

全栈语言核心特性对比

以下表格列出了常见系统性能、Web 全栈以及云原生场景中使用的语言，并对其关键语法特性进行了归纳。

语言	类型系统	变量声明	异步模型	内存管理	模块机制
Python	动态类型	<code>x = 1</code>	async/await	自动 GC	<code>import os</code>
C	静态类型 (无泛型)	<code>int x = 1;</code>	无原生异步	手动 malloc/free	<code>#include</code>
Rust	静态类型 + 泛型 + 所有权	<code>let x = 1;</code>	async fn + .await	所有权模型, 无 GC	<code>use crate::foo</code>
JavaScript	动态类型	<code>let x = 1;</code>	async/await	自动 GC	<code>import ... from ...</code>
TypeScript	静态类型 (基于 JS)	<code>let x: number = 1;</code>	async/await	自动 GC	<code>import ... from ...</code>
Go	静态类型 + 类型推导	<code>x := 1</code>	goroutine + channel	自动 GC	<code>import "fmt"</code>

关键特性解读

1. **类型系统**: Python 和 JavaScript 默认动态类型; Rust、Go 和 TypeScript 采用静态类型并支持类型推导, 适合构建大型项目。
2. **内存管理**: Python、JavaScript、Go 均由垃圾回收器管理内存; C 需要开发者手动管理; Rust 通过所有权系统实现无 GC 内存安全。
3. **异步并发**: JavaScript/TypeScript 与 Python 都提供 async/await; Go 用 goroutine 与 channel 支持并发; Rust 的 async 语法较为复杂但性能强; C 没有内建的异步模型。
4. **模块机制**: Python 使用 import, JS/TS 使用 ES Module, Go 以 package 组织, Rust 则通过 mod/use, C 主要依赖 #include。

示例代码

下面给出简单的斐波那契数列实现, 展示各语言的基本语法差异。

Python

```
async def fib(n: int) -> int:
    if n <= 1:
        return n
    return await fib(n - 1) + await fib(n - 2)
```

C

```
int fib(int n) {  
    if (n <= 1) return n;  
    return fib(n - 1) + fib(n - 2);  
}
```

Rust

```
async fn fib(n: u32) -> u32 {  
    if n <= 1 { return n; }  
    fib(n - 1).await + fib(n - 2).await  
}
```

JavaScript

```
async function fib(n) {  
    if (n <= 1) return n;  
    return await fib(n - 1) + await fib(n - 2);  
}
```

TypeScript

```
async function fib(n: number): Promise<number> {  
    if (n <= 1) return n;  
    return await fib(n - 1) + await fib(n - 2);  
}
```

Go

```
func fib(n int) int {  
    if n <= 1 {  
        return n  
    }  
    return fib(n-1) + fib(n-2)  
}
```

HTML/CSS (简单页面)

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
    body { font-family: Arial, sans-serif; }  
</style>
```

```
</head>  
<body>  
  <h1>Hello World</h1>  
</body>  
</html>
```

上述示例展示了这些语言在语法结构上的基本差异，便于全栈开发者快速比较与学习。