Frontendβ → JavaScript → Data types and operations → Slicing

Theory: Slicing

13 minutes 0 / 5 problems solved

Skip this topic Start practicing

646 users solved this topic. Latest completion was **2 days ago**.

JavaScript provides many features for working with arrays. One of the things you can do with a sequence is retrieving some part of it, which implies getting elements by their indexes. You can do this with the `slice()` method.

§1. Getting sections of sequences

The `slice()` method does not change the source array, but returns a copy of the array containing copies of the elements cut from the source array. Here is what its syntax looks like:

```
arr.slice(begin, end);
```

As you can see, this method accepts two parameters:

- **begin** – instead of this word, specify the index of the element that will be the beginning of the extracted array partition.
- **end** – here, specify the index of the array element by which the extraction will end. This element will not be included in the copy of the array.

Do not forget that indexing in the array starts from zero.

These parameters are optional. If they are not specified, `slice()` will simply create a copy of the array you're working with:

```
let sheep = ["Dolly", "Polly", "Molly"];

console.log(sheep.slice()); // ["Dolly", "Polly", "Molly"];
```

If you do not specify the first parameter, the array will be copied from `0` index. Similarly, if you do not specify the second one, all the elements after the index from which they started will be copied.

If the first parameter is longer than the sequence length, an empty array will return:

```
let nums = [1, 2, 3];

console.log(nums.slice(10)); // []
```

Let's try to work with the two specified parameters:

```
let planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter"];

console.log(planets.slice(1, 4)); // ["Venus", "Earth", "Mars"]
```

§2. Negative index values

Indexes can also be **negative.** If you specify a negative index in the first parameter, it will shift the end of the sequence. Alternatively, it can be called negative offset. For example, `slice(-4)` call will extract the last four elements of the sequence:

```
let fibNums = [0, 1, 1, 2, 3, 5, 8, 13, 21];

console.log(fibNums.slice(-4)); // [5, 8, 13, 21]
```

You can specify the second index as negative: it will also mean a shift from the end of the sequence:

Current topic:

Slicing

Topic depends on:

✕ Arrays

✕ Functions

Table of contents:

↑ Slicing
§1. Getting sections of sequences
§2. Negative index values
§3. Strings
§4. Conclusion
Feedback & Comments

```
let fibNums = [0, 1, 1, 2, 3, 5, 8, 13, 21];

console.log(fibNums.slice(3,-2)); // [2, 3, 5, 8]
```

§3. Strings

In JS, the `slice()` method can also be applied to strings for finding substrings. The principle is similar to the interaction with arrays:

```
let str = "lighthouses";

console.log(str.slice(-6, -1)); // house
```

§4. Conclusion

With `slice()`, you can get any part of an array or a string. Remember that indexes are optional in this method's syntax, so all of them have a default value. Also, keep it in mind that you can set them to negative if the situation asks for it.

Report a typo

66 users liked this theory. **1** didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Start practicing

Comments (4) Hints (0) Useful links (0) Show discussion