Types, Values, Operators

Part 3

Nested loops

```
for( ;condition1; ) {
      statement1;
      for( ;condition2; )
{
            statement2;
      }
}
```

```
while(condition1) {
      statement1;
      while(condition2) {
            statement2;
      }
}
```

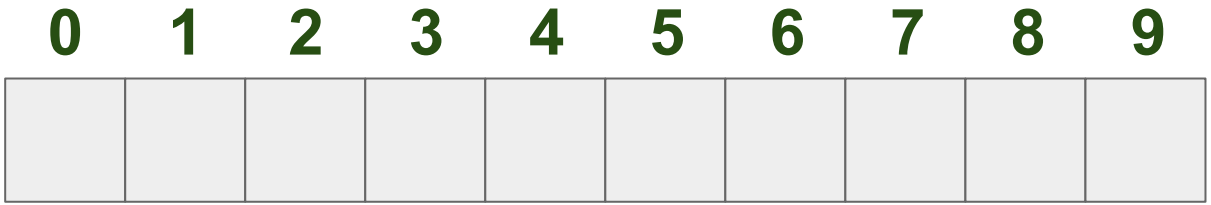
Nested loops

```
1 for(var i = 0; i < 3; i++){
2     console.log(i);
3     for(var j = 0; j < 3; j++) {
4         console.log(j);
5     }
6 }
```

