JavaScript Notes .

💻 JavaScript Notes

Welcome to my personal JavaScript notebook!

Here you'll store methods, functions, and examples that you discover while coding.

🔤 1- STRING METHODS

Method	Description	Example	MDN Link
`.slice()`	Cuts a part of a string	`"hello".slice(1, 3)` → `"el"`	MDN – slice()
`.startsWith()`	Checks if a string starts with something	`"New! hello".startsWith("New!")` → `true`	MDN – startsWith()
`.charCodeAt()`	Returns the Unicode value of a character	`"A".charCodeAt(0)` → `65`	MDN – charCodeAt()
`String.fromCharCode()`	Converts Unicode to character	`String.fromCharCode(66)` → `"B"`	MDN – fromCharCode()
`.padStart()`	Adds padding to start of a string	`"5".padStart(2, "0")` → `"05"`	MDN – padStart()
`.split()`	Splits string into array based on separator	`"a,b,c".split(",")` → `["a","b","c"]`	MDN – split()
`.split("\n")`	Splits string by line breaks	`"a\nb\nc".split("\n")` → `["a","b","c"]`	MDN – split()
`.indexOf()`	Returns first index where character appears (-1 if not found)	`"hello".indexOf("l")` → `2`	MDN – indexOf()
`.lastIndexOf()`	Returns last index where character appears	`"hello".lastIndexOf("l")` → `3`	MDN – lastIndexOf()
`.includes()`	Checks if string contains substring	`"hello".includes("ll")` → `true`	MDN – includes()
`length`	Returns the number of	`"Hello".length` → `5`	MDN – length

Method	Description	Example	MDN Link
	characters in a string		
Counting occurrences	Counts how many times a character appears in a string	`countChar("hello","l")` → 2	-
`String(number)`	Converts number to string to access digits	`String(123)` → `"123"`	MDN – String()
Access characters by index	Access a character in a string	`"123"[1]` → `"2"`	MDN – String.prototype

📅 2- DATE METHODS

Method	Description	Example	MDN Link
`new Date()`	Creates a date object	`new Date()`	MDN – Date()
`.getDate()`	Returns the day of the month	`new Date().getDate()`	MDN – getDate()
`.getMonth()`	Returns month (0–11)	`new Date().getMonth() + 1`	MDN – getMonth()
`.getFullYear()`	Returns 4-digit year	`new Date().getFullYear()`	MDN – getFullYear()
`.getDay()`	Returns day of week (0–6)	`new Date().getDay()` → `3`	MDN – getDay()
`.toLocaleString()`	Formats date/time according to locale	`new Date().toLocaleString("en-US")`	MDN – toLocaleString()

🔢 3- NUMBER & MATH METHODS

Method	Description	Example	MDN Link
`Math.random()`	Generates random number between 0 and 0.999...	`Math.random()` → `0.734`	MDN – Math.random()
`Math.floor()`	Rounds DOWN to nearest integer	`Math.floor(4.7)` → `4`	MDN – Math.floor()
`.toString(base)`	Converts number to string in specified	`15.toString(16)` → `"f"` (hexadecimal)	MDN – toString()

Method	Description	Example	MDN Link
	base		
Random range pattern	Generate random integer in range	Math.floor(Math.random() * 16) → 0-15	-

