SOFT CLUB

JAVA SCRIPT

LECTURE 2

TABLE OF CONTENTS

What is a method in JS?

A method is a block of code which only runs when it is called. You can pass data, known as parameters, into a method. Methods are used to perform certain actions, and they are also known as functions.

CREATE JS STRINGS

"Double Quotes"

```
"Hello"
```

'Single Quotes'

```
'Hello'
```

`Backticks`

```
`Hello ${hi}`
```

SOFT CLUB

JavaScript String Methods

charAt()

concat(str1, str2, ...)

includes()

indexOf()

repeat()

replace()

replaceAll()

search()

slice()

split()

substr()

substring()

toLowerCase()

toUpperCase()

toString()

trim()

JavaScript String method charAt()

The charAt() method returns the character at a specified index (position) in a string.
The index of the first character is 0, the second 1, ...
The index of the last character is string length - 1 .

Get the first character in a string:

```
// 1
let text = "HELLO WORLD";
let letter = text.charAt(0);
console.log(letter);
```

```
node /tmp/1d1LKTWZwx.js
H
```

Get the second character in a string:

```
// 2
let text = "HELLO WORLD";
let letter = text.charAt(1);
console.log(letter);
```

```
node /tmp/1d1LKTWZwx.js
E
```

Get the last character in a string:

```
let text = "HELLO WORLD";
let letter = text.charAt(text.length-1);
console.log(letter);
```

```
node /tmp/Z0DgPxhxm5.js
D
```

JavaScript String method concat()

The concat() method joins two or more strings.
The concat() method does not change the existing strings.
The concat() method returns a new string.

Join two strings:

```
const text1 = 'hello';
const text2 = 'world';
const result1 = text1.concat(' ', text2);
console.log(result1);
```

```
node /tmp/FNBL7MN5w2.js
hello world
```

Join three strings:

```
// 2
let text1 = "Hello";
let text2 = "world!";
let text3 = "Have a nice day!";
let result = text1.concat(" ", text2, " ", text3);
console.log(result);
```

```
node /tmp/FNBL7MN5w2.js
Hello world! Have a nice day!
```

JavaScript String method replace()

The **replace()** method searches a string for a value or a regular expression.
The **replace()** method returns a new string with the value(s) replaced.
The **replace()** method does not change the original string.

Replace Microsoft:

```
// 1
let text = "Visit Microsoft!";
let result = text.replace("Microsoft", "Soft club");
console.log(result);
```

```
node /tmp/FNBL7MN5w2.js
Visit Soft club!
```

JavaScript String method replaceAll()

The **replaceAll()** method returns a new string with all matches of a pattern replaced by a replacement.

```
const p = 'The quick brown dog fox jumps over the lazy dog.';
console.log(p.replaceAll('dog', 'monkey'));
//"The quick brown monkey fox jumps over the lazy monkey."
```

JavaScript String method split()

The **split() method** splits a string into an array of substrings. The split() method returns the new array. The split() method does not change the original string. If (" ") is used as separator, the string is split between words.

Examples:

```
1  // converting the string to an array
2  const text = 'hello';
3  const result = text.split();
4  console.log(result);
5  //2
6  let text2 = "How are you, doing today?";
7  const myArray = text2.split(" ");
8  console.log(myArray);
9  //3
10 let text3 = "How are you, doing today?";
11 const myArray2 = text3.split(" ", 3);
12 console.log(myArray2)
```

```
node /tmp/TGHxjCVANO.js
[ 'hello' ]

[ 'How', 'are', 'you,', 'doing', 'today?' ]

[ 'How', 'are', 'you,' ]
```

JavaScript String method substr(*start, length*)

The substr() method extracts a part of a string.
The substr() method begins at a specified position, and returns a specified number of characters.
The substr() method does not change the original string.
To extract characters from the end of the string, use a negative start position.

Extract a substring from text:

```
1  //1
2  let text = "Hello world!";
3  let result = text.substr(1, 8);
4  console.log(result)
5  //2
6  let text2 = "Hello world!";
7  let result2 = text2.substr(-6, 6);
8  console.log(result2)
```

```
node /tmp/q12McE42mX.js
ello wor

world!
```

JavaScript String method substring(*start,end*)

The substring() method extracts characters, between two indices (positions), from a string, and returns the substring.
The substring() method extracts characters from start to end (exclusive).
The substring() method does not change the original string.
If start is greater than end, arguments are swapped: (4, 1) = (1, 4).
Start or end values less than 0, are treated as 0.

Examples:

```
1  //  Extract a substring from text:
2  let text = "Hello world!";
3  let result = text.substring(2);
4  console.log(result)
5  //  If "start" is less than 0, it will start from index 0:
6  let text2 = "Hello world!";
7  let result2 = text2.substring(-3);
8  console.log(result2)
9  //  Only the last:
10 let text3 = "Hello world!";
11 let result3 = text3.substring(text.length - 1);
12 console.log(result3)
```

```
node /tmp/q12McE42mX.js
llo world!

Hello world!

!
```

JavaScript String method slice(*start, end*)

The **slice()** method **returns a shallow copy of a portion of an array into a new array object selected from start to end** (end not included) where start and end represent the index of items in that array.

Examples:

```
// Slice the first 5 positions:
let text = "Hello world!";
let result = text.slice(0, 5);
console.log(result)
// From position 3 to the end:
let text2 = "Hello world!";
let result2 = text2.slice(3);
console.log(result2)
// The whole string:
let text3 = "Hello world!";
let result3 = text3.slice(0);
console.log(result3)
```

```
node /tmp/q12McE42mX.js
Hello

lo world!

Hello world!
```

JavaScript String method toLowerCase()

The toLowerCase() method converts a string to lowercase letters.
The toLowerCase() method does not change the original string.

