درس ۵

Rust

&str و String

main.rs

```rust
fn first(s: &str) -> &str {
  &s[..1]
}
```

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی Rust

درس ۵

رشته‌ها، slice ها و lifetime ها



String در برابر &str، slice های رشته، slice های آرایه، UTF-8 و annotation های lifetime

String در برابر &str

String یک رشتهٔ مالکیت‌دار و قابل رشد روی heap است. &str یک نمای قرض‌گرفته به داخل دادهٔ ...

تابع‌ها معمولاً باید &str بگیرند: آن وقت هم literal ها و هم String ها را می‌پذیرند.



01

slice های رشته

یک slice یک محدوده از یک رشته را با قوس مربع و دو نقطه قرض می‌گیرد. به داخل دادهٔ اصلی ...

اندکس‌ها موقعیت بایت هستند، و Rust اگر در وسط یک حرف چندبایتی UTF-8 بُرش کنید panic می‌کند.

main.rs

```rust
let s = String::from("Salaam World");
let hello = &s[0..6];          // "Salaam"
let world = &s[7..];           // "World"

fn first_word(s: &str) -> &str {
    s.split(' ').next().unwrap_or("")
}
```

02

UTF-8 و char ها

رشته‌های Rust همیشه UTF-8 معتبر هستند. نمی‌توانید با s[0] یک حرف را اندکس کنید، چون حروف ...

از chars برای پیمایش حروف، و از len برای طول بایت استفاده کنید. متن دری درست کار می‌کند.

main.rs

```rust
let word = "کابل";
word.len();                    // 8 bytes
word.chars().count();          // 4 characters
for c in word.chars() { print!("[{c}]"); }
let upper = "kabul".to_uppercase();
let joined = format!("{}-{}", "Kabul", 2026);
```

03

slice های آرایه

slice ها برای آرایه‌ها و vector ها هم کار می‌کنند:
&[T] یک تابع که یک slice می‌گیرد با ...

slice ها طول خودشان را می‌دانند، پس هیچ buffer overflow وجود ندارد.

RUST PROGRAMMING · 5

main.rs

```
let a = [10, 20, 30, 40, 50];
let part: &[i32] = &a[1..4];
```

part به a[1..4] اشاره می‌کند، طول ۳

a[0]	a[1]	a[2]	a[3]	a[4]
10	20	30	40	50

04

lifetime ها

یک reference هرگز نباید بیشتر از داده‌ای که به آن اشاره می‌کند زنده بماند. معمولاً کمپایلر ...

وقتی یک تابع یک referenceرا از یکی از چند ورودی برمی‌گرداند، یک lifetime مثل 'a می‌نویسید ...

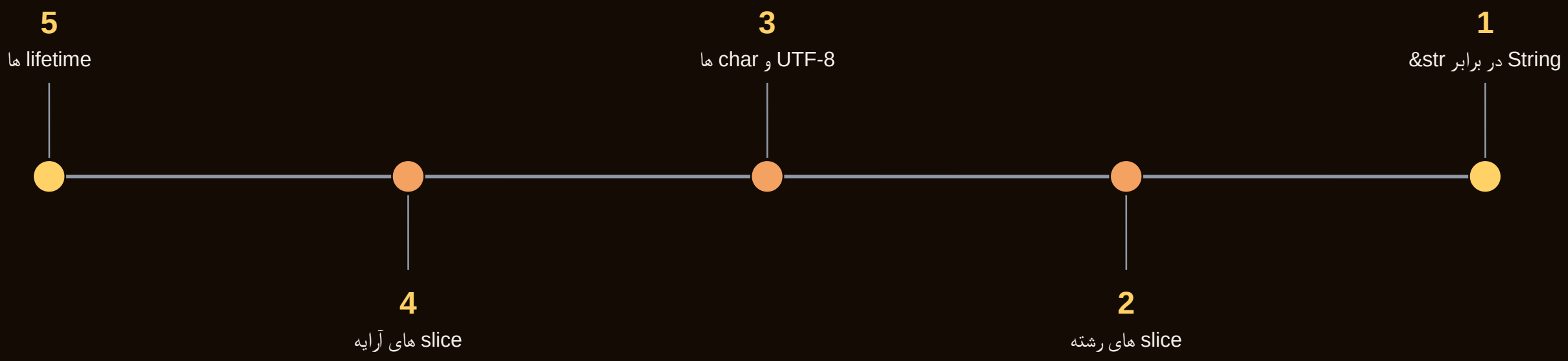
main.rs

```rust
fn longest<'a>(x: &'a str, y: &'a str) -> &'a str {
    if x.len() > y.len() { x } else { y }
}

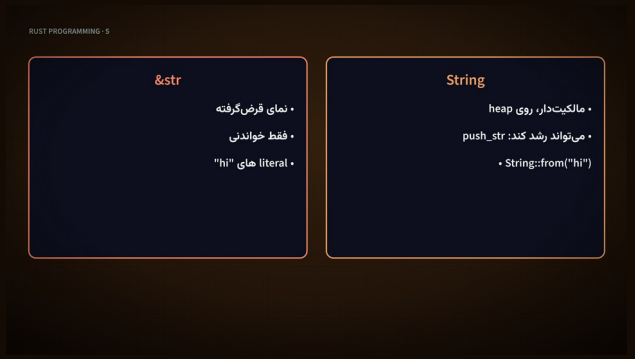
let a = String::from("Mazar-i-Sharif");
let result;
{
    let b = String::from("Herat");
    result = longest(&a, &b);
}   // error if result is used here: b does not live long enough
```

05

نقشهٔ درس


1
String در برابر &str
2
slice های رشته
3
UTF-8 و char ها
4
slice های آرایه
5
lifetime ها

منابع و مطالعهٔ بیشتر





تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity