In JavaScript, objects are a fundamental data type and play a central role in the language.

**1. Key-Value Pairs:**

Objects in JavaScript are collections of key-value pairs. Each key is a string or symbol, and each value can be any data type, including other objects.

**2. Creation of Objects:**

Objects can be created using object literals, constructor functions, the ‘Object.create()’ method, or the class syntax introduced in ECMAScript 2015 (ES6).

**3. Internal Representation:**

Internally, JavaScript engines use various data structures to represent objects. Commonly, objects are implemented using a combination of hash tables and hidden classes.

**4. Dynamic Properties:**

Objects in JavaScript are dynamic, meaning properties can be added or removed at runtime. This flexibility makes them versatile for a wide range of use cases.

**5. Prototypes and Inheritance:**

Objects can inherit properties and methods from other objects through prototypes. JavaScript follows a prototype-based inheritance model, allowing objects to inherit from other objects.

**6. Accessing Properties:**

Properties of an object can be accessed using dot notation (‘obj.property’) or bracket notation (‘obj[‘property’]’). Bracket notation is useful when property names are dynamic or contain special characters.

**7. Object Methods:**

Objects in JavaScript can have methods, which are functions associated with the object. Methods can be called using the object's context, and they can access and manipulate the object's properties.

**8. Object.keys() and Object.values():**

The ‘Object.keys()’ method returns an array of a given object's own enumerable property names, while ‘Object.values()’ returns an array of its own enumerable property values. These methods are useful for iterating over an object's properties.

**9. Object Serialization:**

Objects can be serialized into JSON (JavaScript Object Notation) using ‘JSON.stringify()’. This is often used for data exchange between a client and a server.

**10. Object Destructuring:**

Object destructuring is a feature in JavaScript that allows you to extract properties from objects and bind them to variables in a concise and readable way. This is particularly useful when working with complex objects.