```
// ! String Methods //

// * 0 - Methods Without Cases Definitions
// .length - The length of the string

// .trim() - Remove the white spaces from the string

// .trimStart() - Remove the white spaces from the string Only At the Start

// .trimEnd() - Remove the white spaces from the string Only At the End

// .concat() - add new strings to the end Ex: str.concat("," , " Hello ", "," , "World!")
// Expected output: str,Hello,world!  can concat with all dataTypes

// .toUpperCase() - Convert string to upper case

// .toLowerCase() - Convert string to lower case

// .padStart(The Final Length of string, what do you want to put in the white spaces) - to
increase the size of string with add other strings at the start
//  Note: 0 ,1 in the Final Length is the same and return the string without any increase;\

// .padEnd() - Same as .padStart but at the End

// * 1 - slice method
// const str = "The quick brown fox jumps over the lazy dog.";

// console.log(str.slice(31));
// // Expected output: "the lazy dog."

// console.log(str.slice(4, 19));
// // Expected output: "quick brown fox"

// console.log(str.slice(-4));
// // Expected output: "dog."

// console.log(str.slice(-9, -5));
// // Expected output: "lazy"

// console.log(str.slice());
// // Expected output: "The quick brown fox jumps over the lazy dog."

// * 2 - substr
// const str = "Mozilla";
```

```
// console.log(str.substr());
// // Expected output: "Mozilla"

// console.log(str.substr(1, 2));
// // Expected output: "oz"

// console.log(str.substr(2));
// // Expected output: "zilla"

// console.log(str.substr(-3));
// // Expected output: "lla"

// console.log(str.substr(-3, -1));
// //Expected output: "";

// console.log(typeof str.substr(-3, -1));
// // Expected output: string;

// * 3 - substring
// const str = 'Mozilla';

// console.log(str.substring());
// // Expected output: "Mozilla"

// console.log(str.substring(1, 3));
// // Expected output: "oz"

// console.log(str.substring(2));
// // Expected output: "zilla"

// console.log(str.substring(-1));
// // Expected output: "Mozilla"

// console.log(str.substring(-3,-1));
// // Expecetd output: ""

// * 4 - charAt();
// const sentence = "The quick brown fox jumps over the lazy dog.";

// console.log(sentence.charAt());
// // Expected output: T"

// console.log(sentence.charAt(4));
// // Expected output: q"
```

```
// console.log(sentence.charAt(-4));
// // Expected output: ""

// * 5 - at()
// const sentence = 'The quick brown fox jumps over the lazy dog.';

// console.log(sentence.at());
// // Expected output: "T"

// console.log(sentence.at(4));
// // Expected output: "q"

// console.log(sentence.at(-4));
// // Expected output: "d"

// * 6 - charCodeAt()
// const str = "hello";

// console.log(str.charCodeAt());
// // Expected output: code of the first element

// console.log(str.charCodeAt(4));
// // Expected output: code of element with index 4

// console.log(str.charCodeAt(-4));
// // Expected output: NaN

// * 7 - repeat()
// const mood = "Happy! ";

// console.log(mood.repeat(3));
// // Expected output: "Happy! Happy! Happy! "

// console.log(mood.repeat(1));
// // Expected output: "Haapy! "

// console.log(mood.repeat(0));
// // Expected output: ""

// * 8 - replace()
// const paragraph = "I think Ruth's dog is cuter than your dog!";

// console.log(paragraph.replace("Ruth's", "my"));
// // Expected output: "I think my dog is cuter than your dog!"
```

```
// * 9 - replaceAll()
// const paragraph = "I think Ruth's dog is cuter than your dog!";

// console.log(paragraph.replaceAll('dog', 'monkey'));
// // Expected output: "I think Ruth's monkey is cuter than your monkey!"

// * 10 - toString()
// const stringObj = new String('foo');

// console.log(stringObj);
// // Expected output: String { "foo" }

// console.log(stringObj.toString());
// // Expected output: "foo"

// * 11 - split()
// const str = 'The quick brown fox jumps over the lazy dog.';

// const words = str.split(' ');
// console.log(words[3]);
// // Expected output: "fox"

// const chars = str.split('');
// console.log(chars[8]);
// // Expected output: "k"

// const strCopy = str.split();
// console.log(strCopy);
// // Expected output: Array ["The quick brown fox jumps over the lazy dog."]

