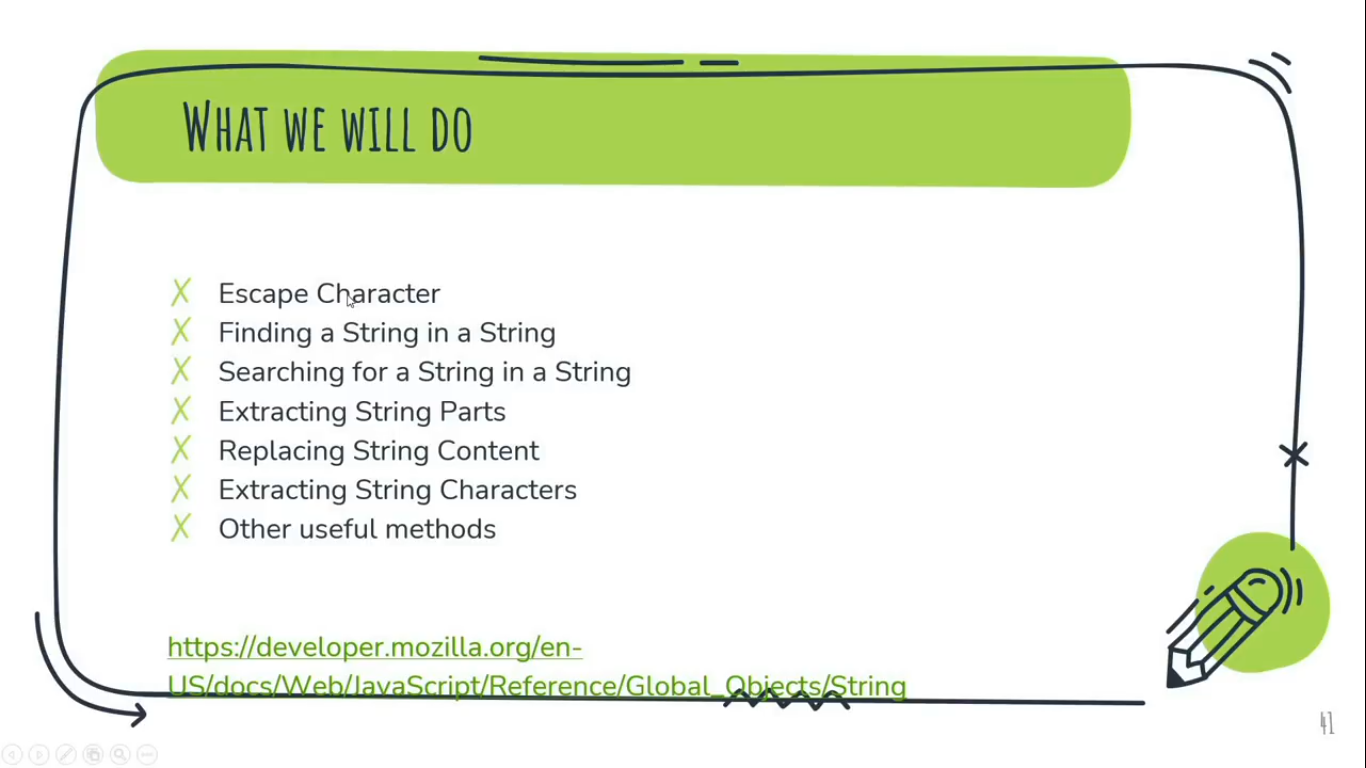
**Strings in JavaScript**

****

* A JavaScript string is zero or more characters written inside quotes.
* JavaScript strings are used for storing and manipulating (controling) text.
* We can use single or double quotes for the strings.
* Strings can be created as primitives(pre-defined), from string literals, or as objects, or using the String() constructor.

String Constructor is just like array constructor where we use array class and pass element like (new Array(‘Chandan’)) that we use to create an array and store it to a variable.  
String Constructor returns strings in the form of an array and the string will be primitive in nature.

