

## زبان برنامه نویسی Rust چیست؟

Rust یک زبان برنامه نویسی سیستم است که توسط شرکت Mozilla توسعه یافته است. این زبان با هدف تولید کدهای سریع، امن و قابل مقیاس پذیر طراحی شده است و از ویژگی هایی مانند مدیریت حافظه بدون نیاز به جمع آوری آشغال (garbage collection) و کنترل دقیق امنیت در زمینه حافظه بهره می برد.

## ویژگی های زبان Rust را به اختصار توضیح دهید؟

امنیت حافظه بدون نیاز به garbage collection

مدیریت موثر منابع

پشتیبانی از concurrency

استفاده از Ownership و Borrowing برای جلوگیری از مشکلات حافظه

سیستم نوع استاتیک و inference قوی

سازگاری با C و C++

توسعه برای برنامه های کاربردی سطح پایین و سیستم ها

## زبان Rust در چه حوزه هایی کاربرد بیشتری دارد؟

سیستم های embedded

توسعه نرم افزارهای وب

امنیت سایبری

بلاکچین و قراردادهای هوشمند

برنامه های کاربردی سیستم

بازی سازی

کاربردهای IoT و رباتیک

## تفاوت زبان Rust با زبان های C و C++ چیست؟

Rust در مقایسه با C و C++ حافظه را بدون نیاز به garbage collector مدیریت می کند و سیستم Ownership و Borrowing آن از بروز خطاهای حافظه جلوگیری می کند.

Rust در زمینه concurrency و ایمنی بهتر از C و C++ عمل می کند.

در Rust قوانین بسیار سخت گیرانه تری در مدیریت حافظه وجود دارد که توسعه دهندگان را از بروز خطاهای امنیتی و حافظه ای محافظت می کند.

## محدودیت های زبان Rust چیست؟

پیچیدگی در یادگیری مفاهیم جدید مانند borrowing و ownership

زمان کامپایل نسبتاً طولانی

پشتیبانی محدود از کتابخانه‌ها و ابزارها نسبت به زبان‌های دیگر

ممکن است برای برنامه‌نویسان تازه‌کار پیچیده به نظر برسد.

### در مورد ابزار Cargo توضیح دهید؟

Cargo ابزار مدیریت بسته‌ها و بیلد (ساخت) برای Rust است. این ابزار به راحتی به برنامه‌نویسان کمک می‌کند تا وابستگی‌های پروژه، کامپایل کردن کد، و توزیع کتابخانه‌ها را مدیریت کنند.

### فایل Cargo.lock چه عملکردی دارد؟

فایل Cargo.lock برای قفل کردن نسخه‌های وابستگی‌ها و کتابخانه‌ها استفاده می‌شود تا از ناسازگاری‌ها در نسخه‌های مختلف جلوگیری شود و برنامه‌ها در محیط‌های مختلف به همان شکل اجرا شوند.

### در مورد IDE مطرح RustRover توضیح دهید؟

RustRover یک IDE اختصاصی برای Rust است که توسط JetBrains ساخته شده است. این IDE ویژگی‌های کاملی مانند تکمیل خودکار کد، اشکال‌زدایی، و مدیریت پروژه‌ها را فراهم می‌کند.

### چند مورد از برنامه‌هایی که به زبان Rust نوشته شده‌اند را نام ببرید؟

( Firefox برای بخش‌هایی از آن )

Dropbox

( Servo مرورگر وب )

( Ripgrep ابزار جستجو در کدها )

( Solana یک بلاکچین )

### قابلیت‌های زبان Rust در Parallelism و Concurrency را به اختصار و با ذکر مثال بیان کنید؟

Rust از ویژگی‌های concurrency و parallelism به طور ایمن پشتیبانی می‌کند. از طریق ابزارهایی مانند `await` و `async` می‌توان به صورت همزمان پردازش‌ها را انجام داد.

```
rust
Copy code
async fn fetch_data() {
    // کد دریافت داده
}
```

### در مورد امکان برنامه‌نویسی زبان Rust برای Microcontroller ها و Robot ها توضیح دهید؟

Rust قابلیت برنامه‌نویسی برای microcontroller ها را با استفاده از پکیج‌هایی مانند `no_std` برای سیستم‌های `embedded` فراهم می‌کند. این ویژگی در ربات‌ها و دستگاه‌های IoT به کار می‌آید.

