PDP UNIVERSITY

Website development

Lecture 10

String

String represents text

String can be written inside the followings: “”, ‘’ or ``:

```
const a = “Hello”;
const a = 'Hello';
const a = `Hello`;
```

Letters of string can be accessed using [index] or charAt():

```
const a = 'Hello';
console.log(a[0]);
console.log(a.charAt(0));
```

String is immutable

String is immutable. Thus, all string methods return new string:

```
const a = 'Hello';
a[0] = “B”; // does not change the string
```

Get part of a string

The following 3 functions can take part of a string:

```
slice(startIndex, endIndex) – can accept negative values
substing(startIndex, endIndex) – does NOT accept negative values
substr(startIndex, length)
```

replace()

`replace()` – replaces some part of a string with another :

```
let text = "Salom Dunyo";
let newText = text.replace("Dunyo", "World");
```

split()

`split()` – converts a string into an array based on a specific letter:

```
let text = "a,b,c,d,e,f";
const arr = text.split(",");
```

includes()

`includes()` – checks whether a string contains a given string:

```
let text = "a,b,c,d,e,f";
let isExist = text.includes(“b"); // true
```

Regular Expression (RegExp)

RegExp provides more efficient ways for searching and replacing characters in string.

RegExp should be written inside / /: /stringToSearch/

test()

`test()` – checks whether a string contains a given string.

Returns boolean value:

```
let text = "apple, peach, lemon";
let stringToSearch = /peach/;
let result = stringToSearch.test(text); // true
```

search()

`search()` – checks whether a string contains a given string. Returns index (-1 if not found):

```
let text = "apple, peach, lemon";
let stringToSearch = /peach/;
let result = text.search(stringToSearch); // 7
```

replace()

`replace()` – searches for a string inside another string and replaces it:

```
let text = "Apple is red. Apple is sweet";
let result = text.replace(/Apple/, "Tomato");
// result = Tomato is red. Apple is sweet
```

To replace all the instances use "g" after RegExp

```
let result = text.replace(/Apple/g, "Tomato");
// result = Tomato is red. Tomato is sweet
```

To ignore the case use "i" after RegExp

```
let result = text.replace(/apPlE/gi, "Tomato");
// result = Tomato is red. Tomato is sweet
```

replace()

[] – use it to search for a range of characters:

```
[A-Z] - letters from A to Z
[a-z] - letters from a to z
[A-z] - letters from A to z
[^abc] - letters other than a, b, or c
[^A-Z] - except upper case
[^a-z] - except for lowercase letters
```

Thank you for your attention