

Closure

```
function externalFunction(x) {
  return function internalFunction(y) {
    return x + y;
  };
}

const add5 = externalFunction(5);
console.log(add5(3)); // Output: 8
```

Allows internal functions to access external function variables even after the external function has executed.

Class

```
class Animal {
  constructor(name) {
    this.name = name;
  }
  talk() {
    console.log("generic sound");
  }
}

class Dog extends Animal {
  constructor(name) {
    super(name);
  }
  talk() {
    console.log("Bau!");
  }
}
```

JSON Parse

```
const jsonString =
'{"name":"Bob","age":25,"city":"New York","interests":["music","sport"]}';
const jsonObj = JSON.parse(jsonString);
console.log(jsonObj);
// Output: { name: 'Bob', age: 25, city: 'New York', interests: [ 'music', 'sport' ] }
console.log(jsonObj.name); // Output: Bob
console.log(jsonObj.interests[0]); // Output: music
console.log(typeof jsonObj); // Output: object
```

JSON.parse() to convert a JSON string into a JavaScript object.

Template literals

```
let firstName = "John";
let lastName = "Doe";
let text = `Welcome ${firstName}, ${lastName}!`;
//Output: Welcome John, Doe!
```

Template Literals allow variables in strings.

Prototype

```
function Person(name) {
  this.name = name;
}

Person.prototype.greet = function() {
  console.log("Hello, I'm " + this.name);
};

const person1 = new Person("Alice");
person1.greet(); // Output: Hello, I'm Alice
```

Fetch

```
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error(error));
```

The Fetch API provides a JavaScript interface for making HTTP requests and processing the responses.

Fetch is the modern replacement for XMLHttpRequest.

JSON Stringify

```
const myObj = {
  name: "Alice",
  age: 30,
  city: "Rome",
  interests: ["programming", "travel"]
};

const jsonString = JSON.stringify(myObj);
console.log(jsonString);
// Output: {"name": "Alice", "age": 30, "city": "Rome", "interests": ["programming", "travel"]}
console.log(typeof jsonString); // Output: string
```

JSON.stringify() to convert a JavaScript object to a JSON string.



By Techelopment
(techelopment)

Published 12th April, 2025.
Last updated 13th April, 2025.
Page 1 of 1.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!

<https://apollopadd.com>