### کاربرد و مفهوم Crate در زبان Rust را بیان کنید؟

Crate یک واحد کد در زبان Rust است که می‌تواند شامل یک پروژه یا کتابخانه باشد. هر crate ممکن است از چندین فایل Rust تشکیل شده باشد.

### تفاوت مفهوم Generator و Iterator در زبان Rust را با ذکر مثال توضیح دهید؟

**Iterator** یک شی است که می‌تواند مقادیر را به صورت متوالی تولید کند.

```
rust
Copy code
let numbers = vec![1, 2, 3];
let iter = numbers.iter();
for number in iter {
    println!("{}", number);
}
```

**Generator** در Rust مشابه `iterator` است، ولی توانایی ذخیره و بازخوانی مقادیر را دارد. برای ایجاد `generator` ها نیاز به کتابخانه‌های خاص است.

### Monomorphization در زبان Rust چیست؟

**Monomorphization** فرآیندی است که در آن Rust نوع‌های عمومی (`generic`) را به نوع‌های خاص در زمان کامپایل تبدیل می‌کند تا کد بهینه‌تری تولید شود.

### در مورد مفهوم Ownership در زبان Rust توضیح دهید؟

**Ownership** به معنای مالکیت داده‌ها در Rust است. یک داده همیشه تنها یک مالک دارد که مسئول آزادسازی آن است. این ویژگی باعث جلوگیری از مشکلات حافظه مانند دسترسی به داده‌های آزاد شده می‌شود.

### قابلیت Borrowing در زبان Rust را با ذکر مثال توضیح دهید؟

**Borrowing** به شما این امکان را می‌دهد که به داده‌ها به صورت قرضی دسترسی پیدا کنید بدون اینکه مالکیت آن‌ها را بدست آورید.

```
rust
Copy code
fn borrow_example(s: &String) {
    println!("{}", s);
}
```

### انواع Inference Type در زبان Rust را با ذکر مثال توضیح دهید؟

Rust از سیستم استنتاج نوع‌ها استفاده می‌کند که به آن اجازه می‌دهد نوع‌ها را بدون نیاز به اعلام صریح از سوی برنامه‌نویس تشخیص دهد.

```
rust
Copy code
let x = 42; // Rust
              // به صورت خودکار به i32 تعیین می‌کند.
```

### مفهوم Pointer Smart در زبان Rust را توضیح دهید؟

**Smart pointers** در Rust ابزارهایی هستند که به طور خودکار حافظه را مدیریت می‌کنند، مانند `Box`, `Rc` و `RefCell`.

### ساختار Trait چیست؟ با ذکر مثال توضیح دهید؟

**Trait** ها مجموعه‌ای از متدها هستند که می‌توانند برای انواع مختلف تعریف شوند.

```
rust
Copy code
trait Speak {
    fn speak(&self);
}

struct Dog;
impl Speak for Dog {
    fn speak(&self) {
        println!("Woof!");
    }
}
```

## Closure چیست را ذکر مثال توضیح دهید؟

Closure ها توابعی هستند که می‌توانند محیط خود را به خاطر بسپارند.

```
rust
Copy code
let add = |x, y| x + y;
println!("{}", add(2, 3));
```

