Caltech | Center for Technology & Management Education

# Full Stack Java Developer

# JavaScript

# Objects

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

# Learning Objectives

By the end of this lesson, you will be able to:

- Understand the ways to create objects and discuss various operations
- Define the factory function and write its syntax
- Incorporate factory functions in the code
- Explain the enumerating properties of an object
- Learn how an object is cloned

# Learning Objectives

By the end of this lesson, you will be able to:

- Analyze the dynamic nature of objects
- Understand the concepts of garbage collection in JavaScript
- Learn Math and String
- List the JavaScript string methods
- Understand template literals and how they are used

# A Day in the Life of a Full Stack Developer

You are working in an organization and have been assigned a project. Each developer has been assigned a module in the project. You have been asked to analyze the source code of the project and come up with ways to optimize the code.

You decide to make use of dynamic objects and enumerating objects.

To do so, you need to understand the ways to create objects and analyze the dynamic nature of objects.

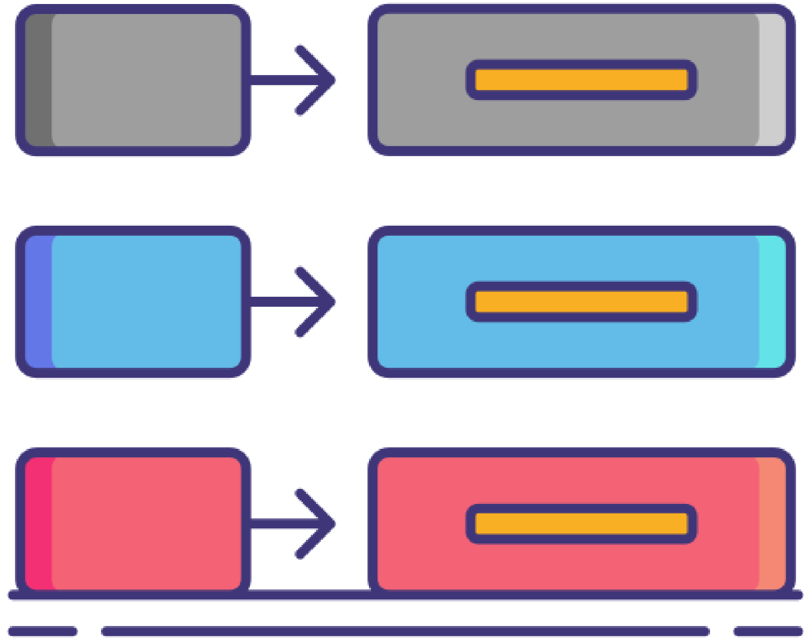
TECHNOLOGY

## What Are Objects?

# Object

In JavaScript, the object is an unordered standalone entity having property and type.

Object stores the data as a key-value pair.

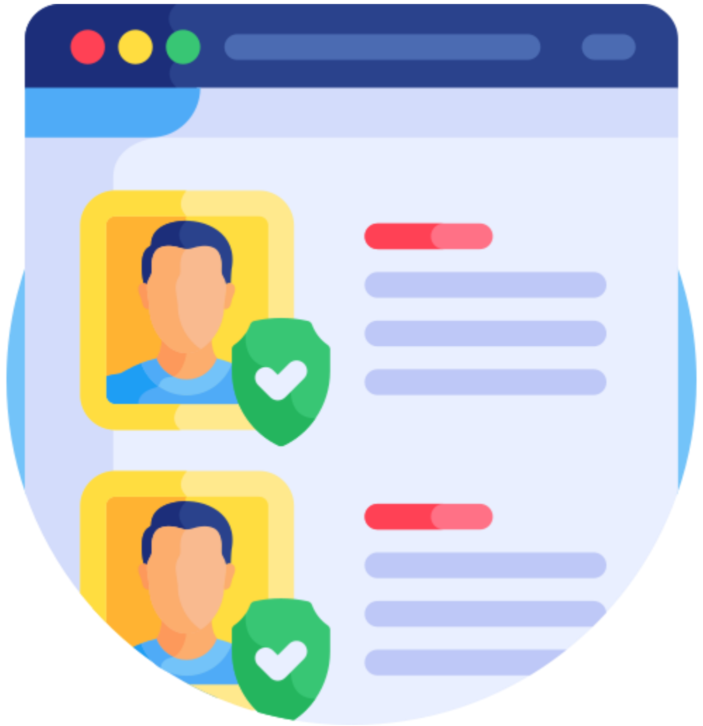


**Syntax:**

```
key : value (key and value separated by a colon)
```

# Object

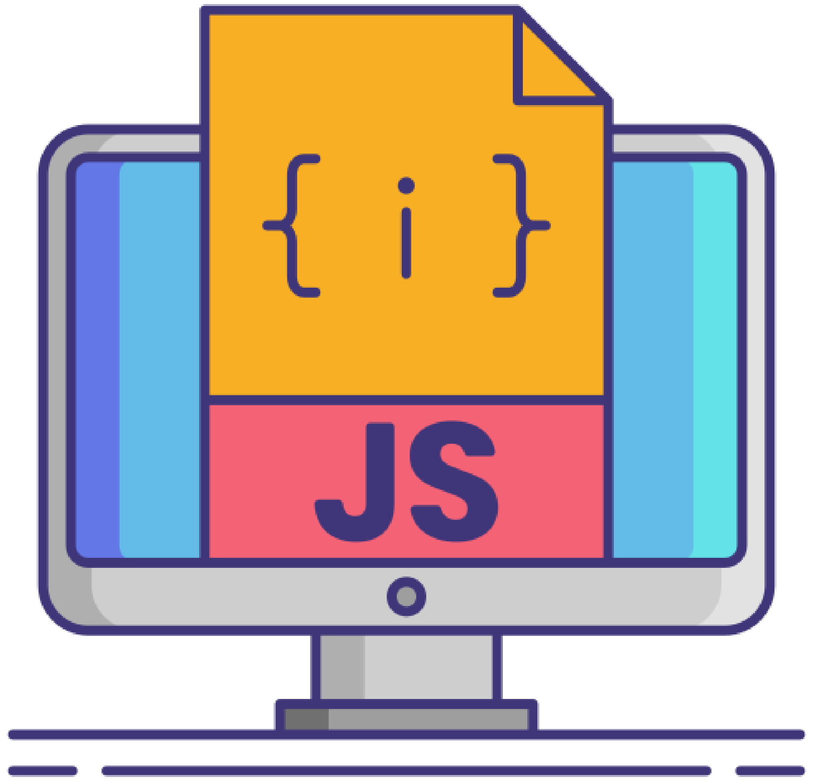
Example:



```
const user = {name:"Mike Hary", email:"mike.hary@example.com", age:30, weight:60};
```

