



## JavaScript Cheatsheet



Haris Ali Khan

May 11, 2024 7 min read

### JavaScript Basics

Set of JavaScript basic syntax to add, execute and write basic [programming](#) paradigms in Javascript

[Learn computer programming](#)

### On Page Script

Adding internal JavaScript to HTML

```
<script type="text/javascript"> //JS code goes here </script>
```

### External JS File

Adding external JavaScript to HTML

```
<script src="filename.js"></script>
```

### Functions

JavaScript Function syntax

```
function nameOfFunction() {  
    // function body  
}
```

### DOM Element

Changing content of a DOM Element

```
document.getElementById("elementID").innerHTML = "Hello World!";
```

### Output

This will print the value of a in JavaScript console

```
console.log(a);
```

### Conditional Statements

Conditional statements are used to perform operations based on some conditions.

**PRINT PDF**

3 Easy Steps:  
1. Click "Print PDF"  
2. Add on our Website  
3. View & Read PDF

```
if (condition) {  
    // block of code to be executed if the condition is true  
}
```

## If-else Statement

If the condition for the if block is false, then the else block will be executed.

```
if (condition) {  
    // block of code to be executed if the condition is true  
} else {  
    // block of code to be executed if the condition is false  
}
```

## Else-if Statement

A basic if-else ladder

```
if (condition1) {  
    // block of code to be executed if condition1 is true  
} else if (condition2) {  
    // block of code to be executed if the condition1 is false and condition2 is true  
} else {  
    // block of code to be executed if the condition1 is false and condition2 is false  
}
```

## Switch Statement

Switch case statement in JavaScript

```
switch (expression) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block  
}
```

**VIEW PDF**

3 Easy Steps:

1. Click "View PDF"
2. Add on our Website
3. View & Read PDF

GoPD

## For Loop

For loop syntax in javascript

```
for (initialization; condition; incrementation;) {  
    // code block to be executed  
}
```

Example:

```
for (let i = 0; i < 5; i++) {  
    text += "Iteration number: " + i + "<br>";  
}
```

## While Loop

Runs the code till the specified condition is true

```
while (condition) {  
    // code block to be executed  
}
```

## Do While Loop

A do while loop is executed at least once despite the condition being true or false

```
do {  
    // run this code in block  
    i++;  
} while (condition);
```

## Strings

The string is a sequence of characters that is used for storing and managing text data.

### charAt method

Returns the character from the specified index.

```
str.charAt(3)
```

### concat method

Joins two or more strings together.

```
str1.concat(str2)
```

### indexOf method

Returns the index of the first occurrence of the specified character from the string else -1 if not found.

```
str.indexOf('substr')
```



```
str.match(/(chapter \d+(\.\d+)*)/i;)
```

## replace method

Searches a string for a match against a specified string or char and returns a new string by replacing the specified values.

```
str1.replace(str2)
```

## search method

Searches a string against a specified value.

```
str.search('term')
```

## split method

Splits a string into an array consisting of substrings.

```
str.split('\n')
```

## substring method

Returns a substring of a string containing characters from the specified indices.

```
str.substring(0,5)
```

# Arrays

The array is a collection of data items of the same type. In simple terms, it is a variable that contains multiple values.

## variable

Containers for storing data.

```
var fruit = ["element1", "element2", "element3"];
```

## concat method

## indexOf method

Returns the index of the specified item from the array.

```
indexOf()
```

## join method

Converts the array elements to a string.

```
join()
```

## pop method

Deletes the last element of the array.

```
pop()
```

## reverse method

This method reverses the order of the array elements.

```
reverse()
```

## sort method

Sorts the array elements in a specified manner.

```
sort()
```

## toString method

Converts the array elements to a string.

```
toString()
```

## valueOf method

returns the relevant Number Object holding the value of the argument passed

```
valueOf()
```

## Number Methods

JS math and number objects provide several constant and methods to perform mathematical operations.

## toExponential method

Converts a number to its exponential form.

```
toExponential()
```



```
toFixed()
toPrecision()
```

## toString method

Converts an object to a string

```
toString()
```

## valueOf method

Returns the primitive value of a number.

```
valueOf()
```

## Maths Methods

### ceil method

Rounds a number upwards to the nearest integer, and returns the result

```
ceil(x)
```

### exp method

Returns the value of  $E^x$ .

```
exp(x)
```

### log method

Returns the logarithmic value of x.

```
log(x)
```

### pow method

Returns the value of x to the power y.

```
pow(x,y)
```

### random method

Returns a random number between 0 and 1.

```
random()
```

### sqrt method

Returns the square root of a number x



## Dates

Date object is used to get the year, month and day. It has methods to get and set day, month, year, hour, minute, and seconds.