## توضیح دهید زبان Rust با کدهای ناامن (Code unsafe) (چطور رفتار می‌کند؟ با ذکر مثال توضیح دهید؟

Rust با استفاده از کدهای ناامن به توسعه‌دهندگان اجازه می‌دهد تا دسترسی مستقیم به حافظه داشته باشند، اما مسئولیت امنیت آن‌ها بر عهده خودشان است.

```
rust
Copy code
unsafe {
    let x: i32 = 42;
    let r: *const i32 = &x;
    println!("{}", *r);
}
```

## درمورد کاربرد زبان برنامه نویسی Rust در حوزه Contracts Smart توضیح دهید؟

Rust در قراردادهای هوشمند بلاکچین استفاده می‌شود زیرا قابلیت اطمینان و ایمنی بالایی در اجرا دارد. یکی از پلتفرم‌هایی که از Rust استفاده می‌کند، Parity Substrate است.

## درمورد کاربرد زبان برنامه نویسی Rust در حوزه Blockchain توضیح دهید؟

Rust به عنوان یکی از زبان‌های محبوب برای توسعه بلاکچین‌های مقیاس‌پذیر و ایمن شناخته می‌شود. پروژه‌هایی مانند Solana و Polkadot از Rust برای نوشتن کد بلاکچین‌های خود استفاده می‌کنند.

## درمورد کاربرد زبان برنامه نویسی Rust در حوزه Intelligence Artificial توضیح دهید؟

Rust به دلیل عملکرد بالا و کنترل دقیق بر منابع، برای توسعه ابزارها و کتابخانه‌های هوش مصنوعی و یادگیری ماشین به کار می‌رود.

## ابزارهای Test نویسی در زبان Rust را نام ببرید و تفاوت آنها را بیان کنید؟

cargo test: ابزار اصلی برای نوشتن و اجرای تست‌ها در Rust است.

[test]: ویژگی برای نوشتن توابع تست در Rust است.

## ابزارهای برنامه نویسی Web در زبان Rust را نام ببرید و تفاوت آنها را بیان کنید؟

Rocket: فریم‌ورک وب برای ایجاد API و اپلیکیشن‌های وب.

Actix: فریم‌ورک وب سریع و کارآمد برای توسعه برنامه‌های وب.

تفاوت‌ها در عملکرد، سادگی و سرعت هستند.

## مکانیزم کنترل خطا (Handling Error) در زبان Rust را با ذکر مثال توضیح دهید؟

Rust از دو نوع اصلی برای مدیریت خطا استفاده می‌کند: Result و Option.

```
rust
Copy code
fn divide(x: i32, y: i32) -> Result<i32, String> {
```

```

                                if y == 0 {
Err("Cannot divide by
    zero".to_string())
                                } else {
Ok(x / y)
                                }
}

```

**درمورد قابلیت‌های زبان Rust در Programming Functional با ذکر مثال توضیح دهید؟**  
 Rust از ویژگی‌های برنامه‌نویسی تابعی مانند first-class functions ، closure ، انواع داده‌های immutable پشتیبانی می‌کند.

```

rust
Copy code
let add = |a, b| a + b;
let result = add(5, 3);

```

**آیا برنامه نویسی Oriented Object در زبان Rust امکانپذیر است؟ با ذکر مثال توضیح دهید؟**  
 Rust از اصول برنامه‌نویسی شی‌گرا مانند ساختار داده‌ها (structs) و رفتارها (traits) پشتیبانی می‌کند، ولی برخلاف زبان‌های دیگر مانند C++ یا Java ، پشتیبانی از ارث‌بری مستقیم ندارد.

```

rust
Copy code
struct Dog;
impl Dog {
    fn bark(&self) {
println!("Woof!");
    }
}

```

**مفهوم Programming Meta (Profiling, Reflection) در زبان Rust را با ذکر یک مثال ساده بیان کنید؟**

**Profiling** در Rust به معنای اندازه‌گیری عملکرد برنامه است که می‌تواند شامل زمان پردازش، مصرف حافظه و سایر معیارهای سیستم باشد. این معمولاً از ابزارهایی مانند cargo bench برای ارزیابی عملکرد کد استفاده می‌شود.