# Enumerating Properties

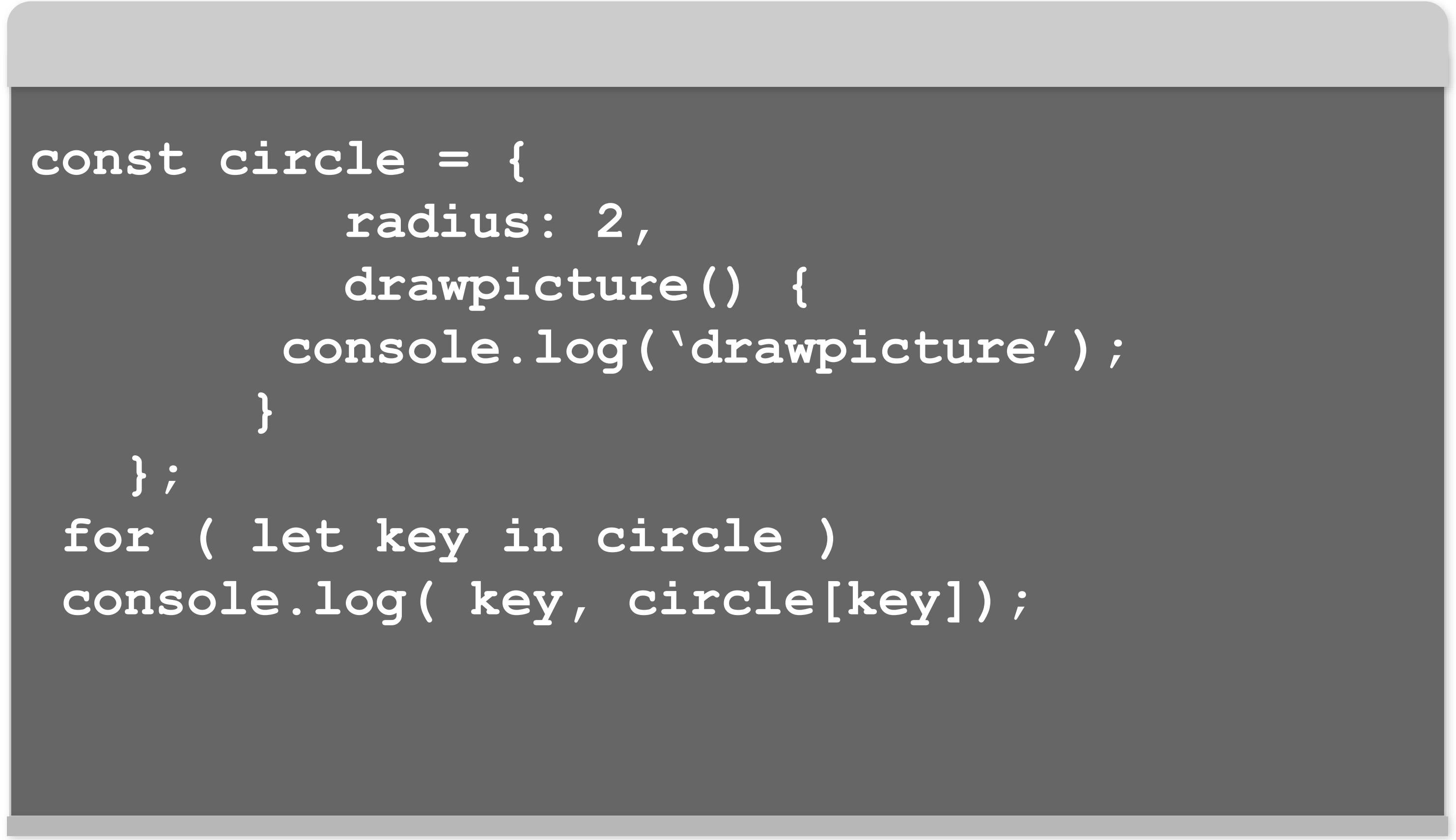
In JavaScript, all reference types are objects.



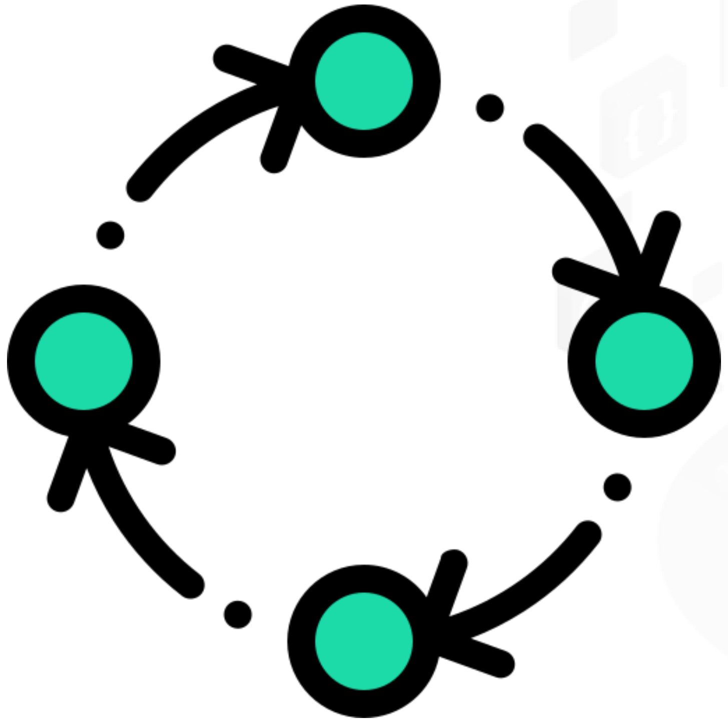
With some approaches or procedures, one may enumerate the characteristics of any given object.

# For-in Loop

For-in loop is used to enumerate all the properties and methods in an object.