⚙️ 4- OPERATORS & SYNTAX

Concept	Description	Example
==	Loose equality (converts types before comparing)	5 == "5" → true
===	Strict equality (checks type and value)	5 === "5" → false
!==	Strict inequality (checks if not equal in type/value)	5 !== "5" → true
+	Adds numbers or joins strings	"Hello " + "World"
-	Subtraction	10 - 3 → 7
/	Division	10 / 2 → 5
<	Less than comparison	5 < 10 → true
>=	Greater than or equal to	amount >= coin → true/false
++	Increment by 1	let x = 0; x++ → 1
-=	Subtract and assign	amount -= coin → subtracts coin from amount
+=	Add and assign (concatenate for strings)	str += char → appends char to str
⇒	Arrow function syntax	const add = (a,b) ⇒ a + b;
${}	Template literals (insert variables)	Hello ${name}
!	Logical NOT (reverses boolean)	!true → false
&& / `		`
%	Modulo operator (remainder)	4 % 2 == 0 → true
Counting pattern	Use variable to accumulate results in loop	let count=0; count++
Tracking max/min	Store largest/smallest value in loop	let max=arr[0]; if(arr[i]>max) max=arr[i];
Nested if-else	Multiple conditions within conditions	if(x){if(y){...}else{...}}

Concept	Description	Example
Early return pattern	Return immediately when condition is met	`if(condition) return result;`

🧱 5- ARRAY METHODS

Method	Description	Example	MDN Link
`.map()`	Creates a new array by applying a function to each element	`[1,2,3].map(n ⇒ n*2)` → `[2,4,6]`	MDN – map()
`.filter()`	Keeps elements that pass a test	`[1,2,3,4].filter(n ⇒ n%2===0)` → `[2,4]`	MDN – filter()
`.forEach()`	Runs a function on each element	`[1,2,3].forEach(n ⇒ console.log(n))`	MDN – forEach()
`.find()`	Returns first element matching condition	`[3,6,9].find(n⇒n>5)` → `6`	MDN – find()
`.reduce()`	Reduces array to single value	`[1,2,3].reduce((a,b)⇒a+b,0)` → `6`	MDN – reduce()
`.sort()`	Sorts array elements	`[3,1,2].sort()` → `[1,2,3]`	MDN – sort()
`.push()` / `.pop()`	Add/remove from end	`arr.push(4)` / `arr.pop()`	MDN – push()
`.shift()` / `.unshift()`	Remove/add from start	`arr.shift()` / `arr.unshift(0)`	MDN – shift()
`.every()`	Tests if ALL elements pass condition	`[2,4,6].every(n⇒n%2===0)` → true	MDN – every()
`.join()`	Joins array elements into string	`["a","b","c"].join("")` → `"abc"`	MDN – join()
`.includes()`	Checks if array contains element	`[1,2,3].includes(2)` → `true`	MDN – includes()
`Array.from()`	Creates array from object with length	`Array.from({length: 3})` → `[undefined, undefined, undefined]`	MDN – Array.from()
`array.push()`	Add element dynamically	`secondArray.push(j)`	MDN – push()
`slice(start, end)`	Get portion of array	`arrays.slice(1)` → all except first element	MDN – slice()

Method	Description	Example	MDN Link
Spread operator `...`	Spread elements of array into new array	`[newStr, ...arrays.slice(1)]`	MDN – spread operator

🎯 6- OBJECT METHODS

Method	Description	Example	MDN Link
`Object.keys()`	Returns array of object's property NAMES	`Object.keys({a:1, b:2})` → `["a","b"]`	MDN – Object.keys()
`.hasOwnProperty()`	Checks if object has specific property	`obj.hasOwnProperty("name")` → true/false	MDN – hasOwnProperty()
`obj[key]`	Access object property value using variable	`obj["name"]` → accesses obj.name	MDN – Property accessors
`.length` on Object.keys()	Get number of properties in object	`Object.keys(obj).length` → 3	-
Compare objects pattern	Check if two objects are equal	`Object.keys(a).every(k => a[k] === b[k])`	-

🔁 7- LOOPS & NESTED LOOPS

Concept	Description	Example
Nested loops	Loop inside a loop to process sub-elements	`for(i...){for(j...){...}}`
While loop	Repeats while condition is true	`while(amount >= coin){amount -= coin;}`
Loop 1→N inclusive	Loop from 1 to N	`for(i=1;i<=5;i++){...}`
Compare arrays manually	Check if arrays are equal by looping	`arraysAreEqual([1,2],[1,2]) → true`
Loop with early return	Return from function as soon as condition is met	`if(condition) return result;`
Process 2D arrays	Parse CSV-like data with nested loops	`for(i=0; i<rows.length; i++){row.split(",")}`

