**Strings**

**1)String constructor:- x**

Creates string objects

let a = new String("friend");

console.log(a);

**2) at() x**

Takes integer and returns a new string with the character at that index.

Can take negative indexes to access characters the back of string

const sentence = "My name is Jay";

console.log(sentence.at(3));

output = n

**3) charat() x**

Same as at but cannot use negative index , if try giving it a negative index it will return an empty string

**4) charcodeat() x**

Returns character code (utf-16) at that index

const sentence = "My name is Jay";

console.log(sentence.charCodeAt(3));

**5) concat() – done d**

Merges two strings

const str1 = "Hello";

const str2 = "Dost";

console.log(str1.concat(" ", str2));

console.log(str1.concat("\*", str2));

Hello Dost

Hello\*Dost

6) endsWith() - ***done***

Returns true or false if a string ends with a characters of the string

const str1 = "My name is khan";

console.log(str1.endsWith("khan"));

7) includes() - **done**

Checks whether a specific substring is anywhere in the string.

const str1 = "My name is khan";

console.log(str1.includes("khan"));

8) indexof() **done**

Returns the index of the starting character in a string.

const str1 = "My name is khan";

console.log(str1.indexOf("khan"));

output 11

9) lastIndexof() ***done***

const str1 = "I am Salman Khan. My father is Monu Khan Sahab";

console.log(str1.lastIndexOf("Khan"));

Returns the index of last occurrence of a word.

10) Repeat() ***done***

Returns copies of the string concatenated

const mood = "Hapi ";

console.log(`I feel so ${mood.repeat(3)}`);

O/P : I feel so Hapi Hapi Hapi

11) replace() ***done***

Returns a new string with a substring replaced.

const mood = "I am very happy ";

console.log(mood);

let x = mood.replace("happy", "sad");

console.log(x);

I am very happy

I am very sad

12) replaceAll() ***done***

Returns a new string with all substring replaced.

const mood = "I am very happy happy happy  ";

console.log(mood);

let x = mood.replaceAll("happy", "sad");

console.log(x);

I am very happy happy happy

I am very sad sad sad

13) search() ***done***

Executes a search and returns the index if matched.

const mood = "I am very happy happy happy  ";

let x = mood.search("very");

console.log(x);

op- 5

14) slice() **done**

Extracts a given part of string

Returns new string

const mood = "I am very happy because I am rich.";

let x = mood.slice(4); // till end

let y = mood.slice(5,15); // till given range

console.log(x);

console.log(y);

op:

very happy because I am rich.

very happy

15) split() - ***done***

Takes a pattern and divides the string into substrings based on that pattern and returns an array.

const mood = "I am very happy because I am rich.";

let x = mood.split(" ");

console.log(x[0]);

console.log(x[7]);

op

I

rich.

16) startsWith() - ***done***

Checks whether a string starts with a specific substring.

const mood = "I am very happy because I am rich.";

console.log(mood.startsWith("I"));

op – true

17) substring() ***done***

Returns substring from the given range

const mood = "I am very happy because I am rich.";

console.log(mood.substring(1)); // till end

console.log(mood.substring(1,10)); // till range

18) toLowercase() - ***done***

const mood = "I am very happy because I am rich.";

console.log(mood.toLowerCase());

console.log(mood);

19) tostring() ***done***

Converts to string

const mood2 = 123;

console.log(mood2);

console.log(mood2.toString());

20) toUppercase() - ***done***

const mood = "I am very happy because I am rich.";

console.log(mood.toUpperCase());

console.log(mood);

I AM VERY HAPPY BECAUSE I AM RICH.

I am very happy because I am rich.

21) trim() ***done***

Removes white spaces from both the ends of the string and returns new string.

const mood = "      I am very happy because I am     rich.      ";

console.log(mood.trim());

I am very happy because I am rich.

22) trimEnd()***done***

Removes white spaces from the end only.

const mood = "      I am very happy because I am     rich.                    ";

console.log(mood.trimEnd());

I am very happy because I am rich.

23) trimStart()***done***

Same

24) valueof() – ***done***

Returns the value of string.

const x = new String("Hello world");

console.log(x.valueOf()); // 'Hello world'

console.log(x); // 'Hello world'

Hello world

[String: 'Hello world']

25) length - ***done***

Returns length of the string

const x = new String("Hello world");

console.log(x.length);

op - 11

Length

Valueof

toLower

toUpper

concat

endswith

includes

split

repeat

startswith

replace

replaceall

search

indexof

lastIndexof

substring

trim

trimstart

trimend