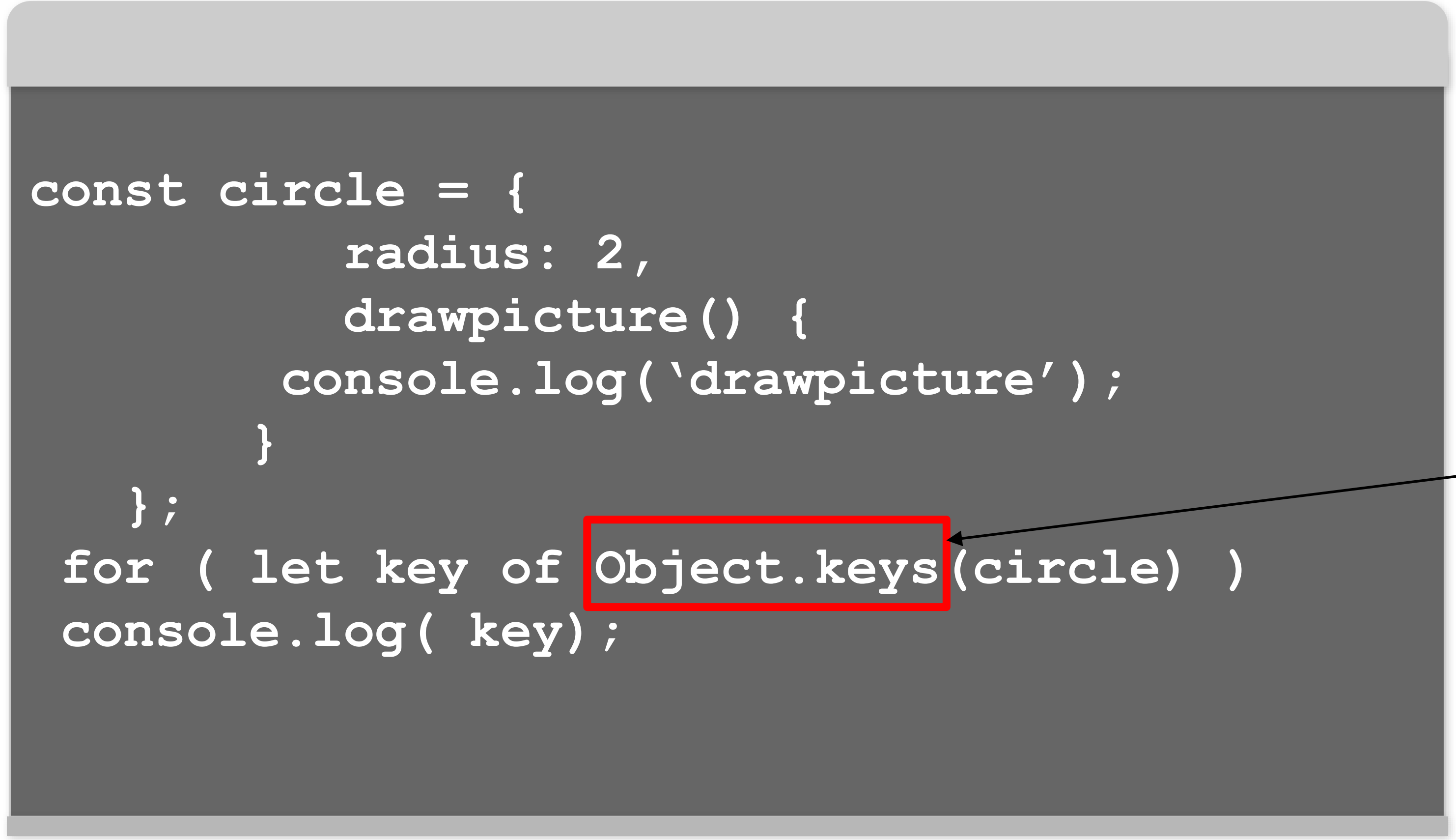


```
const circle = {
          radius: 2,
          drawpicture() {
        console.log(‘drawpicture’);
       }
   };
 for ( let key in circle )
 console.log( key, circle[key]);
```



# For-in Loop

object.keys are used in conjunction with a for-in loop to avoid an error message.

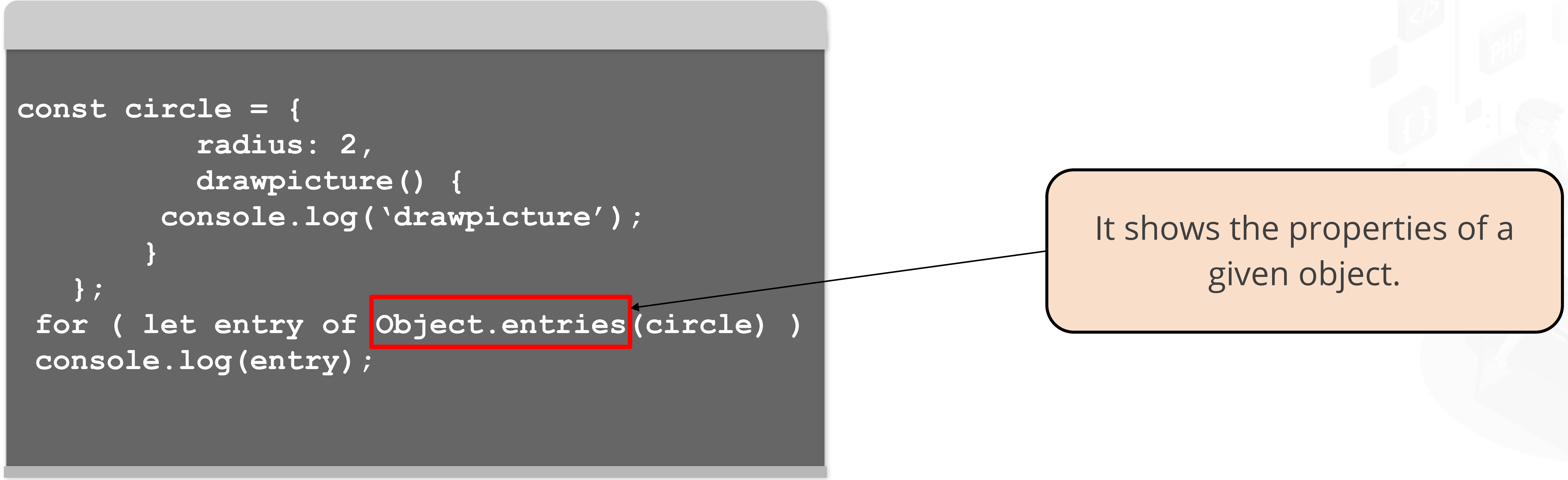


```
const circle = {
          radius: 2,
          drawpicture() {
        console.log('drawpicture');
       }
   };
 for ( let key of Object.keys(circle) )
 console.log( key);
```

The loop will return the key and the method in our circle object in the string array.

# Object.keys

object.keys are used to show the properties of a given object.



```
const circle = {
          radius: 2,
          drawpicture() {
        console.log(‘drawpicture’);
       }
   };
 for ( let entry of Object.entries(circle) )
 console.log(entry);
```

It shows the properties of a given object.

# Cloning an Object

Cloning is used to avoid the creation of duplicates of an existing object.

