1,2,3].filter((num) => num % 2); //? [1,3]  
[1,2,3].every((num) => num > 1); //? false  
[1,2,3].some((num) => num == 3); //? true  
[1,2,3].fill(10); //? [10,10,10]  
[1,2,3].reduce((acc, num) => acc + num, 0); //? 6  
[1,2,3].concat([4,5]); //? [1,2,3,4,5]  
[1,2,3].reverse(); //? [3,2,1]  
[2,1,3].sort(); //? [1,2,3]  
[1,2,3].join('-'); //? 1-2-3  
[1,2,[3]].flat(); //? [1,2,3]  
[1,2,3].find((num, i) => i === 1); //? 2  
[1,2,3].findIndex((num) => num === 2); //? 1  
[1,2,3].toString(); //? 1,2,3  
[1,2,3].slice(1,3); //? [2,3]  
[1,4].splice(1, 0, 2, 3); //? [1,2,3,4] *  
Array.isArray("[1,2,3]"); //? false  
Array.from("123"); //? ['1','2','3']
```



# String Methods Cheatsheet

```
● ● ● ⏷ < > String Methods Cheat Sheet ⏵ + ⏻

'JavaScript'.length; //? 10
'JavaScript'[2]; //? 'v'
'JavaScript'.charAt(2); //? 'v'
'JavaScript'.charCodeAt(2); //? 118
'JavaScript'.indexOf('S'); //? 4
'JavaScript'.toLowerCase(); //? 'javascript'
'JavaScript'.toUpperCase(); //? 'JAVASCRIPT'
'JavaScript'.slice(2, -5); //? 'vaS'
'JavaScript'.substring(-2, 2); //? 'Ja'
'JavaScript'.concat(' Dev'); //? 'JavaScript Dev'
'JavaScript Dev'.split(' '); //? ['JavaScript', 'Dev']
'JavaScript Dev'.includes('Dev'); //? true
'Dev Dev'.replace('Dev', 'JS'); //? 'JS Dev'
'Dev Dev'.replaceAll('Dev', 'JS');//? 'JS JS'
' JavaScript Dev '.trim(); //? 'JavaScript Dev'
' JavaScript Dev '.trimStart(); //? 'JavaScript Dev '
' JavaScript Dev '.trimEnd(); //? ' JavaScript Dev '
'Dev'.padStart(10, '*'); //? '*****Dev
'Dev'.padEnd(10, '*'); //? 'Dev*****'
'JavaScript Dev'.startsWith('J'); //? true
'JavaScript Dev'.endsWith('v'); //? true
'Dev'.repeat(3); //? DevDevDev
'JavaScript Dev'.indexOf('JavaScript'); //? 0
'JavaScript Dev'.lastIndexOf('Dev'); //? 11
'JavaScript Dev'.search('Dev'); //? 11
'JavaScript Dev'.includes('Dev'); //? true
```





# Date Methods Cheatsheet

## Initialization JS

```
const date = new Date();
//? 2023-04-25T19:27:35.363Z
new Date(1682452849453);
new Date('2023-04-25');
new Date("2023-04-25T01:10:00");
new Date(2023, 3, 25, 1, 0, 0, 0);
```

\*dates recalculated for values out of range

year	month	day	hour	min	sec	mlsec
(0-11)	(1-31)	(0-23)	(0-59)	(0-59)	(0-999)	

## Conversions JS

```
date.toString();
//? Tue Apr 25 2023 21:27:35 GMT+0200
date.toTimeString(); //? 21:27:35 GMT+0200
date.toDateString(); //? Tue Apr 25 2023
//? 2023-04-25T19:27:35.363Z
date.toISOString();
date.toLocaleString(); //? 9:27:35 PM
date.toLocaleTimeString(); //? 9:27:35 PM
date.getTime(); //? 1682450855363
```

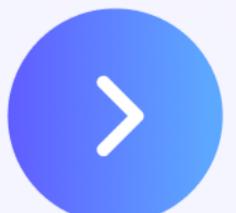
## Get Methods JS

```
date.getFullYear(); //? 2023
date.getMonth(); //? 3
date.getDate(); //? 25
date.getDay(); //? 2
date.getHours(); //? 21
date.getMinutes(); //? 27
date.getSeconds(); //? 35
date.getMilliseconds(); //? 363
date.getTime(); //? 1682450855363
date.getTimezoneOffset(); //? -120
```

## Set Methods JS