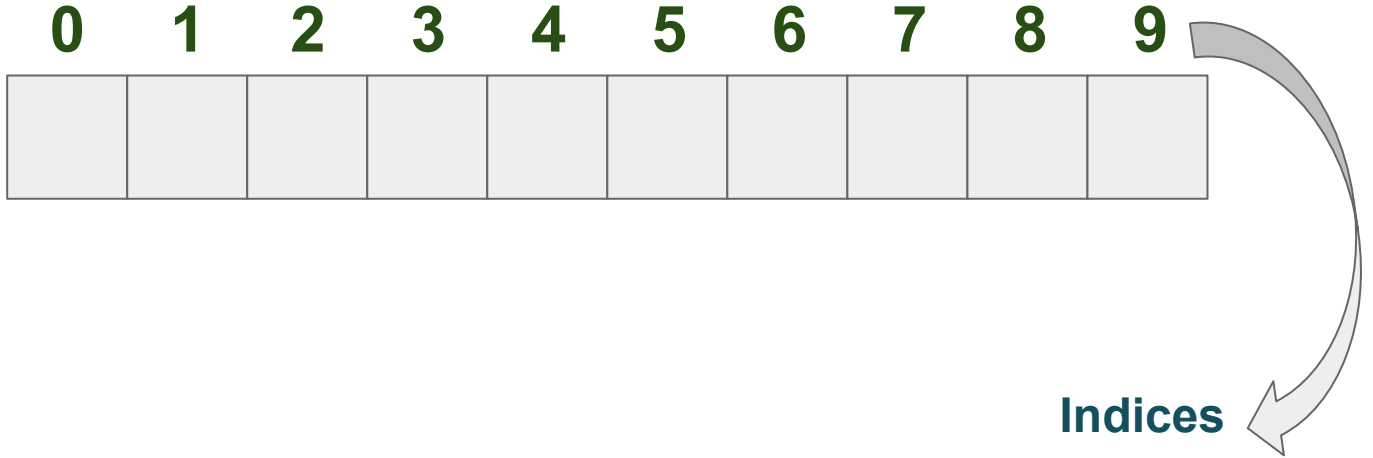
EXAMPLE

Arrays

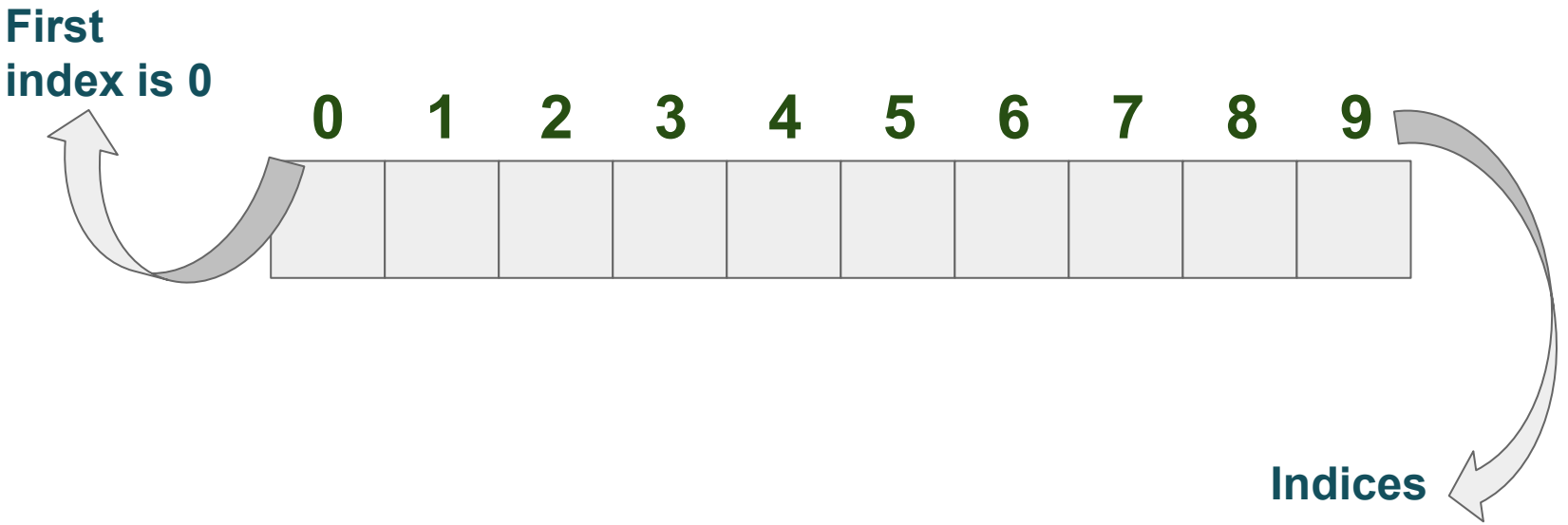
Arrays



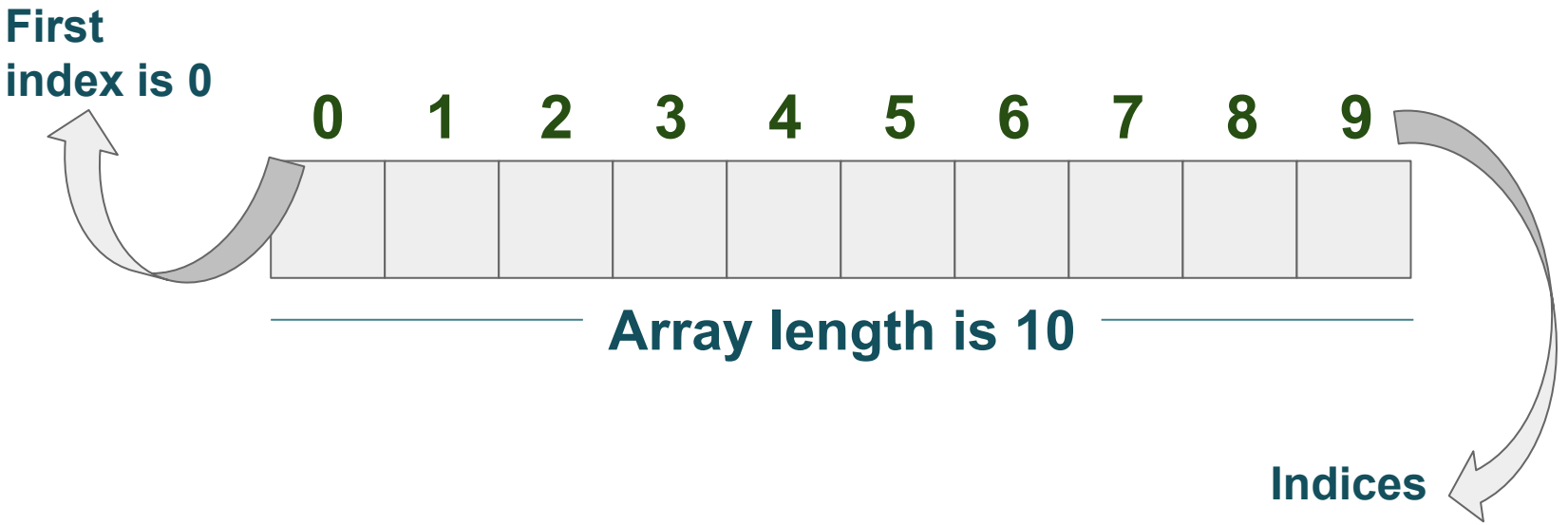
Arrays