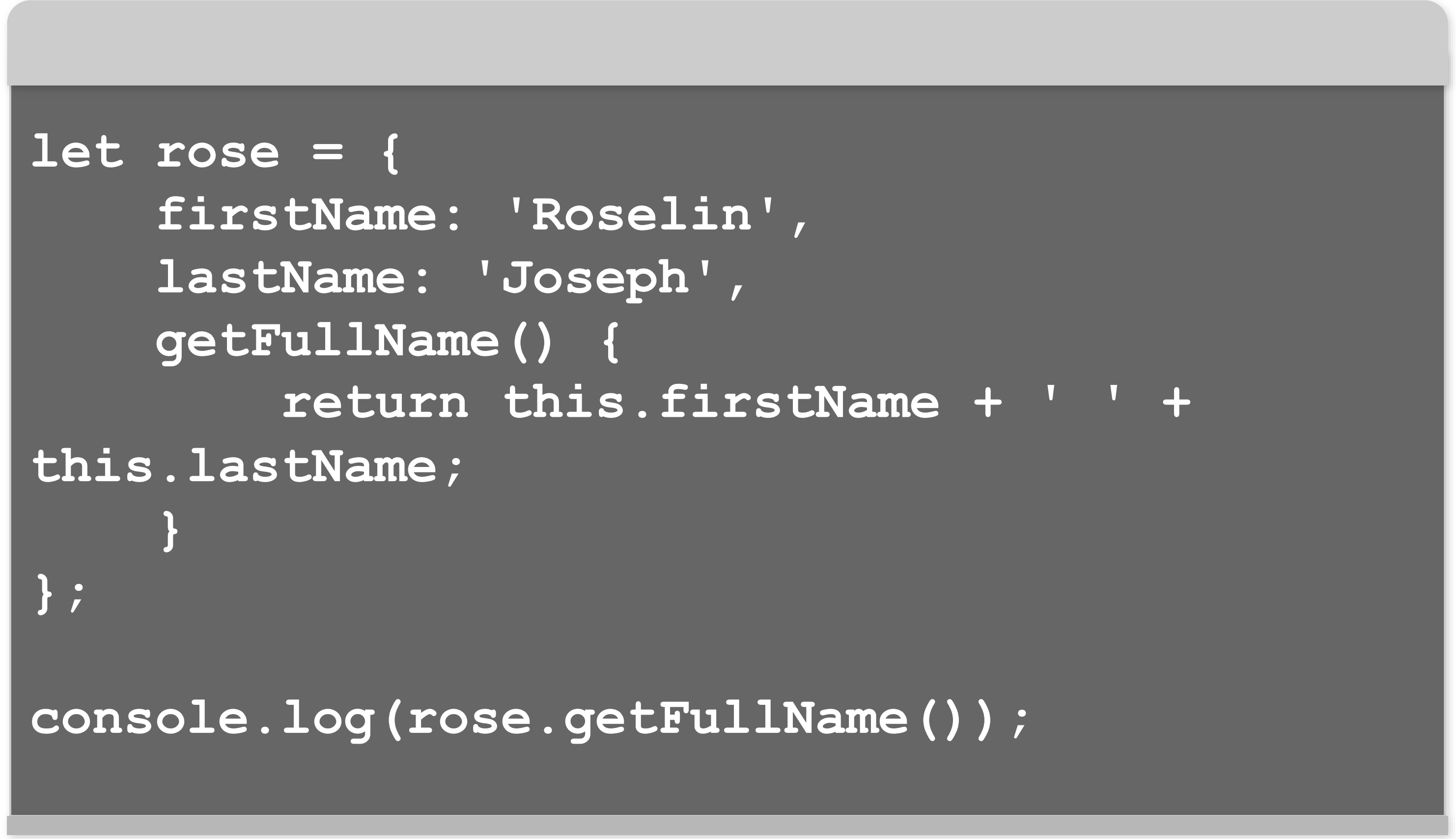
The source object must be JSON-safe when using the JSON method.

# Factory Function

# Factory Function

A factory function is a type of function that returns a new object.

The factory functions create an object and return it.

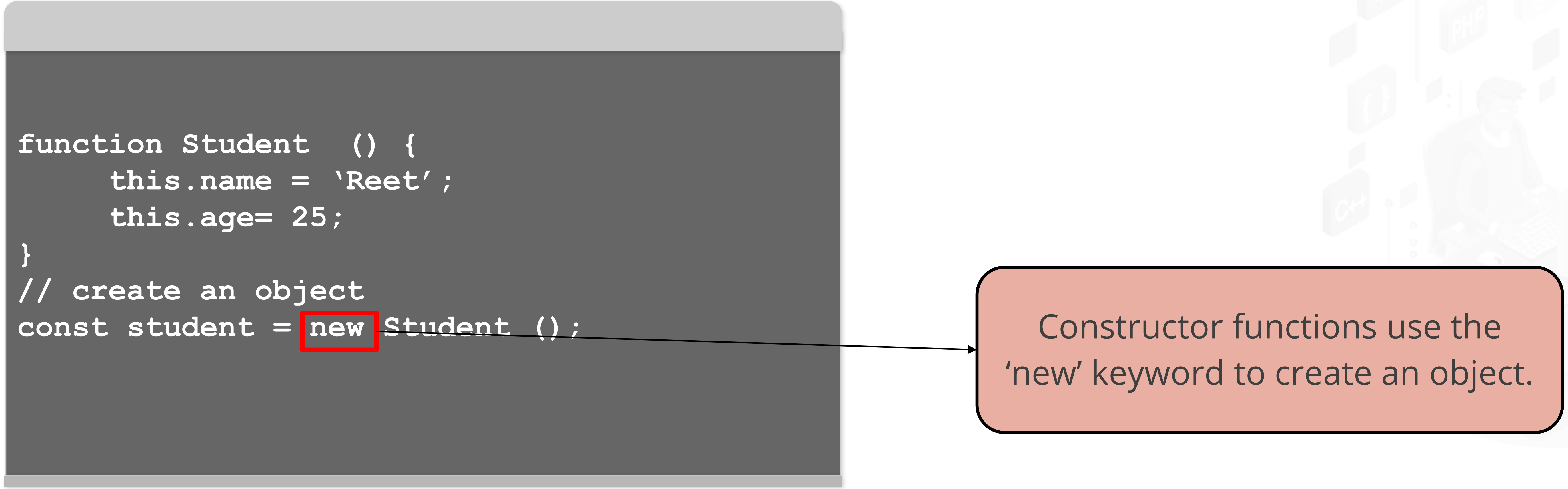


```
let rose = {
    firstName: 'Roselin',
    lastName: 'Joseph',
    getFullName() {
        return this.firstName + ' ' +
this.lastName;
    }
};

console.log(rose.getFullName());
```

# Constructor Function

The constructor function is used to create objects in JavaScript.



```
function Student  () {
      this.name = ‘Reet’;
      this.age= 25;
}
// create an object
const student = new Student ();
```

Constructor functions use the ‘new’ keyword to create an object.