**Reflection** در Rust به صورت مستقیم پشتیبانی نمی‌شود Rust. بیشتر بر اساس سیستم نوع استاتیک و کامپایل تایم عمل می‌کند و به همین دلیل امکان دسترسی به ویژگی‌های متا و تغییرات پویا در زمان اجرا مانند زبان‌های دیگر وجود ندارد.

**راهکار زبان Rust در مواجهه با مشکلات Safety Memory چیست؟ با ذکر یک مثال ساده توضیح دهید؟**  
 Rust از سیستم‌های Ownership و Borrowing برای جلوگیری از مشکلات حافظه استفاده می‌کند. این ویژگی‌ها تضمین می‌کنند که داده‌ها به طور ایمن مدیریت شوند و از دسترسی به حافظه آزاد شده جلوگیری می‌شود.

```

rust
Copy code
fn main() {
    let s = String::from("Hello");
    let r = &s; // قرض دادن به صورت immutable
    println!("{}", r); // استفاده از s از طریق اشاره‌گر
}

```

### درمورد امکان فراخوانی کدهای دیگر زبان‌ها در زبان Rust توضیح دهید؟

در Rust ، می‌توان کدهای زبان‌های دیگر مانند C یا C++ را با استفاده از ویژگی‌های **FFI (Foreign Function Interface)** فراخوانی کرد. این کار معمولاً با استفاده از کلمه کلیدی `extern` انجام می‌شود.

```
rust
Copy code
extern "C" {
    fn c_function();
}
```

درمورد امکان فراخوانی کدهای زبان Rust در زبان‌هایی مانند **Java** ، **C#** ، **C** و **Python** توضیح دهید؟  
برای فراخوانی کدهای Rust در زبان‌های دیگر می‌توان از **FFI** استفاده کرد. برای مثال:

**C, C++:** با استفاده از `extern "C"`.

**Java, C#:** از طریق **JNI (Java Native Interface)** یا **P/Invoke** در **C#**.

**Python:** از طریق کتابخانه‌هایی مانند **PyO3** یا **rust-cpython**.

### در قالب یک مثال **Programming Asynchronous** در زبان Rust را توضیح دهید؟

Rust از `async/await` برای نوشتن کدهای غیر همزمان پشتیبانی می‌کند.

```
rust
Copy code
use tokio;

#[tokio::main]
async fn main() {
    let result = fetch_data().await;
    println!("{}", result);
}

async fn fetch_data() -> String {
    "Data fetched".to_string()
}
```

### **Loading Lazy** چیست؟ با ذکر مثال در زبان Rust توضیح دهید؟

**Lazy** یک روش برای بارگذاری تأخیری است که به شما اجازه می‌دهد تا داده‌ها را تنها زمانی که نیاز دارید بارگذاری کنید.

```
rust
Copy code
use once_cell::sync::Lazy;

static HELLO: Lazy<String> = Lazy::new(|| "Hello, World!".to_string());

fn main() {
    println!("{}", *HELLO); // بارگذاری داده به صورت lazy
}
```

### ساختار داده **Tree Search Binary** را در زبان Rust پیاده‌سازی نمایید؟

در اینجا یک پیاده‌سازی ساده از درخت جستجوی دودویی (**Binary Search Tree**) آورده شده است:

```

rust
Copy code
struct Node {
    value: i32,
    left: Option<Box<Node>>,
    right: Option<Box<Node>>,
}

impl Node {
    fn new(value: i32) -> Self {
        Node {
            value,
            left: None,
            right: None,
        }
    }

    fn insert(&mut self, value: i32) {
        if value < self.value {
            if let Some(ref mut left) =
                self.left {
                left.insert(value);
            } else {
                self.left =
                Some(Box::new(Node::new(value)
                ));
            }
        } else {
            if let Some(ref mut right) =
                self.right {
                right.insert(value);
            } else {
                self.right =
                Some(Box::new(Node::new
                w(value)));
            }
        }
    }
}

```

### ساختر داده AVL Tree را در زبان Rust پیاده‌سازی نمایید؟

درخت‌های AVL از نوع درخت‌های جستجو هستند که خود را متعادل نگه می‌دارند. در اینجا یک پیاده‌سازی ساده آورده شده است، ولی برای پیاده‌سازی کامل باید جابجایی‌ها و چرخش‌ها را مدیریت کنید.

