

# JavaScript Strings - Full Mastery Pack

## Section 1: Concept Notes

### String Types

Primitive vs Object. Use primitive strings like: `let str = 'Hello';`

### Boxing

JS temporarily converts primitives to objects to allow method usage.

### String Creation

Use `"` or `'` or backticks. Avoid `new String()`.

### Template Literals

Use backticks (```) for multi-line and interpolated strings.

### Escape Characters

`\n`, `\\`, `\"`, `\t` are used to escape special characters.

### `slice(start, end)`

Extracts string portion. Supports negative index.

### `substring(start, end)`

No negative index support. Reverses indices if needed.

### `substr(start, length)`

Deprecated. Use `slice()` instead.

### `replace()` / `replaceAll()`

`replace()` changes only first match, `replaceAll()` changes all.

### `split(delimiter)`

Returns array split on delimiter.

### `trim()`

Removes whitespace from both sides.

### `concat()` vs `+`

Both join strings. Template literals preferred now.

### Case Conversion

Use `.toUpperCase()` and `.toLowerCase()`.

## **Searching**

`indexOf()` and `lastIndexOf()` return index of substring.

## Section 2: MCQs - Mock Test

**1. What will `typeof new String('hello')` return?**

- A. 'string'
- B. 'object'
- C. 'function'
- D. 'undefined']

**2. Which method allows negative indexes?**

- A. slice
- B. substring
- C. substr
- D. All of the above

**3. What does `str.trim()` do?**

- A. Changes case
- B. Trims whitespace
- C. Replaces characters
- D. Joins strings

**4. What's the difference between `replace()` and `replaceAll()`?**

- A. No difference
- B. `replaceAll` replaces all matches
- C. `replaceAll` works only with arrays
- D. `replace` does nothing

**5. What is returned by `str.split(' ')`?**

- A. String
- B. Object
- C. Array
- D. Boolean

**6. How can you check first occurrence of 'a' in a string?**

- A. `str.indexOf('a')`
- B. `str.charAt('a')`
- C. `str.search('a')`
- D. `str.find('a')`

**7. Does `substring()` support negative indices?**

- A. Yes
- B. No
- C. Only in Chrome
- D. Only in strict mode

**8. Which is preferred: + or template literals?**

- A. +
- B. template literals
- C. both same
- D. depends on data type

**9. What does \n escape sequence do?**

- A. Adds tab
- B. Adds newline
- C. Ends string
- D. Skips line

**10. How can you get last occurrence of a substring?**

- A. indexOf()
- B. findLast()
- C. lastIndexOf()
- D. reverseIndex()

### Section 3: Output-Based Questions

```
console.log('hi hi hi'.replace('hi', 'hello'));
```

```
console.log('Aman'.toUpperCase().toLowerCase());
```

```
console.log(' Hero '.trim());
```

```
console.log('JavaScript'.slice(-6));
```

```
console.log('Hello World'.substring(5, 0));
```

## Section 4: Practice Set

1. Write a function to manually split a string by space without using `split()`.
2. Write a function that replaces all 'hi' with 'hello' in a sentence.
3. Create a function to trim leading and trailing whitespace.
4. Extract first and last word of any sentence.
5. Convert 'JavaScript Is Fun' to 'javascript is fun'.
6. Implement a function that mimics `slice()` using `substring()`.
7. Write a function to count how many times a letter appears in a string.
8. Given 'Hello from Aman', return the initials.
9. Convert a sentence to CamelCase.
10. Find all indexes where 'a' occurs in 'banana'.

## Section 5: Quick 1-Page Cheat Sheet

### String Creation

"let str = 'Hello';" or "new String('Hello')" (Not recommended)

### Check Type

"typeof str" vs "typeof new String()"

### Length

"str.length" returns the length

### Concatenation

"str1 + str2" or "str1.concat(str2)" or Template Literals

### Template Literals

Use backticks: `Hello \${name}`

### Escape Characters

\n (newline), \\ (backslash), \" (quote), \t (tab)

### slice(start, end)

Returns part of the string. Allows negative indices.

### substring(start, end)

Similar to slice but no negative indices allowed.

### substr(start, length)

Deprecated, use slice instead.

### replace()

Replaces first match. Use replaceAll() for all matches.

### split(separator)

Splits string into array

### trim()

Removes whitespace from both ends

### toUpperCase() / toLowerCase()

Change case of string

### indexOf(sub)

Returns first index of substring

### **lastIndexOf(sub)**

Returns last index of substring