# Constructor Function

The constructor function is used to create multiple similar objects.

```
let vehicle1 = new Vehicle('Honda','Activa')
let vehicle2 = new Vehicle('Bajaj','Super')
```

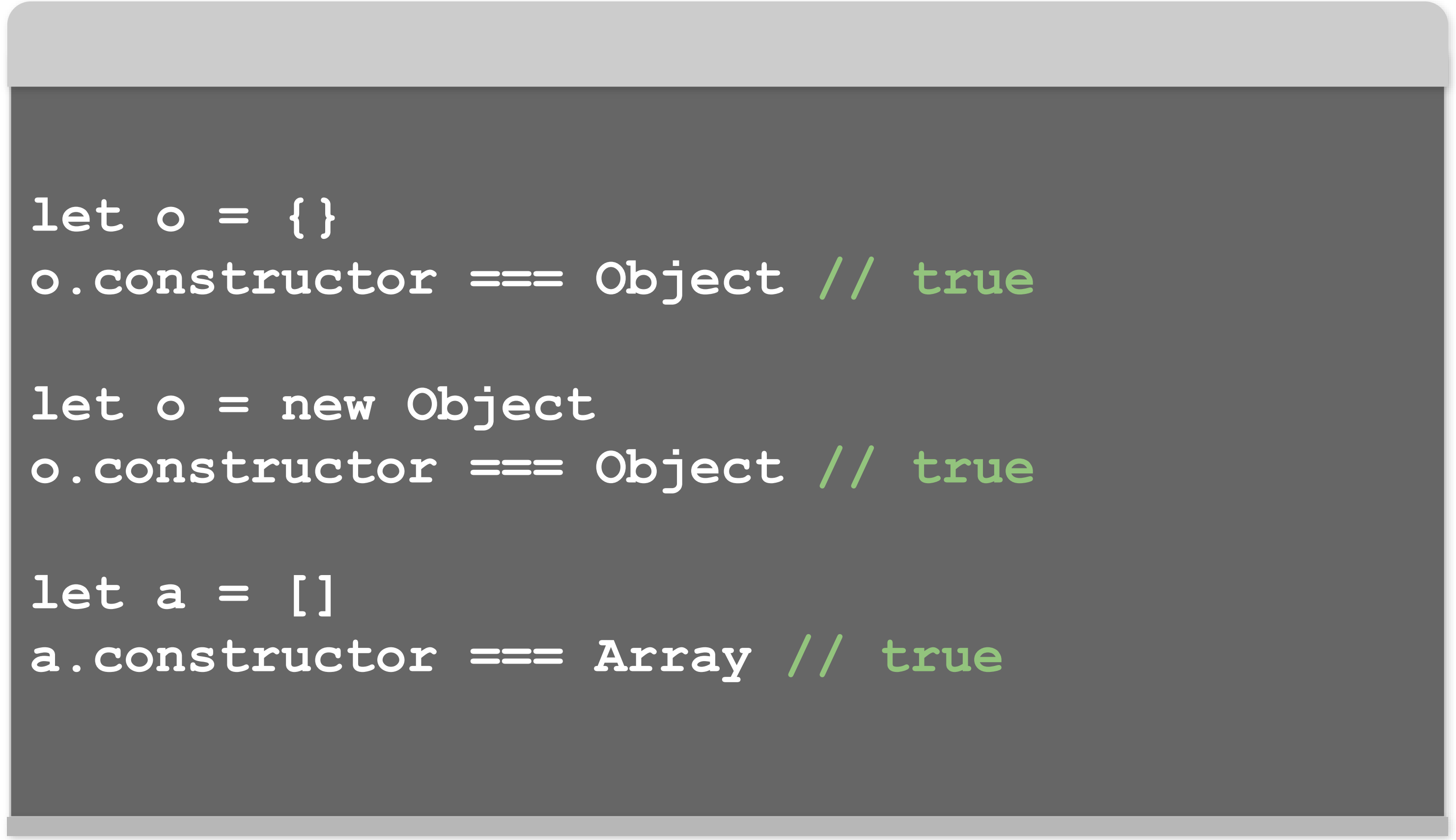
# Constructor Function

The function Object() { [native code] } property returns a reference to the instance object.

- ✓ The constructor value is a reference to the function itself.
- ✓ All objects by default have a constructor property.

# Constructor Function

Syntax:



```
let o = {}
o.constructor === Object // true

let o = new Object
o.constructor === Object // true

let a = []
a.constructor === Array // true
```

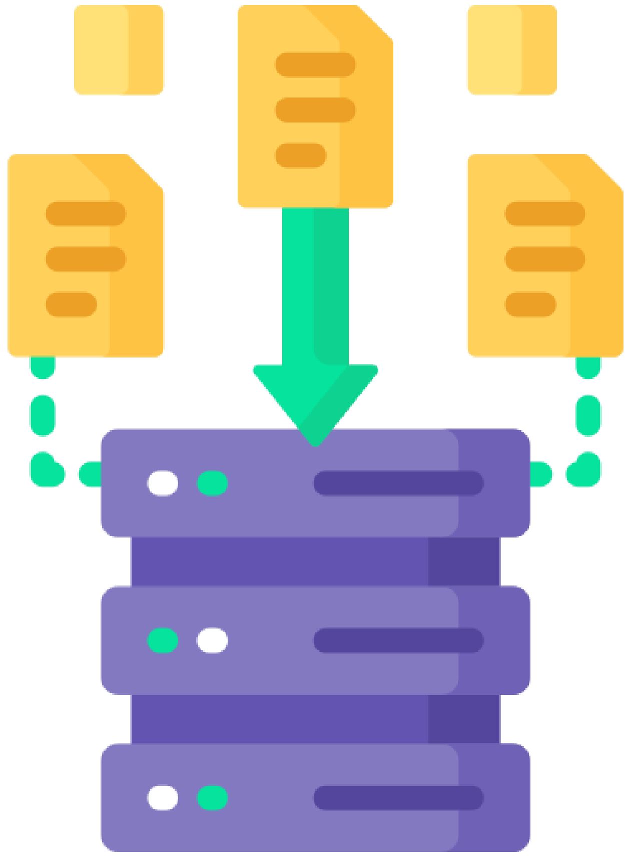
TECHNOLOGY

## Value vs. Reference Types

# Value vs. Reference Types

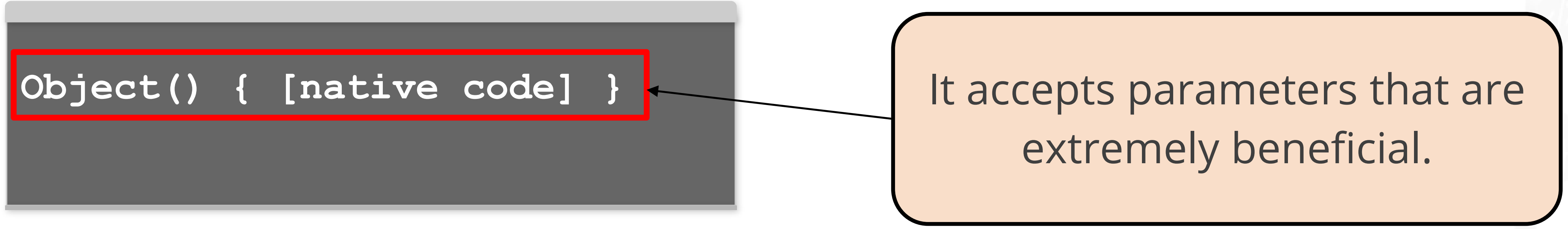
When any data type stores data within its memory allocation, it is referred to as a value type.