### ساختر داده Tree-heap Max را در زبان Rust پیاده‌سازی نمایید؟

درخت heap یک درخت باینری است که برای ذخیره‌سازی داده‌ها به‌گونه‌ای طراحی شده که بزرگترین یا کوچکترین مقدار در ریشه قرار دارد.

### یک سرویس API RESTful جهت پردازش درخواست‌های JSON بنویسید؟

برای نوشتن یک API ساده می‌توان از فریم‌ورک‌هایی مانند Rocket یا Actix استفاده کرد:

```

rust
Copy code
#[macro_use] extern crate rocket;
use rocket::serde::{json::Json, Deserialize, Serialize};

#[derive(Serialize, Deserialize)]
struct User {

```

```

        name: String,
        age: u32,
    }

    #[post("/user", data = "<user>")]
    fn create_user(user: Json<User>) -> Json<User> {
        Json(user.into_inner())
    }

    #[launch]
    fn rocket() -> _ {
        rocket::build().mount(
            "/",
            routes![create_user])
    }

```

**یک سرویس ساده جهت پردازش درخواست‌های مبتنی بر پروتکل gRPC بنویسید؟**  
 برای نوشتن یک سرویس gRPC در Rust، می‌توان از crate‌هایی مانند tonic استفاده کرد.

**یک سرویس ساده جهت پردازش درخواست‌های مبتنی بر Assembly Web بنویسید؟**  
 برای نوشتن کدهای Assembly برای وب در Rust، معمولاً از Crate‌هایی مانند wasm-bindgen یا wasmer استفاده می‌شود.

**Programming Socket در زبان Rust را به همراه یک مثال بیان کنید؟**  
 استفاده از سوکت‌ها در Rust به کمک کتابخانه‌های استاندارد امکان‌پذیر است:

```

rust
Copy code
use std::net::TcpListener;

fn main() -> std::io::Result<()> {
    let listener =
    TcpListener::bind("127
    .0.0.1:7878")?;
    for stream in listener.incoming() {
        let stream = stream?;
        // پردازش درخواست‌ها
    }
    Ok(())
}

```

**برنامه‌ای به زبان Rust بنویسید که عملیات CRUD را بر روی یک پایگاه داده انجام دهد؟**  
 برای اینکار می‌توان از کتابخانه‌هایی مانند diesel یا sqlx استفاده کرد.

**با استفاده از یک ORM در زبان Rust برنامه‌ای بنویسید که عملیات CRUD را بر روی یک پایگاه داده انجام دهد؟**

استفاده از Diesel برای مدیریت پایگاه داده‌ها به این صورت است:

```

rust
Copy code
#[macro_use] extern crate diesel;
use diesel::prelude::*;
use my_project::models::{Post, NewPost};

fn create_post(conn: &PgConnection, title: &str, body: &str) -> Post {
    let new_post = NewPost { title, body
    };
}

```



```
diesel::insert_into(posts::table)
    .values(&new_post)
    .get_result(conn)
    .expect("Error saving new post")
}
```

**کتابخانه‌های Parsing در زبان Rust را نام ببرید؟ و عملکرد یک Parser را در قالب یک مثال توضیح دهید؟**  
کتابخانه‌هایی مانند nom و pest برای پارس کردن رشته‌ها در Rust کاربرد دارند. این مثال با nom است:

```
rust
Copy code
use nom::IResult;
use nom::bytes::complete::{tag};

fn parse_hello(input: &str) -> IResult<&str, &str> {
    tag("hello")(input)
}
```

