**TITLE: Object and Its Internal Representation in JavaScript**

* Introduction of object in JS(JavaScript):

In JS, objects are a fundamental data structure, used to represent complex data.

They are used to store data and functionality together. They are made of properties and methods.

* Internal Representation:

When we create an object in JavaScript, memory is allocated for it in a special area known as the heap. This is where all dynamically allocated memory resides.

In JavaScript, objects are implemented as hash tables or dictionaries, using a data structure called a "property bag". This bag contains a collection of key-value pairs, where:

- Keys are strings (property names)

- Values can be any type (primitive, object, function, etc.)

For Example:

let bike = {name: 'SuperSport', maker:'BMW s1000rr', engine:'999cc'};

Each key-value pair is stored in a single entry, with the key being unique within the object.

* Property Entries:

In JavaScript, objects are collections of key-value pairs, where keys are strings (or symbols) and values can be of any data type, including other objects. Objects are used to represent real-world entities, data structures, and more complex data types.

const person = {  
 name: "ROHIT",  
 age: 22,  
 email: "thisisid@gmail.com"  
};

A property entry consists of:

- Key (string): like name “Rohit”

- Value (any type): like age ‘22’ which is a number

- Enumerator (a flag indicating enumerability)

- Configurable (a flag indicating configurability)

These flags control the behaviour of the property during enumeration, deletion, and modification.

* Memory Layout:

Objects in JavaScript are stored in memory as a contiguous block, with each property entry occupying a fixed size slot. The memory layout resembles a linked list, with each entry pointing to the next.

Implications:

- Property access is fast (O (1) lookup)

- Property enumeration follows the order of insertion

- Object size grows dynamically as properties are added

* Primitive Values:

When storing primitive values (like numbers or Booleans) as object properties, JavaScript uses a technique called "boxing". The primitive value is wrapped in an object wrapper, creating a new object with a single property.

This process allows primitive values to be treated as objects, enabling methods like toString() and valueOf().

**Conclusion:**

In JavaScript, objects are represented as hash tables with key-value pairs, stored in a contiguous memory block. Understanding this internal representation reveals the reasons behind objects' behaviour, such as fast property access and enumeration order. By grasping these fundamentals, you'll become a master of JavaScript objects and unlock the secrets of this powerful language.

By:

HARSH KANWAT