JavaScript

Features

Dynamic Typing: Variables in JavaScript are not bound to a specific data type.

```
let example = 42;          // Number
example = "Hello";         // String
```

Event-Driven Programming: Supports event-driven, functional, and imperative programming styles.

```
document.getElementById("myButton").addEventListener("click", function
    alert("Button was clicked!");
});
```

Prototypal Inheritance: Objects can inherit properties and methods from other objects.

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

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

let person = new Person("Alice");
person.greet();  // Hello, Alice
```

First-Class Functions: Functions are treated as first-class citizens.

```
function add(a, b) {
    return a + b;
}

let operation = add;
console.log(operation(5, 3));  // 8
```

Asynchronous Programming: Supports asynchronous programming using callbacks, promises, and async/await.

```
async function fetchData() {
    let response = await fetch('https://api.example.com/data');
    let data = await response.json();
    console.log(data);
}

fetchData();
```

Wide Browser Support: JavaScript is natively supported by all modern web browsers.

Areas of Application

- **Web Development:** Creating interactive and dynamic websites.
- **Server-Side Development:** Using environments like Node.js to create server-side applications.
- **Mobile App Development:** Frameworks like React Native and Ionic allow JavaScript to be used for mobile development.
- **Game Development:** Used in developing browser-based games.
- **Desktop Applications:** Frameworks like Electron enable JavaScript to be used for desktop app development.

Java

Features

Object-Oriented: Promotes a clear modular structure and reusability.

```
public class Animal {
    private String name;

    public Animal(String name) {
        this.name = name;
    }

    public void speak() {
        System.out.println("My name is " + name);
    }
}

public class Dog extends Animal {
    public Dog(String name) {
        super(name);
    }

    @Override
    public void speak() {
        System.out.println("Woof! My name is " + getName());
    }
}

Dog dog = new Dog("Buddy");
dog.speak();  // Woof! My name is Buddy
```

Platform-Independent: Write Once, Run Anywhere (WORA) capability due to the Java Virtual Machine (JVM).

```
// Java code runs on any platform with a JVM
System.out.println("This code runs on any platform with a JVM!");
```

Robust and Secure: Strong memory management, lack of pointers, exception handling, and security features.

```
try {
    int[] numbers = {1, 2, 3};
    System.out.println(numbers[5]);  // This will throw an exception
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("Array index out of bounds!");
}
```

Multithreaded: Built-in support for multithreading.

```
public class MyThread extends Thread {
    public void run() {
        System.out.println("Thread is running");
    }
}

MyThread t1 = new MyThread();
t1.start();
```

Automatic Memory Management: Garbage collection to manage memory automatically.

Areas of Application

- **Web Applications:** Using frameworks like Spring and JavaServer Faces (JSF).
- **Enterprise Applications:** Widely used in large-scale enterprise systems.
- **Mobile Applications:** Development for Android applications.
- **Embedded Systems:** Used in devices such as mobile phones, sensors, and gateways.
- **Big Data Technologies:** Used in Hadoop ecosystem tools like Apache Hadoop, Apache Spark, etc.

Python

Features

Easy to Read and Write: Simple syntax and readability.

```
def greet(name):
    print(f"Hello, {name}!")

greet("Alice")
```

Interpreted Language: Code is executed line by line which makes debugging easier.

```
print("This will run first")
print("This will run next")
```

Dynamically Typed: Variable types are determined at runtime.

```
example = 42        # Integer
example = "Hello"   # String
```

Extensive Libraries: A rich set of libraries for various domains like web development, data science, AI, etc.

```
import numpy as np

data = np.array([1, 2, 3, 4, 5])
print(data.mean())
```

Portable: Runs on various platforms without requiring changes to the code.

```
print("This code runs on any platform with a Python interpreter!")
```

Object-Oriented: Supports object-oriented programming for better code reusability and structure.

```
class Animal:
    def __init__(self, name):
        self.name = name

    def speak(self):
        print(f"My name is {self.name}")

class Dog(Animal):
    def speak(self):
        print(f"Woof! My name is {self.name}")

dog = Dog("Buddy")
dog.speak()  # Woof! My name is Buddy
```

Areas of Application

- **Web Development:** Frameworks like Django and Flask.
- **Data Science and Analytics:** Libraries like Pandas, NumPy, and Matplotlib.
- **Machine Learning and AI:** Libraries such as TensorFlow, Keras, and PyTorch.

- **Automation and Scripting:** Used for writing scripts to automate tasks.
- **Software Development:** Used in developing desktop applications with frameworks like PyQt and Tkinter.
- **Networking:** Libraries like Twisted and Scapy for network programming.