JavaScript

Objects

Pooja Godse
&
ALK Bilahari

Procareer Academy

Outline

- Objects
- Types of Objects
- Creating and Accessing of Objects
- Number Object
- Boolean Object
- Array Object
- String Object

Recap

- Const Arrays
- Array Iteration
- Array Methods

Procareer Academy

Objects

- An object is an **unordered collection of properties**, each of which has a name and a value.
- The '**key**' is a **string** or a **symbol** and should be a legal identifier.
- The '**value**‘ can be any JavaScript value like **Number, String, Boolean, or another object**.
- JavaScript provides **built-in objects; user-defined JavaScript objects** can be created using object literals.

Procareer Academy

Objects

Syntax:

- {
- key1 : value1,
  - key2 : value2,
  - key3 : value3
- };

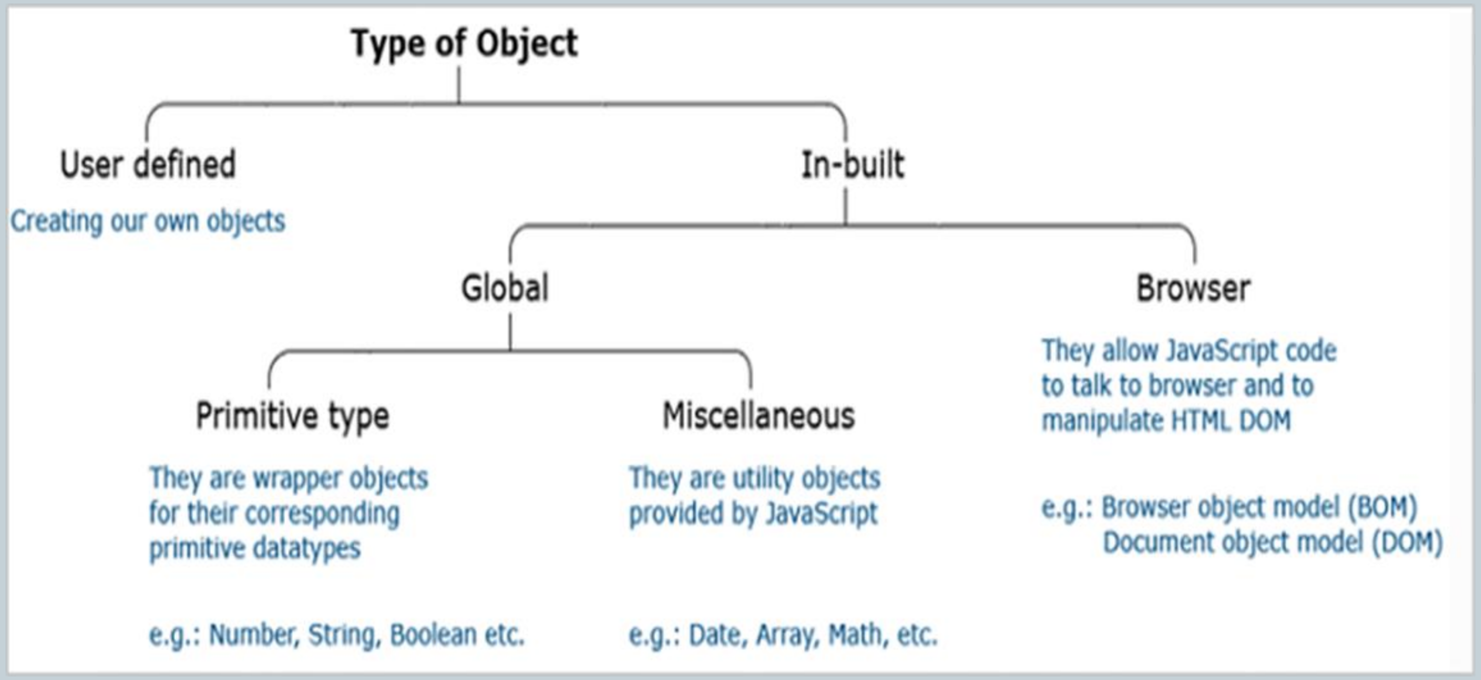
Example

Below is the modern way to create objects in a simpler way:

```
let name="Arnold";
let age=65;
let country="USA";
let obj={ name, age, country};
```

