

## Object Function

① Object.freeze (empobj)  $\Rightarrow$  By using this fun<sup>n</sup> of object we can not add, delete and modify any property in existing object.

② Object.seal ( )  $\Rightarrow$  prevent to add & delete new property but they allow to replace or edit existing property value

empobj = {  
 1  
}

empobj  
seal  $\Rightarrow$  {  
 1  
}

③ Object.defineProperty (objectname, 'property name', {  
writable: false,  
configurable: false  
})

$\Rightarrow$  We restrict the <sup>specific</sup> property settings with the help of defineProperty fun<sup>n</sup>

# String Operation

① Split() → They split the string in subString (subpart) with delimiter which is we passed as a parameter.

e.g. `str.split(' ')`

② slice() → slice **its** take a substring from given string using start index & end index but exclude the value of end index.

③ substring() → substring fun<sup>n</sup> also return the substring value from given input string but they do not operate on -ve values of parameter.

Diagram illustrating string slicing on the string "I tried Solution":

Indices: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

String: I | t | r | i | e | d |   | S | o | l | u | t | i | o | n

Annotations:

- For "I tried": start index 0, end index 6 (exclusive), resulting in indices -5 to -1.
- For "Solution": start index 8, end index 15 (exclusive), resulting in indices 1 to 7.
- Combined slice: start index -5, end index -1, resulting in indices -3 to -1.
- Final slice: start index -3, end index -1, resulting in indices -2 to -1.

Diagram illustrating string slicing on the string "I tried Solution":

Annotations:

- `str.slice(0, -1)`: start index 0, end index -1 (exclusive), resulting in the entire string.
- `str.slice(8, -1)`: start index 8, end index -1 (exclusive), resulting in "Solution".
- `str.slice(-2, -1) ⇒ 0`: start index -2, end index -1 (exclusive), resulting in the character 'o'.
- `str.slice(-3, -4)`: start index -3, end index -4 (exclusive), resulting in an empty string.

0 1 2 3 4 5

ltview2 Solution



ltv

substring (0, 3)

substring (5, -2)  
↑  
0

slice

slice (-2, -3)  
invalid  
range

str.substring (0, -1)  
↑  
-1 to 0

str.substring (0, 0)  
↑  
-1

str.substring (3, -4)  
↑  
0  
(3, 0) x  
←

(5, 0)

(0, 5) ltview →

substring

slice (-2, -3)  
↑  
0  
slice (0, -2)

slice (4, 2)

↓  
slice (2, 4)

## Array

⇒ Array is also one of the non primitive data types  
⇒ reference type data type

$obj_1 == obj_2$

$a_1 == a_2$

let arr = [ "Itview", 12, true, null, undefined,