🔢 8- STRING & NUMBER MANIPULATION

Concept	Description	Example	MDN Link
Access digits in number	Convert number→string to manipulate digits	`let str=String(12); str[0]="$"`	MDN – String()
Replace digit	Modify a character in number string	`"12"[1]="9"` → `"19"` (via new string)	MDN – String.prototype
Combine modified element + rest	`[newStr, ...arrays.slice(1)]`	`["$2", 5, 44]`	MDN – spread operator
Generate hex color	Create random color codes using base 16	`"#" + Math.floor(Math.random()*16).toString(16)`	-

---

🏗️ 9- BUILT-IN JAVASCRIPT OBJECTS & NAMING

Object	Description	Example	MDN Link
`String`	Built-in object for string operations	`String(123)` → `"123"`	MDN – String
`Number`	Built-in object for number operations	`Number("42")` → `42`	MDN – Number
`Array`	Built-in object for array operations	`Array.from({length: 3})`	MDN – Array
`Object`	Built-in object for object operations	`Object.keys({a:1})` → `["a"]`	MDN – Object
`Math`	Built-in object for mathematical operations	`Math.floor(4.7)` → `4`	MDN – Math
`Date`	Built-in object for date/time operations	`new Date()`	MDN – Date
`Boolean`	Built-in object for boolean operations	`Boolean(0)` → `false`	MDN – Boolean
`JSON`	Built-in object for JSON parsing/stringifying	`JSON.parse('{"a":1}')`	MDN – JSON
`console`	Built-in object for debugging output	`console.log("debug")`	MDN – console

⚠️ Naming Best Practices

NEVER use built-in object names as variable or parameter names!

❌ BAD	✅ GOOD	Why
`function test(String){...}`	`function test(str){...}`	`String` is a built-in object
`let Array = [1,2,3]`	`let arr = [1,2,3]`	`Array` is a built-in object
`let Number = 42`	`let num = 42`	`Number` is a built-in object
`let Object = {a:1}`	`let obj = {a:1}`	`Object` is a built-in object
`let Date = new Date()`	`let date = new Date()`	`Date` is a built-in object

Recommended variable names:

- For strings: `str`, `text`, `input`, `message`, `name`
- For numbers: `num`, `count`, `total`, `value`, `amount`
- For arrays: `arr`, `list`, `items`, `values`, `data`
- For objects: `obj`, `config`, `options`, `props`, `data`

💡 **Why this matters:** Using built-in object names as variables overwrites their default behavior and causes errors throughout your code.

📋 10- COMMON PATTERNS & ALGORITHMS

Pattern	Description	Example
Alphabetical sorting	Split string into chars, sort, join back	`"HELLO".split('').sort().join('')` → `"EHLLO"`
Vowel counting	Loop through string and check if char is in vowels array	`vowels.includes(char)`
Greedy algorithm	Make locally optimal choice at each step (coin change)	`while(amount >= coin){amount -= coin}`
Remove duplicates	Build new string by checking if char already exists	`if(!result.includes(char)) result += char`
Find first unique char	Use indexOf and lastIndexOf to check uniqueness	`if(str.indexOf(char) === str.lastIndexOf(char))`
Leap year calculation	Check if year is divisible by 4, 100, 400	\`year%4===0 && (year%100!==0
CSV parsing	Split string by newlines, then by commas	`csv.split("\n").map(row ⇒ row.split(","))`
Check all array elements	Use `.every()` to verify all meet condition	`arr.every(n ⇒ n%2===0)`
Object comparison	Compare two objects key by key	`Object.keys(a).every(k ⇒ a[k]===b[k])`

Pattern	Description	Example
Random generation	Generate random values in specific range	`Math.floor(Math.random() * max)`