- Numeric data types
- Boolean, char, and date
- Structure data
- Enumerations



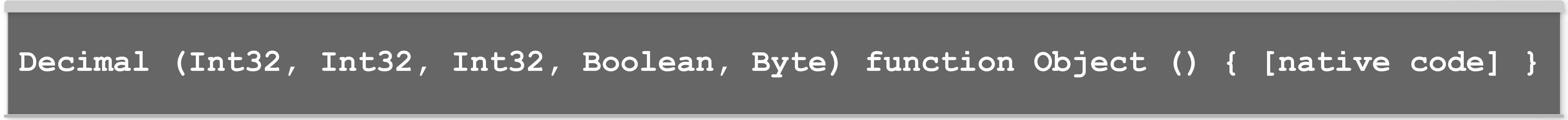
# Value vs. Reference Types

The reserved keyword is used to designate a value type.



```
Object() { [native code] }
```

It accepts parameters that are extremely beneficial.

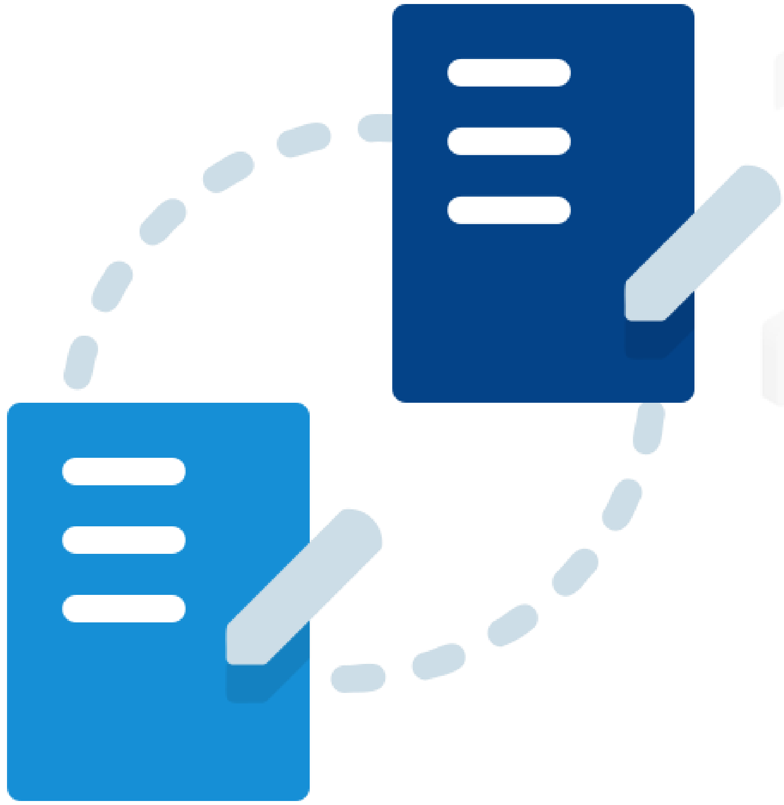


```
Decimal (Int32, Int32, Int32, Boolean, Byte) function Object () { [native code] }
```

# Value vs. Reference Types

Examples of reference types:

1. Strings
2. All arrays
3. Class types
4. Delegates

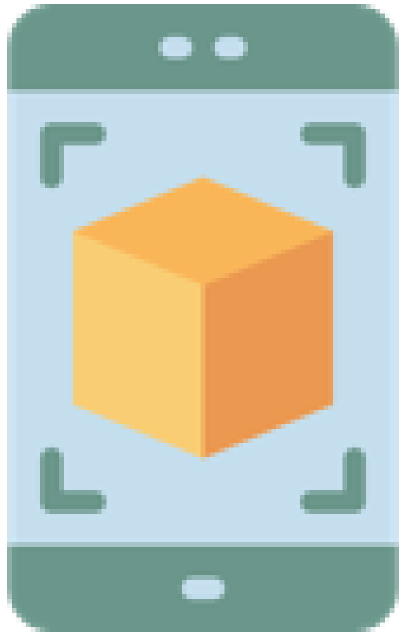


# Garbage Collection

# Garbage Collection

Garbage collection is automatic memory management used by JavaScript.

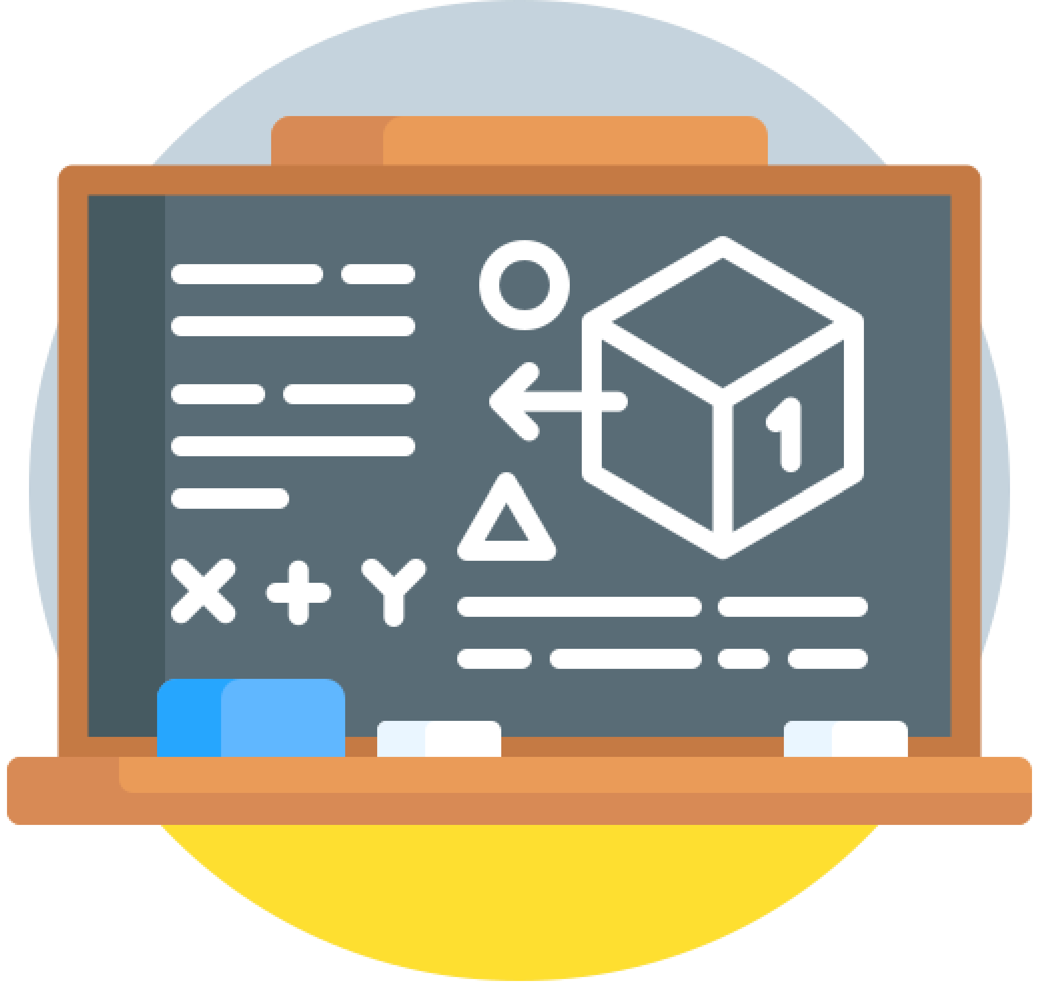
- ✓ It keeps track of memory allocation.
- ✓ It decides reclaiming of memory that is not in use.