Example:

```
// Convert to lowercase:
let text = "Hello World!";
let result = text.toLowerCase();
console.log(result)
```

```
node /tmp/q12McE42mX.js
hello world!
```

JavaScript String method toUpperCase()

The toUpperCase() method converts a string to uppercase letters, using current locale.
The toUpperCase() method does not change the original string.

Example:

```
// Convert to uppercase:
let text = "Hello World!";
let result = text.toLocaleUpperCase();
console.log(result)
```

```
node /tmp/q12McE42mX.js
HELLO WORLD!
```

JavaScript String method trim()

Method **trim()** removes whitespace from both sides of a string.
The **trim()** method does not change the original string.

Example:

```
// Remove spaces with trim():
let text = "       Hello World!        ";
let result = text.trim();
console.log(result)
```

```
node /tmp/q12McE42mX.js
HELLO WORLD!
```

JavaScript String method includes()

The includes() method returns true if a string contains a specified string.
Otherwise it returns false.
The includes() method is case sensitive.

Examples:

```
ludes "world".
let text = "Hello world, welcome to the universe.";
let result = text.includes("world");
console.log(result)
// Start at position 6:
let text2 = "Hello world, welcome to the universe.12";
let result2 = text2.includes("world", 6);
console.log(result2)
```

```
node /tmp/q12McE42mX.js
true

true
```

JavaScript String method search()

The search() method matches a string against a regular expression **
The search() method returns the index (position) of the first match.
The search() method returns -1 if no match is found.

Examples:

```
// Search for "Blue":
let text = "Mr. Blue has a blue house";
let position = text.search("Blue");
console.log(position)
```

```
node /tmp/q12McE42mX.js
4
```

JavaScript String method toString()

SOFT CLUB

The toString() method returns a string representing the object.

By default toString() takes no parameters.

C: > Users > adnan > OneDrive > Desktop > JS numtostr.js > ...

```
var n = 99;
console.log(typeof(n));
var st = n.toString();
console.log(typeof(st));
```

Code

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude)

```
C:\Program Files\nodejs\node.exe .\numtostr.js
number
string
```

Output

JavaScript String method charAt()

The `indexOf()` method returns the position of the first occurrence of a value in a string.
The `indexOf()` method returns -1 if the value is not found.
The `indexOf()` method is case sensitive.

```
1  const message = "JavaScript is not Java";
2  const index = message.indexOf("is");
3  console.log('index: ' + index);  // index: 2
```

```
node /tmp/Ql80ZbdYct.js
index: 11
```

JavaScript String method repeat()

The repeat() method creates a new string by repeating the given string a specified number of times and returns it.

```
const holiday = "Happy holiday!";
const result = holiday.repeat(3);
console.log(result);
```

```
node /tmp/3SJlvCrvwM.js
Happy holiday!Happy holiday!Happy holiday!
```

JavaScript Number methods

JavaScript Number methods Math.round(),ceil(),floor()

The Math.floor() function rounds down a number to the next smallest integer.

```
let number = 38.8;
let roundedNumber = Math.floor(number);
console.log(roundedNumber);
```

```
node /tmp/Jww9kTELqr.js
38
```

The Math.round() function returns the number rounded to the nearest integer.

```
let number = 3.87;
let roundedNumber = Math.round(number);
console.log(roundedNumber);
```

```
node /tmp/mH3w7DA7Rw.js
4
```

The ceil() method rounds a decimal number up to the next largest integer and returns it.

```
let number = Math.ceil(4.3);
console.log(number);
```

```
node /tmp/yVro3yT8qw.js
5
```

JavaScript Number methods Math.max() and Math.min()

The max() method finds the maximum value among the specified values and returns it.

```
let numbers = Math.max(12, 4, 5, 9, 0, -3);
console.log(numbers);
```

```
node /tmp/Y7L1E2Mj22.js
12
```

The min() method finds the minimum value among the specified values and returns it.

```
let numbers = Math.min(12, 4, 5, 9, 0, -3);
console.log(numbers);
```

```
node /tmp/dNVmPNFZKB.js
-3
```

JavaScript Number methods Math.pow() and Math.sqrt()

The pow() method computes the power of a number by raising the second argument to the power of the first argument.

```
let power = Math.pow(5, 2);
console.log(power);
```

```
node /tmp/fnDIIV7ssk.js
25
```

The sqrt() method computes the square root of a specified number and returns it

```
let number = Math.sqrt(4);
console.log(number);
```

```
node /tmp/6ENxHqGWMY.js
2
```

JavaScript String method Math.abs() and Math.random()

The abs() method finds the absolute value of the specified number (without any sign) and returns it.

```
let number= Math.abs(-2);
console.log(number);
```

```
node /tmp/Mjdgn7d8Cr.js
2
```

The Math.random() function returns a floating-point, pseudo-random number between **0** (inclusive) and **1** (exclusive).

```
let randomNumber = Math.random()*10
console.log(randomNumber)
```

```
node /tmp/vLBwWfhdK3.js
4.051126874036138
```

JavaScript Number method isNaN()

The isNaN() function checks if a value is **NaN (Not-a-Number)** or not.

```
let number = NaN;
let number2 = 1;
let result = isNaN(number);
let result2 = isNaN(number2);
console.log("Is number a NaN ? ->", result);
console.log("Is number a NaN ? ->", result2);
```

```
node /tmp/TZgV5YfWrI.js
Is number a NaN ? -> true
Is number a NaN ? -> false
```

SOFT CLUB

"YOUR FUTURE IS CREATED BY WHAT

YOU DO { TODAY }

NOT ~~TOMORROW~~"

Thanks