① #7 String in JavaScript [6:40:40]

__/__/__

- Escape character
- Finding String in a String
- Searching for String in a String
- Extracting String Parts
- Replacing String Content
- Extracting String character
- Other useful Method

1 String in JavaScript

If string is zero or more characterist written inside goutes.

- JavaScript string are Store and monuplating (changes) text.
- You can use single quotes or double goutes
- ★ String can be created or primitives, from string literals, or as objects, using the String() constructor

```
let myName = "Moteeullah Azmi";
let myChannelName = 'Azmi Web3';

★★ // let ytName = new String ("Azmi Dev")

console.log (myName);        ⇒ Moteeullah Azmi

console.log ((ytName));      ⇒ [String : '~~Hopa Fe~~
                                 Azmi Web3']
```

2

⇒ Length ()

Reflect the length of String.

```
let myName = "Aimi";
console.log(myName.lenghth); ⇒ 5
```

(दबा के निकाला।)

1. <u>Escape character</u>

The

<u>Backslash Scape</u>

It turns special character into string character.

Code	Result	Description
\'	'	Single - qoutes
\"	"	Double - qoutes
\\	\	Backslash

<u>Escape character</u>

If you not used mess, then apply alternates qoutes.

```
let anySentence = "Welcome to \"Aimian\" World".
                                ↓         ↓
                                "         "
```

Or

```
let any Sentence = 'Welcom to "Aimian" World.
```

2. Finding a String in a String

a) indexOf (searchValue [, fromIndex]) (Forward)

The indexOf() method returns the index of (the position of) the first occurance specied text in a string.

```
Const myBio = "I am Azmi from Azmi india";
console.log (myBio.indexOf ("Azmi");
                                        ⇒ 5
console.log (my Bio. index Of ('am")    ⇒ 2 ⇒ i _ am
                                              0 1  2
console.log (my Bio. indexOf ('Azmi', 9)  After - 9
                                        ⇒ (16)
```

b) lastIndexOf (backward)

Return the index within the calling String Object of the last occurence of searchValue, or -1 if not fund.

programm

```
Const myData = 'I am Azmian Web3';
console.log (myBio. last indexOf ("A", 6)

'I am Az
       (6)
      ←
[(5)]
```

8

3. Searching for a String in a String

"Only search a data present or not.
not start a index position."

```
const myBioData = 'Ia am Mateenllah';
let SData = myBioData.search ("am");
    console.log(SData);
```

output ⇒ 2

index

4.) Extracting String parts

There are three method

1) slice (start, end)
2) substring (start, end)
3) substr (Start , length)

a) slice () Method

Azmi,
0 1 2 3 4

⇒ Azmi

Extract part of Slice String and return the extracted part.

```
let str = "Azmi, Adil, Emad";
  let res = str.slice(0,4);
  console.log(res)
```

Azmi;
0 1 2 3 4

⇒ Azmi why ⇒ ?

why ?

slice()

Starting element at the given start argument, and end at, but does not include, the given end argument.

Note

The original array will not be changed.

Remember

Javascript counts positions from zero, First position is 0.

```
var   str = "Apple, Banna, Kiwi, mongo";

      let res = str.slice (7,-2);
      Console.log(res);
```

-2 end

Apple, B
0 1 2 3 4 5 6 7

go
-2 -1

output Bananaa, Kiwi, Man

10. Challange Time

⇒ Display only 280 characters of a string like the one used in Twitter?

```
let myTweets = "  ________

                                  ";

let myActual Tweet = myTweets.slice(0,280);
console.log (myActualTweet);
console.log (myActualTweet.length);
```

output 280 charaters

↓
280

6

b) Substring Method

Substring() is similar to slice()

Difference

Substring() cannot accept negative index.

```
var str = "Apple, Banana, Kiwi";
let res = str.substring(0,4);
console.log (res)
```

Exclude

output Appl

A p p l e
0 1 2 3 4
exclud

⇒
```
var str = "Apple, Banna, Kiwi';
let res = str.substring (8, -2)
console.log (res)
```

output ⇒ Apple,_B

= Apple,_Ban
0 1 2 3 4 5 6 7 8 → exclud

Note • If we give negative value then the charect are counted from the 0th position

c) The substr Method

substr() is similar to slice().

differeng

The difference is that the second parameter specifies the length of extracted parts.

var str = "Apple, Banana, Kiwi";

let res = str.substr(0,4); ⇒ Appl

let res = str.substr(~~@~~ 7,-2); ⇒ empty

Imp ⇒ let res = Str.substr(-4); ⇒ Kiwi

5. Replacing String Content

The replace() method replaces a specified value in a String.

let myBioData = 'I am Moteullah azmi ~~Azmi~~ to change' ~~Azmi~~ azmi

Syntax

String.prototype.replace(searchFor, replaceWith)

let replaceData = myBioData.replace('azmi' 'Azmi');

~~@~~ output

I~~@~~ am Moteullah Azmi to change azmi

↑

not chang

Point Remember

1: The replace() method does not original change the string it is called on. Its return a new string.

2. By default, the replace() method replace only first match.

3. By default, the replace() method is case sensitive.

Case Sensative

6. Extracting String character

There are 3 method

a> charAt (position)

b> charCodeAt (position)

* c> property access [ ]

a> charAt() Method

method returns the character at a ~~a~~ specified index (position) in a string.

Programme

```
let str = "HELLO WORLD";

Console.log (str.charAt (9)); L
                        (0) ⇒ H
```

b> CharCodeAt()

```
Console.log(str.CharCodeAt(0)); ⇒ 72 ↔ H
```

" • It method returns the unicode of the character at a specified index in a string."

- The method return a UTF-16 code (a integer 0 and 65535).

- The Unicode Standard provide a unique number for every character, no matter the platform, device, application or language. UTF-8 is ~~particular~~ popular Unicode encoding which has 88-bit code units,

⇒ Unicode of last character in a string

```
let str = "Hello World";

let last~~[illegible]~~Str = str.length;-1;
console.log(str.charCodesAt(lastChar));
```

output ÷ 68 → D

2. ~~oth~~ c> property Access

ECMAScript 5 (2009) allows property access [ ] on strings.

```
Var str = "Hello World";
console.length (str [0]);
```

→ not array deal string

0 ⇒ H

7. Other useful Method

```
let myName = " Moteeullah Azmi";
console.log (myName. toUpperCase ());
console.log (my Name. toLower Case () );
```

output → MOTEEULLAH

moteullah

~~Con~~ concat() method

```
let fName = "Moteeullah";
let lName = " Azmi";
console.log (fName + lName)
console.log (fName. concat (lname));
console.log (fName, concat (" ", lname));"
```

Moteeullah Azmi

Moteullah Azmi

Moteeullah Azmi

trim()

Trim method used White space remove from both side. not middle.

```
Var str = "... Hello World ....";
console.log(str.trim());
```

"Hello_World"

★ ★★ ⇒ How to convert string to an Array?

Split method

[7:38:00]

```
Var txt = "a,b,c,d,e";
console.log(txt.split(","));   // split commas
  〃   〃  ,  〃 (" "));        // split on space
console.log(txt.split("|"));   // Split on Pipe
```

split commas → ['a', 'b', 'c', 'd', 'e']

split on space → ['a, b, c, d, e']

if ⇒ var = "a,b,c,| d,e"

['a,b,c', 'd,e']

Two Part

if space "a_b,c_d,e"

['a', 'b', 'c d', 'e']