Js Strings

→ Can use (^) Single / Double (^) Quotes

15 Sunday

→ **length property** (^) To find (^) length of string.

→ **Escape character** (^) Backslash escape char ~ (^) /

String ← Special ← Turns
Char into

Code	Res	Detail
\'	'	Single Quote
\"	"	Double —"—
\\	\	Backslash

\b, \f, \n, \r, \t, \v

72

String → Methods
- Slice()
- Substring()
- Substr()
- replace()
- replace All()

- concat()
- toUppercase()
- toLowercase()
- trim()
- Split()

- charAt()
- charCodeAt()

9
- trimStart()
- trimEnd()
- padStart()
- padEnd()

(String extraction / replace)

10

S.no	Name	Detail	Syntax	eg
1	Slice()	Extracts → Part of string ↓ in Extracted Return part ↳ new string. Counts position from 0	Slice (start, end) ↓ Not Included	$\overset{-5\,-4\,-3\,-2\,-1}{\underset{0\,1\,2\,3\,4\,5\,6\,7}{Apple\ Bs}}$ slice (3,6) → le slice (3,7) → le B slice (3) → le Bs slice (-2) → Bs slice (-5,-1) → le B
2)	Substring()	↳ Similar to slice() ↳ Start & end values <0 is treated as 0	Substring (start, end)	Substring (3,6) → le Substring (-5,-1) → ' ' Substring (2,2) → Ap as() Consider
3)	Substr()	↳ Similar to slice() ↳ 2nd parameter ↳ specify → length of extracted part	Substr (start, length)	Substr(0,5) → Apple Substr (0,7) → Apple B
4)	replace() ↓ Case Sensitive	replace specified value with another value in string. returns new string. replaces only 1st match	replace (Text to be replaced , Text to replace)	

↳ to replace all match → use → R.E → with /g flag set

↑ flag set

String → convert → Array

12) split() — split (seperator) → omitted → return same string
↳ " " → return array of single chars

17 Tuesday 077-289 | Week 12		

replaceAll()
("Cats", "Dogs")

8 | replaceAll() | 1) allow → specify |
9 | | 2) instead ← Reg' exp | ~x~ | who Cats we Cats ↓ ↓

10 | | String → to be replaced |

11 | | If → para 1) Re ↓ | replaceAll() ("Cats/g, "Dogs")

12 | | Type ⟨ else → global flag set ⟩ Error |

1 | |

6) toUppercase() — Convert to uppercase | ~x~ | ~∞~

7) toLowercase() — Convert to lowercase | ~x~ | ~y~

8) concat() — Join 2 or more Strings | String.concat("", string) ↓1 ↓2 | ~y~

9) trim() — Remove whitespace from both side of strings | ~x~ | ~y~

Start 1) remove 1) trimStart()
of string
End of string 1) 2) 1) trimEnd() | ~y~

10) charAt() — returns 1) char. ↓ specified ← at index in string | ~x~ | ~∞~

11) charCodeAt() returns unicode → char 1) UTF string ← specify index ← at 1) → returns Code (0 & 65535)

12	indexOf()	Return Index (pos) the first occurrence of string in string	indexOf("string", start)	⌐x⌐
13	lastIndexOf()	Return Index of last occurrence of specified text in string. Return -1 if not found	lastIndexOf("str", start)	⌐x⌐
14	Searches	Searches string for string/Re Returns position of match	Search("string")	⌐x⌐
15	match()	Return of Array results ← contain match string against string/Re Return only 1st match in string Return of Iterator → matchAll()	match("string")	⌐x⌐
16	Includes ()	(else false) Returns True if it ← contain ← string Specified value Case sensitive	n) Include("string", "pos")	

Start

match string

| 17 | Startwith() | n) Return n) if n) String " begin with specified value True | |
| 18 | endwith() | n) Return n) if n) String n) end ⌐n⌐ True | |

19

079-287 | Week 12

String Templates
Back-Ticks

Thursday Use Back-Ticks rather than quotes to
define string.

Can use both 'single' / "double" quotes inside string.

allow Multiline Strings

Interpolation :' provide
↓
easy → Interpolate Variables / expressions into Strings.

'Syntax → $\boxed{\${ \ldots }}$

eg: let f = "John";

let l = "Doe";

let text = `Welcome ${f} , ${l} `;