**مفهوم Expression Regular چیست؟ در زبان Rust با بیان یک مثال توضیح دهید؟**  
**Regular Expressions** برای جستجو و تطبیق الگوها در رشته‌ها استفاده می‌شود. در Rust می‌توان از crate **regex** استفاده کرد:

```
rust
Copy code
use regex::Regex;

let re = Regex::new(r"\d+").unwrap();
let text = "There are 100 apples.";
for cap in re.captures_iter(text) {

    println!("{}",
cap.get(0).unwrap().as_str());
}
```

**عملکرد کتابخانه SysInfo در زبان Rust چیست؟ با ذکر یک مثال ساده توضیح دهید؟**  
**sysinfo** یک کتابخانه است که اطلاعات سیستم مانند مصرف حافظه و پردازنده را ارائه می‌دهد:

```
rust
Copy code
use sysinfo::{System, SystemExt};

let mut system = System::new_all();
system.refresh_all();
println!("Total memory: {} KB", system.get_total_memory());
```

**عملکرد کتابخانه gui-windows-native و Windows در زبان Rust چیست؟ با ذکر یک مثال ساده توضیح دهید؟**

این کتابخانه‌ها برای ساخت رابط‌های گرافیکی در ویندوز استفاده می‌شوند. برای مثال، **winapi** برای کار با ویژگی‌های سیستم عامل ویندوز به کار می‌رود.

**برنامه‌ای برای انجام یک پردازش ساده بر روی یک Image بنویسید؟**  
می‌توان از کتابخانه‌هایی مانند **image** برای پردازش تصاویر استفاده کرد:

```
rust
Copy code
use image::GenericImageView;
```

```
let img = image::open("image.png").unwrap();
println!("Dimensions: {:?}", img.dimensions());
```

**برنامه‌ای برای انجام یک پردازش ساده بر روی یک Audio بنویسید؟**  
از کتابخانه‌هایی مانند `rodio` می‌توان برای پردازش صدا استفاده کرد.

**برنامه‌ای برای انجام یک پردازش ساده بر روی یک Video بنویسید؟**  
برای پردازش ویدئو در `Rust` می‌توان از کتابخانه‌هایی مانند `ffmpeg` استفاده کرد.

**برنامه‌ای برای دانلود یک فایل از Internet بنویسید؟**  
با استفاده از کتابخانه‌هایی مانند `request` می‌توان فایل‌هایی از اینترنت دانلود کرد.

**برنامه‌ای برای خواندن و نوشتن یک فایل CSV ساده بنویسید؟**

```
rust
Copy code
use csv::ReaderBuilder;

let mut rdr =
ReaderBuilder::new().has_headers(false).from_path("data.csv").unwrap();
    for result in rdr.records() {
        let record = result.unwrap();
        println!("{:?}", record);
    }
```

**برنامه‌ای برای خواندن و نوشتن یک فایل Excel ساده بنویسید؟**  
برای خواندن و نوشتن فایل‌های `Excel` می‌توان از کتابخانه‌هایی مانند `calamine` استفاده کرد.

**مدل MVC را در قالب یک برنامه پیاده‌سازی کنید؟**  
پیاده‌سازی `MVC` در `Rust` معمولاً به وسیله فریمورک‌هایی مانند `Rocket` یا `Actix` انجام می‌شود.

**معماری Architecture Clean را در قالب یک برنامه پیاده‌سازی کنید؟**  
معماری `Clean` به جداسازی منطق تجاری از زیرساخت‌ها و رابط‌ها اشاره دارد.

**اصول SOLID را در زبان Rust پیاده‌سازی نمایید؟**  
در `Rust`، می‌توان اصول `SOLID` را با استفاده از `Traits` و انواع داده‌ها پیاده‌سازی کرد.