```
date.setFullYear(2023); //? set year
date.setMonth(11); //? set month
date.setDate(1); //? set date
date.setHours(20); //? set hours
date.setMinutes(20); //? set minutes
date.setSeconds(20); //? set seconds
date.setMilliseconds(20); //? set ms
//? number of ms from Jan 1st, 1970
date.setTime(1681051613351);
```

\*for more complex scenario, consider using <https://date-fns.org/>





# DOM Methods Cheatsheet

**Accessing Elements**

```
//? find element by its id  
document.getElementById("id");  
//? find element by its class  
document.getElementsByClassName("class");  
//? find element by its name attribute  
document.getElementsByName("name");  
//? find element by its tag  
document.getElementsByTagName("tag");  
//? find first element matching selector  
document.querySelector("selector");  
//? find all elements matching selector  
document.querySelectorAll('selector');
```

**Creating/Appending Elements**

```
//? create element node  
document.createElement('name');  
//? create text node  
document.createTextNode('text');  
//? append child for elem  
elem.appendChild(child);  
//? remove child for elem  
elem.removeChild(child);  
//? replace child with new  
elem.replaceChild(newChild, oldChild);
```

**Modifying Elements**

```
//? modifying element  
elem.outerHTML = '<h2>OuterHTML</h2>';  
elem.innerHTML = 'New <strong>inner</strong>';  
//? modify text content  
elem.textContent = 'New.textContent';  
//? modify style  
elem.style = 'color: blue;';  
  
outerHTML  
  
<p>Learn with <strong>Atheros</strong></p>  
textContent  
  
innerHTML
```

**Accessing Parent, Children, Siblings**

```
//? access element children  
elem.children; elem.childNodes;  
//? get first child  
elem.firstChildElement; elem.firstChild;  
//? get last child  
elem.lastChildElement; elem.lastChild;  
//? get parent  
child.parentNode;  
//? previous sibling  
child.previousSibling;  
//? get next sibling  
child.nextSibling;  
  
//? node info  
child.nodeName;  
child.nodeType;  
child.nodeValue;
```

**Modifying Attributes**

```
//? get current attribute value  
elem.getAttribute('title');  
//? set attribute with new value  
elem.setAttribute('title', 'new title');  
//? is attribute on element true/false  
elem.hasAttribute('title');  
//? remove attribute from element  
elem.removeAttribute('title');
```

**Modifying Element Classes**

```
elem.classList //? access classes as a list  
elem.classList.add('my-class'); //? add class  
elem.classList.remove('my-class'); //? remove class  
//? add/remove class if present  
elem.classList.toggle('my-class');  
//? is class applied to element? true/false  
elem.classList.contains('my-class');
```





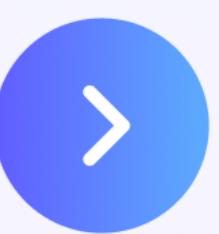
# DOM Events Cheatsheet

The screenshot shows a dark-themed cheatsheet titled "DOM Events Cheatsheet". The interface is organized into several sections:

- Event Listeners**: A code snippet demonstrating event registration and unregistration:

```
//? register event listener
document.addEventListener('click', (event) => {
  console.log('Click Event', event);
});

//? unregister event listener
document.removeEventListener('click', (event) => {
  console.log('Unregistered Event', event);
});
```
- Keyboard Events**:
  - keydown**: key is pressed down
  - keyup**: key is released
- Form Events**:
  - blur**: element has lost focus
  - change**: user modifies 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 when form is reset
- 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 is released over an element
  - mouseover**: mouse pointer enters an element
  - mousemove**: mouse pointer moves over an element
- Window Events**:
  - abort**: resource was not fully loaded, but not due to an error
  - error**: error event occurs if resource failed to load or can't be used
  - load**: document has finished loading
  - unload**: document is being unloaded
  - scroll**: document is scrolled
  - resize**: window is resized



# Become a modern front-end developer with my new ebook available now!



[learning.atheros.ai](https://learning.atheros.ai)

[Link in BIO](#)



Grab 55% early bird discount now!



**Modern Frontend Developer:  
HTML and CSS**



Learn how to implement smooth and responsive UI!



David Mraz & Atheros Learning

v1