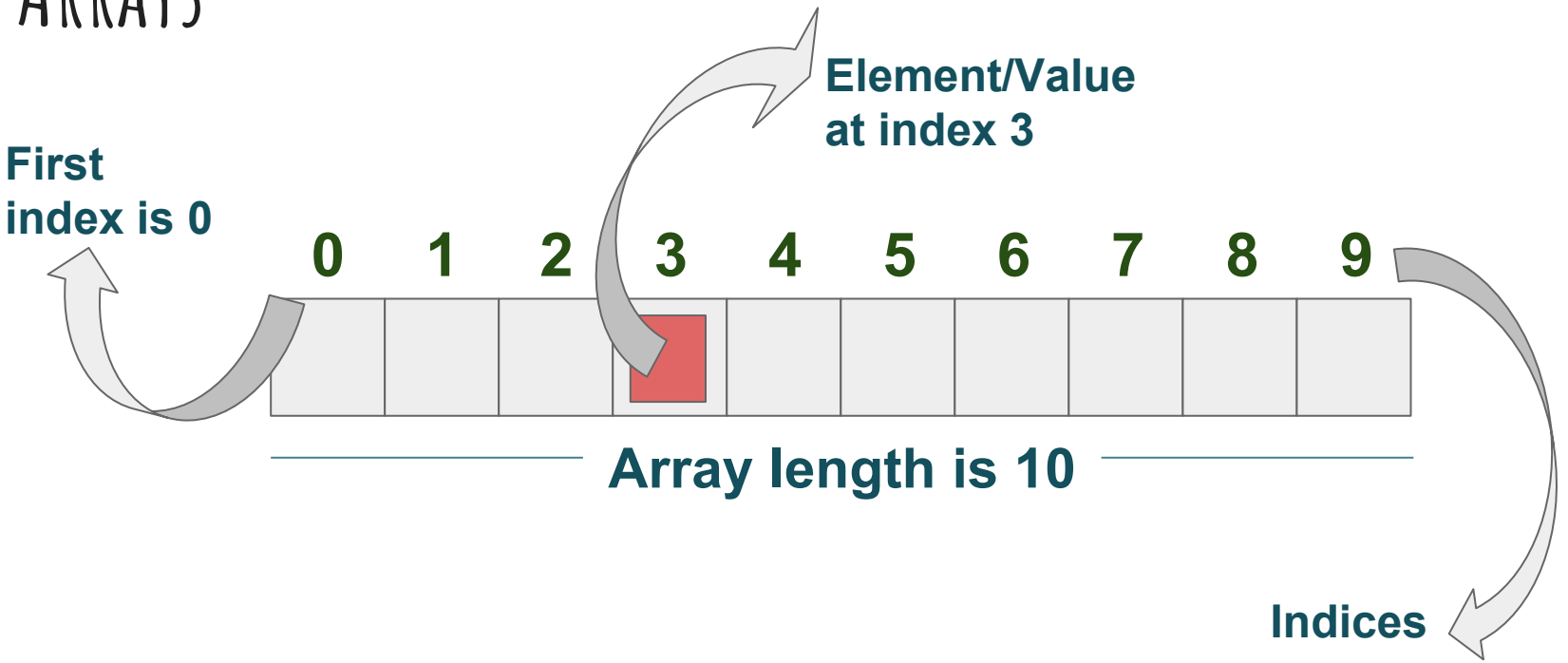
Arrays



Arrays



Arrays


Element/Value
at index 3
First
index is 0
0 1 2 3 4 5 6 7 8 9
Array length is 10
Indices