Example:

let myName = new String("Chandan Kumar", "King");

console.log(myName);

Output:-  
String {'Chandan Kumar'}

**Here, It returns strings in the form of an array and “Chandan Kumar” is a primitive value.**

**\*How to find the length of a string?**

**Sol:-** We are going to **use**

**String.prototype.length** property which reflects the length of the string.

Example:

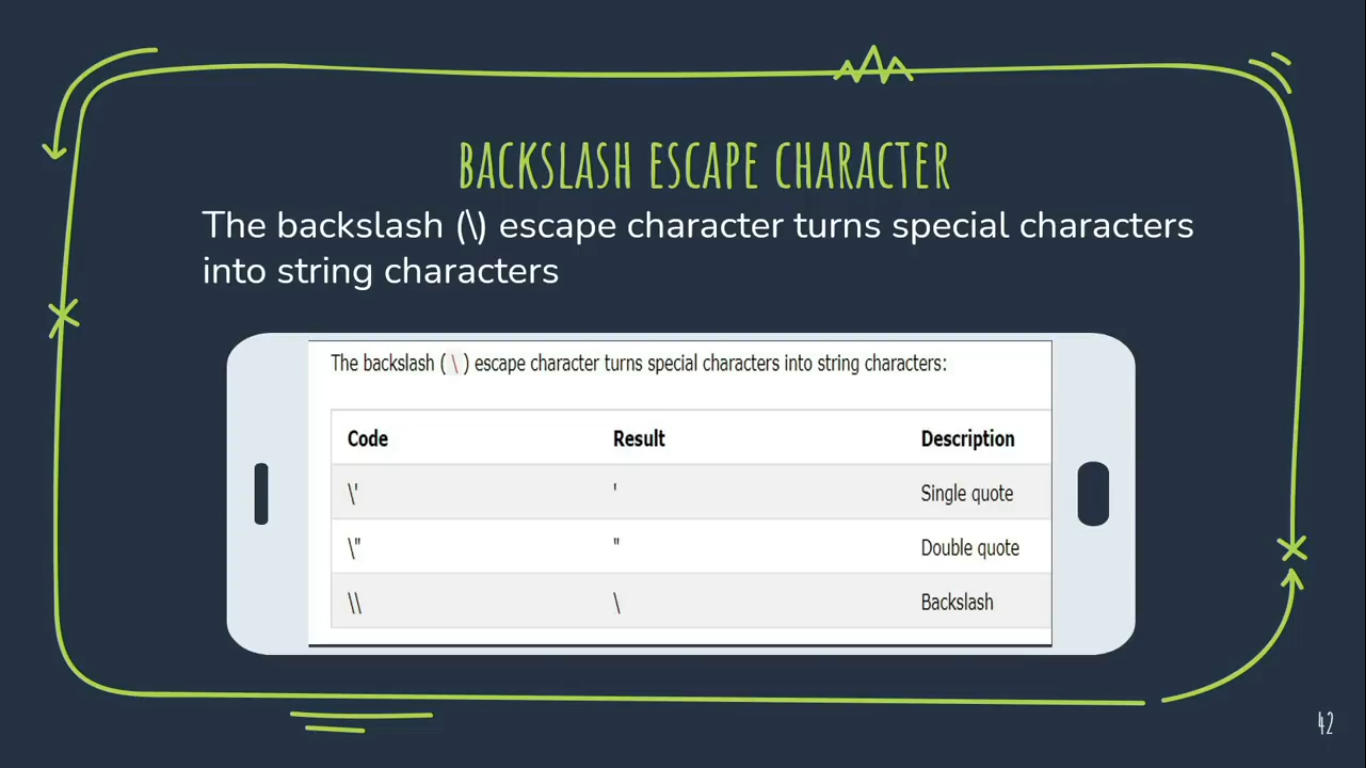
let myName = "Chandan Kumar";

document.write(myName.length);

Output:-

13

**Escape Character**

****

It would be better, to use alternate quotes if both single & double quotes are included in a single string. Otherwise, we have to use Backslash (\) Character in order to use same quotes multiple times in a single string.