Types of Objects

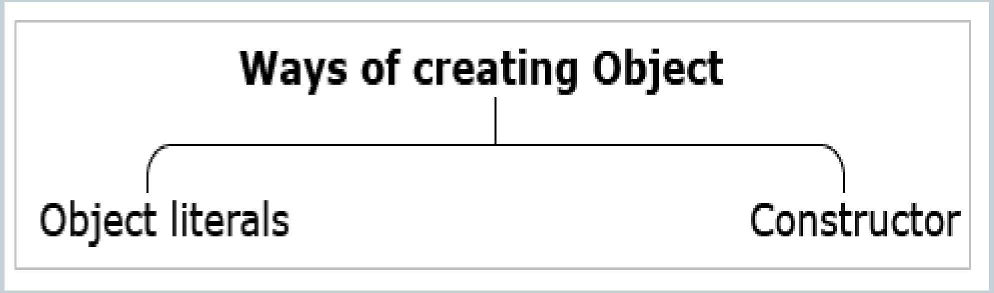
JavaScript objects are categorized as follows:


Type of Object
User defined
Creating our own objects
In-built
Global
Browser
They allow JavaScript code to talk to browser and to manipulate HTML DOM
Primitive type
They are wrapper objects for their corresponding primitive datatypes
e.g.: Number, String, Boolean etc.
Miscellaneous
They are utility objects provided by JavaScript
e.g.: Date, Array, Math, etc.
e.g.: Browser object model (BOM) Document object model (DOM)

Procareer Academy

Creation of Objects

- In JavaScript objects, the **state and behavior** is represented as a collection of properties
- Each property is a [key-value] pair where the key is a string and the value can be any JavaScript primitive type value, an object, or even a function.
- JavaScript objects can be created using two different approaches.



Object Literals

- Objects can be created using **Object literal** notation.
- Object literal notation is a comma-separated list of name-value pairs wrapped inside curly braces.
- This promotes the encapsulation of data in a tidy package.

Object Literals

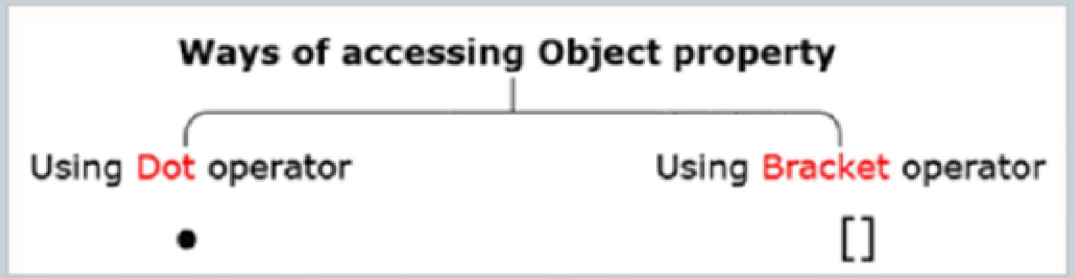
Syntax:

```
let name = "Arnold";
let age = 65;
let country = "USA";
let obj = {
  name: name,
  age: age,
  country: country
};
```

Procareer Academy

Accessing Object Properties

- The variables or methods of an object can be accessed in two different ways using:
  - **Dot operator**
  - **Bracket operator**



Accessing Object Properties

Syntax:

- For retrieving state/behavior value,
  - objectName.key;
  - *//OR*
  - objectName[key];

- For setting state/behavior value,
  - objectName.key = value;
  - *//OR*
  - objectName[key] = value;

Accessing Object Properties

To work with all the keys of an object, there is a particular form of the loop: for..in. This is a different way from the for() construct.

Syntax:

```
for (key in object) {
  // executes the body for each key among object properties
}
```

Accessing Object Properties

Example:

```
let user = {
   name: "Rexha",
   age: 24,
   isConfirmed: true
};

for (let key in user) {
     console.log(key);  // name, age, isConfirmed
    console.log(user[key]); // Rexha, 24, true
}
```

Adding new property

- Objects are dynamic; properties can usually be added and deleted.
- We can add a property to an object after object creation.

Example:

1. let person = { firstName: 'John', lastName: 'Doe' };
2. person.age=25;
3. console.log(person);

Modifying Properties

To change the value of a property, we use the **assignment operator (=)**

Example:

```
1. let person = { firstName: 'John', lastName: 'Doe' };
2. person.firstName = 'Jane';
3. console.log(person);
```

Output:

```
{ firstName: 'Jane', lastName: 'Doe' }
```

Deleting a Property

Syntax:

- **delete** objectName.propertyName;
- Example:
- **delete** person.age;
- If we attempt to re-access the age property, we'll get an undefined value.

Property Existence

- To check if a property exists in an object, we use the **in** operator:
  - propertyName **in** objectName
- The in operator returns **true** if the propertyName exists in the objectName.

Example:

1. *let employee = { firstName: 'Peter', lastName: 'Doe', employeeId: 1 };*
2. *console.log('ssn' in employee); //false*
3. *console.log('employeeId' in employee); //true*

Procareer Academy

Questions?

Procareer Academy

Thank you.

Procareer Academy