### Pulling Date from the Date object

Returns the date from the date object

```
getDate()
```

### Pulling Day from the Date object

Returns the day from the date object

```
getDay()
```

### Pulling Hours from the Date object

Returns the hours from the date object

```
getHours()
```

### Pulling Minutes from the Date object

Returns the minutes from the date object

```
getMinutes()
```

### Pulling Seconds from the Date object

Returns the seconds from the date object

```
getSeconds()
```

### Pulling Time from the Date object

Returns the time from the date object

```
getTime()
```



## click

Fired when an element is clicked

```
element.addEventListener('click', () => {  
  // Code to be executed when the event is fired  
});
```

## oncontextmenu

Fired when an element is right-clicked

```
element.addEventListener('contextmenu', () => {  
  // Code to be executed when the event is fired  
});
```

## dblclick

Fired when an element is double-clicked

```
element.addEventListener('dblclick', () => {  
  // Code to be executed when the event is fired  
});
```

## mouseenter

Fired when an element is entered by the mouse arrow

```
element.addEventListener('mouseenter', () => {  
  // Code to be executed when the event is fired  
});
```

## mouseleave

Fired when an element is exited by the mouse arrow

```
element.addEventListener('mouseleave', () => {  
  // Code to be executed when the event is fired  
});
```

## mousemove

Fired when the mouse is moved inside the element

```
element.addEventListener('mousemove', () => {  
  // Code to be executed when the event is fired  
});
```

## Keyboard Events

### keydown





```
});
```

## keypress

Fired when the user presses the key on the keyboard

```
element.addEventListener('keypress', () => {  
    // Code to be executed when the event is fired  
});
```

## keyup

Fired when the user releases a key on the keyboard

```
element.addEventListener('keyup', () => {  
    // Code to be executed when the event is fired  
});
```

## Errors

Errors are thrown by the [JS compiler](#) or interpreter whenever they find any fault in the code, and it can be of any type like syntax error, run-time error, logical error, etc. JS provides some functions to handle the errors.

## try and catch

Try the code block and execute catch when err is thrown

```
try {  
    Block of code to try  
}  
catch (err) {  
    Block of code to handle errors  
}
```

## Window Methods

Methods that are available from the window object

### alert method

Used to alert something on the screen

The `blur()` method removes focus from the current window.

```
blur()
```

## setInterval

Keeps executing code at a certain interval

```
setInterval(() => {  
    // Code to be executed  
}, 1000);
```

## setTimeout

Executes the code after a certain interval of time

```
setTimeout(() => {  
    // Code to be executed  
}, 1000);
```

## close

The `Window.close()` method closes the current window

```
window.close()
```

## confirm

The `window.confirm()` instructs the browser to display a dialog with an optional message, and to wait until the user either confirms or cancels

```
window.confirm('Are you sure?')
```

## open

Opens a new window

```
window.open("https://www.codewithharry.com");
```

## prompt

Prompts the user with a text and takes a value. Second parameter is the default value

```
var name = prompt("What is your name?", "Harry");
```

## scrollBy

```
window.scrollBy(100, 0); // Scroll 100px to the right
```



```
window.scrollTo(500, 0); // Scroll to horizontal position 500
```

## clearInterval

Clears the setInterval. var is the value returned by setInterval call

```
clearInterval(var)
```

## clearTimeout

Clears the setTimeout. var is the value returned by setTimeout call

```
clearTimeout(var)
```

## stop

Stops the further resource loading

```
stop()
```

## Query/Get Elements

The browser creates a DOM (Document Object Model) whenever a web page is loaded, and with the help of HTML DOM, one can access and modify all the elements of the HTML document.

### querySelector

Selector to select first matching element

```
document.querySelector('css-selectors')
```

### querySelectorAll

A selector to select all matching elements

```
document.querySelectorAll('css-selectors', ...)
```

### getElementsByName

## getElementsByClassName

Select elements by class name

```
document.getElementsByClassName('class-name')
```

## Get Element by Id

Select an element by its id

```
document.getElementById('id')
```

## Creating Elements

Create new elements in the DOM

### createElement

Create a new element

```
document.createElement('div')
```

### createTextNode

Create a new text [🔗 node](#)

```
document.createTextNode('some text here')
```

[Download this Cheatsheet](#)

### Add a new comment

Post Comment

## Comments (79)



**abdurrehmans059** 2024-10-23

Thanks Harry bhai. One adjustment that can be done is that the text in "Download this cheat sheet" button is hard to read in light mode.

REPLY

REPLY

**tayyabrasheed0121\_gm** 2024-09-23

Hello! This is a amazing step to help beginners in coding. Thank you so much Harry bhai ! [🔗 Learn computer programming](#)

REPLY

**thepoplechoice21\_gm** 2024-09-23

Bhaiyaa react Js ke notes bhi upload kar do > Bahut hi jyda help ho jayegi humari . ❤️

REPLY

**devin8659\_gm** 2024-09-12

bas kro harry bhai ab Rulaoge kya ❤️ , big fan one i will meet you

REPLY

🔍 This site uses Google AdSense ad intent links. AdSense automatically generates these links and they may help creators earn money.

**CodeWithHarry**

Copyright © 2024 CodeWithHarry.com