Example-1: **\\ BackSlash Character**

let myName = "Chandan Kumar is a \"Web Developer\"";

document.write(myName);

Output:-  
Chandan Kumar is a "Web Developer"

Example-2: [**\\ Alternate**](file:///\\Alternate) **Quote**

let myName = 'Chandan Kumar is a "Web Developer"';

document.write(myName);

Output:-  
Chandan Kumar is a "Web Developer"

**Finding a string in a string**

**#1 Method** String.prototype.indexOf()

The indexOf() method returns the index of (the position of ) the first occurrence/letter of a specified text in a string. Return -1, if no data found or element doesn’t match. It’s all property will work as same as of indexOf() method worked for array.

**Syntax**: indexof(searchValue, fromIndex)

Example-2:

let myName = 'Chandan Kumar is a Web Developer';

document.write(myName.indexOf('Kumar'));

Output:-  
8

Example-1:

let myName = 'Chandan Kumar is a Web Developer';

document.write(myName.indexOf('a',6));

Output:-  
2

**#2 Method** String.prototype.lastIndexOf()

**Syntax:**   
String.prototype.lastIndexOf(searchValue, fromIndex)

lastIndexOf() method returns the index within the calling string object of the last occurrence of searchValue, or -1 if not found.

Example:

let myName = 'Chandan Kumar is a Web Developer';

document.write(myName.lastIndexOf('a', 15));

Output:-

17

**Searching for a String in a String**

**#1 Method** String.prototype.search()

Syntax:   
search(regexp)

The search() method searches a string for a specified value and returns the position of the match. Return -1 if no element found.

Example:

let myName = 'Chandan Kumar is a Web Developer';  
document.write(myName.search('Kumar'));

Output:-  
8

**Note:**The search() method cannot take a second start position parameter like, search(‘Kumar’, 10). The second parameter will not affect to the final result anymore.

**Extracting String Parts**

There are 3 methods for extracting a part of a string:

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

**#1 Method** - slice() method

The slice() method extracts a part of a string and returns the extracted part in a new string.

The slice() method selects the elements starting at the given start argument, and ends at, but does not include, the given end argument.

**Note:** The original string will not be changed.

The method takes 2 parameters: the start position,

And the end position (optional).

Example - 1:

let str = "Apple, Banana, Kiwi";

document.write(str.slice(0,4));

Output:-

Appl **//string taken from 0 up to 3rd index of string but below than 4**

Example - 2:

let str = "Apple, Banana, Kiwi";

document.write(str.slice(4));

Output:-

e, Banana, Kiwi **//string taken from 4th index of string.**

Example - 3:

let str = "Apple, Banana, Kiwi";

document.write(str.slice(7,-2));

Output:-

Banana, Ki   
**//string taken from 7th index upto 16th index of string i.e. -2 index from last**

Here, the starting index of the string is from ‘B’ i.e. 7 and go upto the end of the whole string then move back by -2 from the last index of the text.

**#2 Method** - substring() method

The substring() method is similar to slice().

The difference is that substring() cannot accept negative indexes.

Example – 1

let str = "Apple, Banana, Kiwi";

document.write(str.substring(0,4));

Output:-

Appl **//string taken from 0 to 4**

Example – 2

let str = "Apple, Banana, Kiwi";

document.write(str.substring(4));

Output:-

e, Banana, Kiwi **//string 4 to end**

**Note:** If we give negative value to the second argument then the characters are counted from the 0th position.

Example – 3

let str = "Apple, Banana, Kiwi";

document.write(str.substring(7, -2));

Output:-

Apple, **//string taken upto 6 but below than 7**

**#3 Method** - substr() method

The substr() method is similar to slice().

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

**Note:** No value will return by substr() method using negative value in the second parameter of this method.

Example:

let str = "Apple, Banana, Kiwi";

document.write(str.substr(-4));

Output:-

Kiwi **//string taken from the last index by -4**

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

let tweet = "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor reprehenderit voluptate velit essecillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat nonproident, sunt culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmodempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodoconsequat. Duis aute irure dolor reprehenderit voluptate velit ";

let myTweet = tweet.slice(0, 280);

document.write(myTweet);

**Replacing String Content()**

**#1 Method –** string.prototype.replace()

Syntax:   
String.prototype.replace(searchFor, replace with)

The replace() method replaces a specified value with another value in a string instantly when the value found. But it will replace only the first one value at a time.