# Math Object

# Math Object

The JavaScript math object comes with several constants, functions, and methods to perform a mathematical operation.



# Math Object

Types of math object methods are:

| Method | Description |
|---|---|
| abs() | Returns the absolute value of the given number |
| acos() | Returns the arccosine of the given number in radians |
| asin() | Returns the arcsine of the given number in radians |
| atan() | Returns the arc-tangent of the given number in radians |
| cbrt() | Returns the cube root of the given number |
| ceil() | Returns the smallest integer value greater than or equal to the given number |
| cos() | Returns the cosine of the given number |
| cosh() | Returns the hyperbolic cosine of the given number |

# Math Object

Types of math object methods are:

| Method | Description |
|---|---|
| exp() | Returns the exponential form of the given number |
| floor() | Returns the largest integer value lower than or equal to the given number |
| hypot() | Returns the square root of the sum of the squares of given numbers |
| log() | Returns the natural logarithm of a number |
| max() | Returns the maximum value of the given numbers |
| min() | Returns the minimum value of the given numbers |
| pow() | Returns the value of the base to the power of the exponent |
| random() | Returns the random number between 0 and 1 |

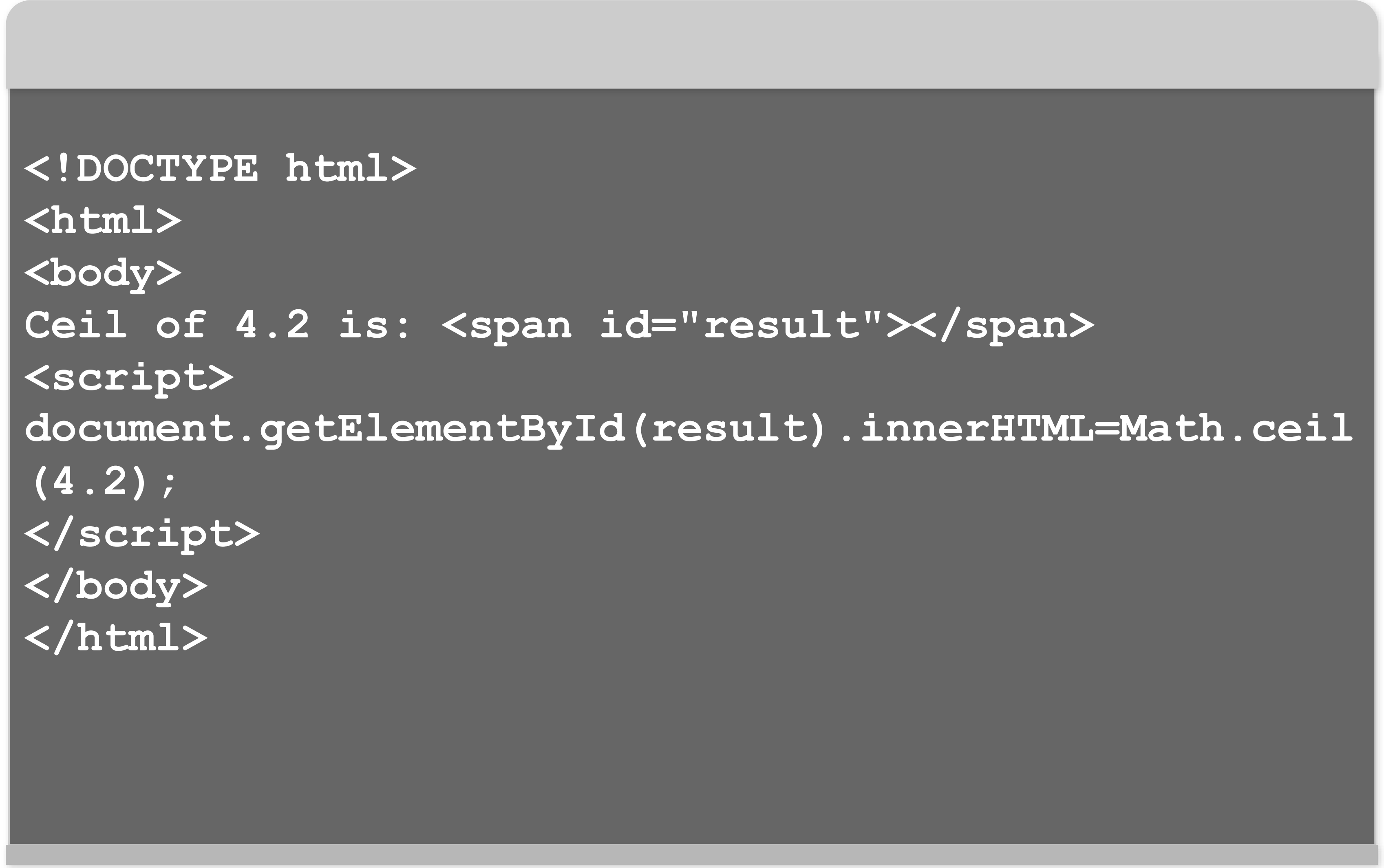
# Math Object

Types of math object methods are:

| Method | Description |
|---|---|
| round() | Returns the closest integer value of the given number |
| sign() | Returns the sign of the given number |
| sin() | Returns the sine of the given number |
| sinh() | Returns the hyperbolic sine of the given number |
| sqrt() | Returns the square root of the given number |
| tan() | Returns the tangent of the given number |
| tanh() | Returns the hyperbolic tangent of the given number |
| trunc() | Returns an integer part of the given number |

