JavaScript Stack Implemented using arrays. Day 23rd

A stack is a data structure that holds a list of elements. A stack works based on the LIFO principle i.e., Last In, First out, meaning that the most recently added element is the first one to remove.

A stack has two main operations that occur only at the top of the stack: push and pop. The push operation places an element at the top of stack whereas the pop operation removes an element from the top of the stack.

The name stack comes from the analogy to a set of physical items e.g., DVD disc, books, stacked on top each other

A stack has many applications. For example, the simplest one is to reverse a word. To do it, you push a word into the stack, letter by letter, and pop the letters from the stack.

The other applications of the stack are “undo” mechanism in text editors, syntax parsing, function call, and expression conversion (infix to postfix, infix to prefix, postfix to infix, and prefix to infix).

JavaScript arrays type provides the `push()` and `pop()` methods that allow you to use an array as a stack.

Stack Methods in JavaScript.

a). **push() method**

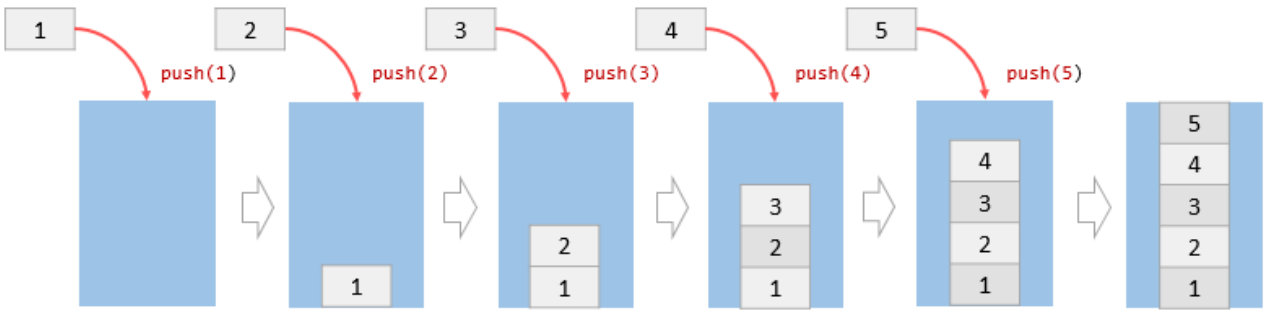
The push() method allows you to add one or more elements to the end of the array. The push() method returns the value of the length property that specifies the number of elements in the array.

If you consider an array as a stack, the push() method adds one or more element at the top of the stack.

The following example creates an empty array named stack and adds five numbers, one by one, at the end of the stack array. It is like to push each number into the top of the stack.

```
let stack = [];
stack.push(1);
console.log(stack);
stack.push(2);
console.log(stack);
stack.push(3);
console.log(stack);
stack.push(4);
console.log(stack);
stack.push(5);
console.log(stack);
```

Try running the above snippets and see the output.



The above diagram demonstrates what happens in the above snippets. Initially, the stack is empty. Each time, we call the push() method to add a number to the stack. After 5 calls, the stack has 5 elements.

Note that the push() method also allows you to add multiple items to the end of the array at a time.

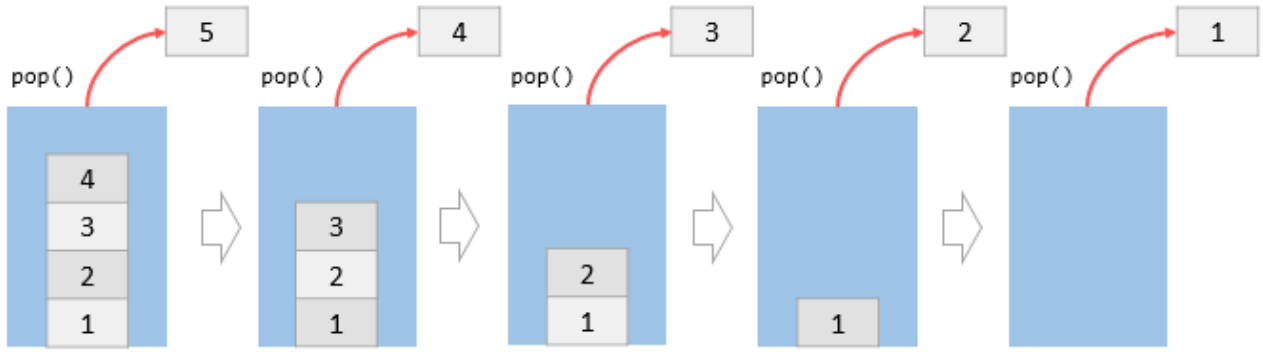
b). pop() method.

The pop() method removes the element at the end of the array and returns the element to the caller. If the array is empty, the pop() method returns undefined.

The following example shows how to pop elements from the top of the stack using the pop() method.

```
console.log(stack.pop());
console.log(stack);
console.log(stack.pop());
console.log(stack);
console.log(stack.pop());
console.log(stack);
 console.log(stack.pop());
 console.log(stack);
console.log(stack.pop());
console.log(stack);
```

Try running the above snippets and see the output.



This image illustrates whats happens in the snippets.

Initially, the stack has 5 elements. The pop() method removes the element at the end of the array i.e., at the top of the stack one at a time. After five operations, the stack is empty.

Use Case:

Reverse a string using a JavaScript stack.

The following example shows you how to reverse a string using a stack.

```
function reverse(str) {
 let stack = [];
for (let i = 0; i < str.length; i++) {
stack.push(str[i]);
}
let reverseStr = '';
 while (stack.length > 0) {
 reverseStr += stack.pop();
 } return reverseStr;
}
console.log(reverse('JavaScript Stack'));
```

Trying ruing the program above and the output should be: kcatS tpircSavaJ

Note that:

The reverse() function accepts a string argument and returns its reversed version with the following logic:

1. First, loop through the str and push each letter into the stack array.

2. Second, pop each letter from the stack and construct the reversed string.

Best Wishes **Lux Tech Academy**.