Here are conceptual examples that demonstrate **JavaScript Object Literals**, their properties, methods, and practical usage:

**1. Declaring Object Literals**

* An object literal is created using {} with key-value pairs.

let person = {

firstName: "John",

lastName: "Doe",

age: 30,

isEmployed: true

};

console.log(person.firstName); // Output: "John"

console.log(person["age"]); // Output: 30

**2. Nested Objects**

* Objects can have other objects as values.

let employee = {

name: "Alice",

position: "Developer",

address: {

city: "New York",

zip: 10001

}

};

console.log(employee.address.city); // Output: "New York"

**3. Adding/Updating Properties**

* Add or update properties dynamically.

let car = {

brand: "Toyota",

model: "Corolla"

};

// Add a new property

car.year = 2023;

// Update an existing property

car.model = "Camry";

console.log(car); // Output: { brand: "Toyota", model: "Camry", year: 2023 }

**4. Deleting Properties**

* Remove properties using the delete operator.

let user = {

username: "john\_doe",

password: "12345"

};

delete user.password;

console.log(user); // Output: { username: "john\_doe" }

**5. Object Methods**

* Methods are functions defined inside an object.

let calculator = {

add: function (a, b) {

return a + b;

},

subtract(a, b) {

return a - b; // Shorthand method declaration

}

};

console.log(calculator.add(5, 3)); // Output: 8

console.log(calculator.subtract(10, 4)); // Output: 6

**6. Looping Through Properties**

* Use for...in to iterate over an object's properties.

let student = {

name: "Jane",

age: 22,

course: "Computer Science"

};

for (let key in student) {

console.log(`${key}: ${student[key]}`);

}

// Output:

// name: Jane

// age: 22

// course: Computer Science

**7. Using this Keyword**

* this refers to the object the method belongs to.

let user = {

name: "John",

greet() {

return `Hello, ${this.name}!`;

}

};

console.log(user.greet()); // Output: "Hello, John!"

**8. Object Destructuring**

* Extract properties into variables.

let product = {

id: 101,

name: "Laptop",

price: 50000

};

let { name, price } = product;

console.log(name); // Output: "Laptop"

console.log(price); // Output: 50000

**9. Object Short-Hand Syntax**

* Use shorthand when property names match variable names.

let brand = "Apple";

let model = "iPhone";

let phone = { brand, model }; // Shorthand

console.log(phone); // Output: { brand: "Apple", model: "iPhone" }

**10. Dynamic Property Keys**

* Use computed property names.

let key = "color";

let value = "red";

let item = {

[key]: value

};

console.log(item); // Output: { color: "red" }

**11. Merging Objects**

* Use Object.assign() or the spread operator.

let obj1 = { a: 1, b: 2 };

let obj2 = { b: 3, c: 4 };

let merged = Object.assign({}, obj1, obj2);

console.log(merged); // Output: { a: 1, b: 3, c: 4 }

let spreadMerged = { ...obj1, ...obj2 };

console.log(spreadMerged); // Output: { a: 1, b: 3, c: 4 }

**12. Checking for Properties**

* Use in or hasOwnProperty.

let person = { name: "Alice", age: 25 };

console.log("name" in person); // Output: true

console.log(person.hasOwnProperty("age")); // Output: true

**13. Freezing and Sealing Objects**

* **Freeze**: Prevent adding, deleting, or modifying properties.
* **Seal**: Allow modifications but prevent adding/deleting properties.

let obj = { name: "John" };

Object.freeze(obj);

obj.name = "Doe"; // Fails silently in strict mode

console.log(obj.name); // Output: "John"

Object.seal(obj);

obj.name = "Doe"; // Works

delete obj.name; // Fails silently in strict mode

**14. Using Object Methods**

**Object.keys()**

* Returns an array of property names.

let person = { name: "Alice", age: 25 };

console.log(Object.keys(person)); // Output: ["name", "age"]

**Object.values()**

* Returns an array of property values.

console.log(Object.values(person)); // Output: ["Alice", 25]

**Object.entries()**

* Returns an array of key-value pairs.

console.log(Object.entries(person)); // Output: [["name", "Alice"], ["age", 25]]

**15. Converting Objects to JSON**

**To JSON**

let user = { name: "Alice", age: 25 };

let jsonString = JSON.stringify(user);

console.log(jsonString); // Output: '{"name":"Alice","age":25}'

**From JSON**

let json = '{"name":"Alice","age":25}';

let obj = JSON.parse(json);

console.log(obj.name); // Output: "Alice"

**16. Using Objects for Maps**

let map = {};

map["key1"] = "value1";

map["key2"] = "value2";

console.log(map["key1"]); // Output: "value1"

console.log(Object.keys(map)); // Output: ["key1", "key2"]

Would you like to dive deeper into any of these concepts or see practical challenges?