Arrays: Declare, Get and Change

```
var arr = [10, 20, 30, 40, 50, 60];
var l = arr.length;
var element0 = arr[0];
var element4 = arr[2];
arr[1] = 25;
```

EXAMPLE

Arrays: Pop and Push

```
var arr = [‘ok’, ‘wow’, ‘hi’, ‘yeah’];
var last = arr.pop();
var newLast = ‘bye’;
arr.push(newLast);
```

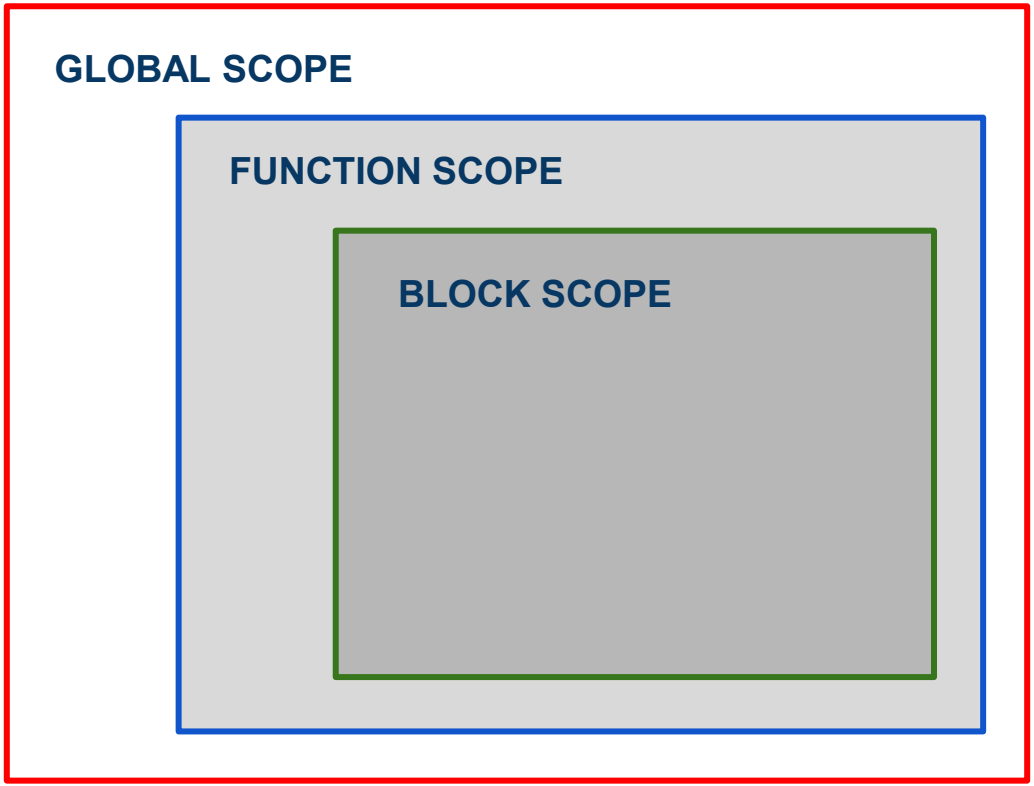
EXAMPLE

Strings

Strings can behave them like arrays.
But there are differences!

EXAMPLE

SCOPE



EXAMPLE

Let, const

Declare variables that are limited in scope to the block.

EXAMPLE

Var

The scope of a variable declared with var is its current execution context, [which is either the enclosing function or,] for variables declared outside any function, global.

EXAMPLE

Hoisting

Variable declarations are processed before any code is executed.

Declaring a variable anywhere in the code is equivalent to declaring it at the top.

This also means that a variable can appear to be used before it's declared.

This behavior is called "hoisting".

Hoisting

let will hoist the variable to the top of the block. However, referencing the variable in the block before the variable declaration results in a ReferenceError.

Thank you