**Objects in JavaScript**

Object is a collection of key-value pairs, where each key is a string (or a symbol in ES6+) and each value can be of any data type, including other objects. Objects are versatile and can represent a wide range of real-world entities or data structures. They are used for organizing and storing related data.

**EXAMPLE:**

const person = {

firstName: "John",

lastName: "Doe",

age: 30

};

### Internal Representation of Objects

1. **Property Names**: Object keys are stored as strings (or symbols). These keys are hashed for efficient lookup.
2. **Property Values**: The values associated with object properties are stored in memory locations or references to other objects or data.
3. **Hidden Classes**: JavaScript engines often use a concept called hidden classes or shapes to optimize property access. When an object's properties are accessed or modified, the engine associates it with a particular hidden class. This helps improve property lookup speed.
4. **Property Descriptors**: Each property has a descriptor that defines its characteristics, such as whether it is writable, enumerable, or configurable.
5. **Prototypes**: Objects in JavaScript can be linked to other objects through a prototype chain. If a property is not found in an object, the engine looks up the prototype chain to find it.
6. **Object Layout**: JavaScript engines create an internal layout for each object type. This layout includes information about the object's properties and methods.