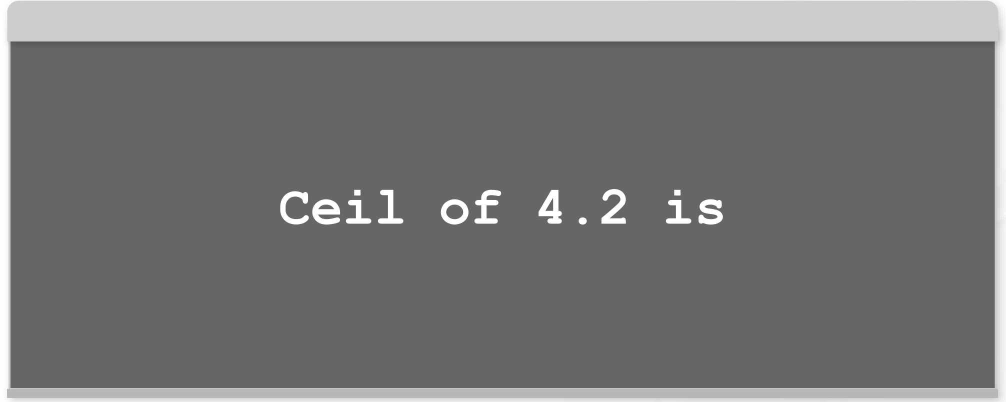
# Math Object

Example:



```
<!DOCTYPE html>
<html>
<body>
Ceil of 4.2 is: <span id="result"></span>
<script>
document.getElementById(result).innerHTML=Math.ceil
(4.2);
</script>
</body>
</html>
```

Output:



```
Ceil of 4.2 is
```

TECHNOLOGY

# String

# String

JavaScript string represents a sequence of characters.

1 By string literal

2 By string object

# String Literals

The string literal is enclosed within double quotes.

```
var stringName = “string value”;
```

**Example:**

```
<!DOCTYPE html>
<html>
<body>
<script>
var str="Example of string literal";
document.write(str);
</script>
</body>
</html>
```

**Output**

```
Example of string literal
```

# String Object

The string object is created with the help of the ‘new’ keyword.

```
var stringName = new String(“string object”);
```

**Example:**

```
<html>
<body>
  <script>
      var stringName = new String(“Hello World!”);
      Document.write(stringName) ;
  </script>
</body>
</html>
```

**Output**

```
Hello World!
```

TECHNOLOGY

# JavaScript String Method

# JavaScript String Method

| Method | Description |
|---|---|
| charAt() | It provides the char value present at the  specified index. |
| charCode() | It provides the Unicode value of a character present  at the  Specified index. |
| concat() | It provides the combination of two or more strings. |
| indexOf() | It provides the position of a char value present in the given string. |
| lastIndex() | It provides the position of a char value present in the given String by searching for a character from the last position. |
| search() | It searches a specified regular expression in a given String and returns its position if the match occurs in this. |

# JavaScript String Method

| Method | Description |
|---|---|
| match() | It searches a specified regular expression in each string and returns the regular expression if a match occurs. |
| replace() | It replaces a given string with the specified replacement. |
| substr() | It is used to fetch the part of the given string based on the specified starting position and the length. |
| substring() | It is used to fetch the part of the given string based on the specified index. |
| slice() | It is used to fetch the part of the given string. It allows you to assign a positive as well as a negative index. |
| toLowerCase() | It changes the uppercase letter in the given string to a lowercase letter. |

# JavaScript String Method

| Method | Description |
|---|---|
| toLocaleLowerCase() | It changes the uppercase letter in the given string to a lowercase letter based on the host's current locale. |
| toUpperCase() | It converts the given string into an uppercase letter. |
| toLocaleUpperCase() | It converts the given string into an uppercase letter based on the host's current locale. |
| toString() | It provides a string representing a particular object. |
| valueOf() | It provides the primitive value of the string object. |
| split() | It splits a string into the substring array, then returns the newly created array. |
| trim() | It trims the white space from the left and the right-side string. |

# JavaScript String Method

**Example:**

```
<!DOCTYPE html>
<html>
<body>
<script>
var str = “Javascript string methods”
document.write(str.charAt(8));
</script>
</body>
</html>
```

**Output:**

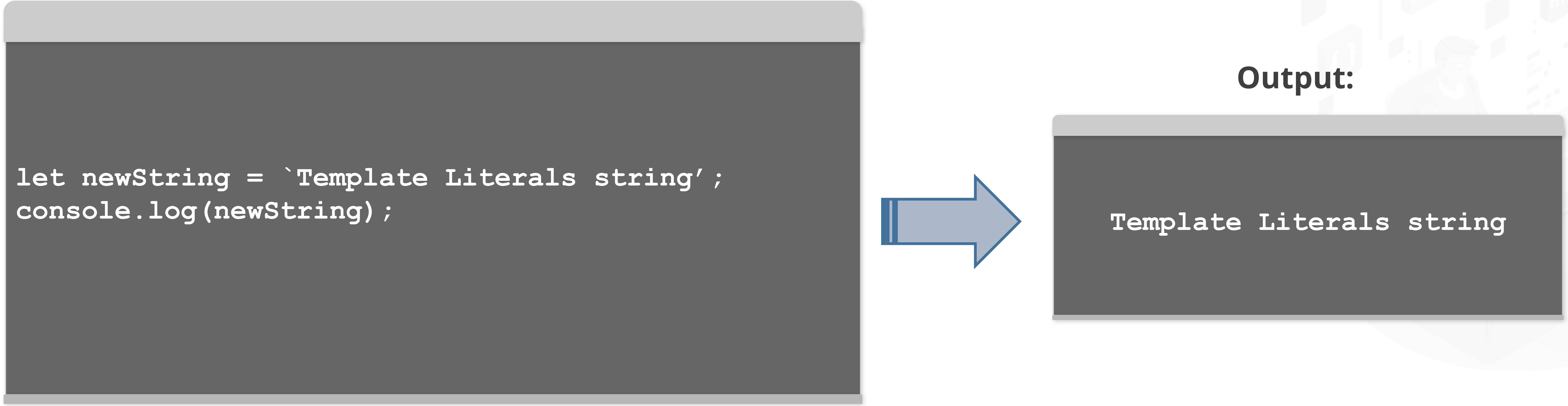
```
Check the output
```

# Template Literals

# Template Literals

It allows using strings or embedded expressions in the form of the string.

**Example:**



```
let newString = `Template Literals string’;
console.log(newString);
```

**Output:**

```
Template Literals string
```

# Key Takeaways

- In JavaScript, the object is an unordered standalone entity having property and type.
- object.keys are used in conjunction with the for-in loop to avoid an error message.
- The constructor function is used to create objects in JavaScript.
- Garbage collection is automatic memory management used by JavaScript.
- Template literals allow using strings or embedded expressions in the form of strings.

Thank You