Example:

let myBio = "My Name is Chandan Kumar Chandan";

document.write(myBio.replace('Chandan', 'King'));

Output:-

My Name is King Kumar Chandan

**Note:**

* The replace() method does not change the main string. It returns a new string.
* By default, the replace() method replaces only the first match.
* By default, the replace() method is case sensitive. Means the searching parameter in this method is case sensitive.

**#2 Method –** string.prototype.replaceAll()

Syntax:   
String.prototype.replaceAll(searchFor, replace with)

The replaceAll() method replaces a specified value with another value in a string. It will replace all multiple value in a single string at a time.

Example:

let myBio = "My Name is Chandan Kumar Chandan";

document.write(myBio.replaceAll('Chandan', 'King'));

Output:-

My Name is King Kumar King

**Extracting String Characters**

There are 3 methods for extracting string characters:

1. charAt(position)
2. charCodeAt(postion)
3. Property acces [ ]

**#1 Method –** charAt()

The charAt() method returns the character or string value at a specified index(position) in a string.

Example:

let myBio = "My Name is Chandan Kumar Chandan" document.write(myBio.charAt(11));

Output:-

C

**#1 Method –** charCodeAt()

The charCodeAt() method returns the Unicode of the character at a specified index in a string. The method returns a UTF-16 code (an integer value between 0 and 65535 in JavaScript).

The Unicode Standard provides a unique number for every character, no matter the platform, device, application, or language. UTF-8 is a popular Unicode encoding which has 88-bit code units which is currently using JavaScript as well.

Example:

let myBio = "My Name is Chandan Kumar Chandan" document.write(myBio.charCodeAt(11));

Output:-

67

**\*Return the Unicode of the last character in a string**

**Sol:**🡪

let myBio = "My Name is Chandan Kumar Chandan";

document.write(myBio.charCodeAt(myBio.length-1));

Output:-

110

**#3 Method –** Property Access []

ECMAScript 5 (2009) allows property access [ ] on strings. It work as an array with strings.

Example:

let myBio = "My Name is Chandan Kumar Chandan";

document.write(myBio[11]); **//Index number**

Output:-  
C

Other useful methods of Strings

**#1 Method –** toUpperCase()

Example:

let myBio = "My Name is Chandan Kumar";

document.write(myBio.toUpperCase());

Output:-  
MY NAME IS CHANDAN KUMAR

**#2 Method –** toLowerCase()

Example:

let myBio = "My Name is Chandan Kumar";

document.write(myBio. toLowerCase());

Output:-  
my name is chandan kumar

**#3 Method –** concat()

Example - 1:

let fName = "Chandan";

let lName = "Kumar";

document.write(fName.concat(lName));

Output:-  
ChandanKumar

Example - 2:

let fName = "Chandan";

let lName = "Kumar";

document.write(fName.concat(“ ”,lName));

Output:-  
Chandan Kumar

**#3 Method –** concat()

The trim() method removes whitespace from both sides of a string. But not remove whitespace in between of the string.

Example:

let str = " Hello World "; **//white space included on both side & middle of the string**

console.log(str.trim());

Output:-  
Hello World

**Converting a String to an Array**

A string can be converted to an array with the split() method.

Example:

var txt = "a, b, |, d, e";

console.log(txt.split(",")); **//split elements from commas**

console.log(txt.split(" ")); **//split elements from spaces**

console.log(txt.split("|")); **//split elements from pipe**

console.log(txt.split("$")); **//split elements from $**

Output:-  
['a', ' b', ' c', ' d', ' e'] //**array length = 5**

['a,', 'b,', 'c,', 'd,', 'e'] //**array length = 5**

['a, b, ', ', d, e'] //**array length = 2**

['a, b, c, d, e'] //**array length = 1 (no $ is present)**