Modern JavaScript: let/const, Arrow Functions, Destructuring, Classes, Promises

1. let & const

`let` and `const` are block-scoped variable declarations.

- `let` allows reassignment

- `const` creates a constant (cannot be reassigned)

Example:

```
let count = 5;
count = 6;
const pi = 3.14;
// pi = 3.15; // Error
```

2. Arrow Functions

Arrow functions provide a shorter syntax and do not bind their own `this`.

Example:

```
const greet = () => {
  return "Hello!";
};
const add = (a, b) => a + b;
```

3. Destructuring

Destructuring extracts values from arrays or properties from objects into distinct variables.

Example:

```
const person = { name: "Shekhar", age: 45 };
const { name, age } = person;
const arr = [1, 2, 3];
const [a, b] = arr;
```

4. JavaScript Classes

Classes are templates for creating objects. They use constructor methods and can include methods.

Example:

```
class Person {
  constructor(name) {
    this.name = name;
  }
  greet() {
    return "Hello, " + this.name;
  }
}
const user = new Person("Taruna");
console.log(user.greet());
```

Modern JavaScript: let/const, Arrow Functions, Destructuring, Classes, Promises

5. Promises

A Promise represents the eventual result of an asynchronous operation.

Example:

```
const fetchData = () => {
  return new Promise((resolve, reject) => {
    setTimeout(() => resolve("Data Loaded!"), 1000);
  });
};
fetchData()
  .then(data => console.log(data))
  .catch(err => console.error(err));
```