// const strCopy = str.split();
// console.log(strCopy[1]);
// // Expected output: undefined

// ! Array Methods

// * 0 - Methods Without Cases Definitions
// .length - The length of the Array

// .concat() - add new arrays to the end Ex: array.concat(array1 , array2 , .....)
// Expected output: all the elements of the arrays

// * 1 - slice()
// const animals = ['ant', 'bison', 'camel', 'duck', 'elephant'];
```

```
// console.log(animals.slice(2));
// // Expected output: Array ["camel", "duck", "elephant"]

// console.log(animals.slice(2, 4));
// // Expected output: Array ["camel", "duck"]

// console.log(animals.slice(1, 5));
// // Expected output: Array ["bison", "camel", "duck", "elephant"]

// console.log(animals.slice(-2));
// // Expected output: Array ["duck", "elephant"]

// console.log(animals.slice(2, -1));
// // Expected output: Array ["camel", "duck"]

// console.log(animals.slice());
// // Expected output: Array ["ant", "bison", "camel", "duck", "elephant"]

// * 5 - find()
// const array1 = [5, 12, 8, 130, 44];

// const found = array1.find((element) => element > 10); function linke lambda function in python

// console.log(found);
// // Expected output: 12

// * 2 - at()
// const array1 = [5, 12, 8, 130, 44];

// console.log(array1.at(4));
// // Expected output: "q"

// console.log(array1.at(-4));
// // Expected output: "d"

// console.log(array1.at());
// // Expected output: "5"

// * 3 - join()
// const elements = ['Fire', 'Air', 'Water'];

// console.log(elements.join());
// // Expected output: "Fire,Air,Water"

// console.log(elements.join(''));
```

```
// // Expected output: "FireAirWater"

// console.log(elements.join('-'));
// // Expected output: "Fire-Air-Water"

// * 4 - toString()
// const array1 = [1, 2, "a", "1a"];

// console.log(array1.toString());
// // Expected output: "1,2,a,1a"

// * 5 - splice() : don't return a new array and modify the current array
// const months = ['Jan', 'March', 'April', 'June'];
// months.splice(1, 0, 'Feb');
// // Inserts at index 1
// console.log(months);
// // Expected output: Array ["Jan", "Feb", "March", "April", "June"]

// months.splice(4, 1, 'May');
// // Replaces 1 element at index 4
// console.log(months);
// // Expected output: Array ["Jan", "Feb", "March", "April", "May"]

// * 6 - toSpliced(): like splice() but return a new array with the modifying but the current
array still
// const months = ["Jan", "Mar", "Apr", "May"];

// months.toSpliced(1, 0, "Feb");
// console.log(months);
// Expected output : Array ["Jan", "Mar", "Apr", "May"]

// const mu = months.toSpliced(1, 0, "Feb");
// console.log(mu);
// Expected output : Array ["Jan", "Feb","Mar", "Apr", "May"]

// * 7 - push()
// const animals = ["pigs", "goats", "sheep"];

// const count = animals.push("cows");
// console.log(count);
// // Expected output: 4
// console.log(animals);
// // Expected output: Array ["pigs", "goats", "sheep", "cows"]

// animals.push("chickens", "cats", "dogs");
```

```
// console.log(animals);
// // Expected output: Array ["pigs", "goats", "sheep", "cows", "chickens", "cats", "dogs"]

// animals.push(); : Don't cause errors
// console.log(animals);
// // Expected output: Array ["pigs", "goats", "sheep", "cows", "chickens", "cats", "dogs"]

// * 8 - pop()
// const plants = ["broccoli", "cauliflower", "cabbage", "kale", "tomato"];

// console.log(plants.pop());
// // Expected output: "tomato"

// console.log(plants);
// // Expected output: Array ["broccoli", "cauliflower", "cabbage", "kale"]

// plants.pop();

// console.log(plants);
// // Expected output: Array ["broccoli", "cauliflower", "cabbage"]

// console.log(plants.pop(2)); Don't cause errors and make the same result of pop()

// * 9 - shift():like pop() in all cases, but it's remove the element at the start

// * 10 - unshift():like push() in all cases, but it's add the elements at the start
```