Fundamentals | Functions

What are functions?

- A way to encapsulate program functionality
- Optionally accept data
- Optionally return data
- Utilized for code organization
  - Also makes code easier to read

Anatomy of a function

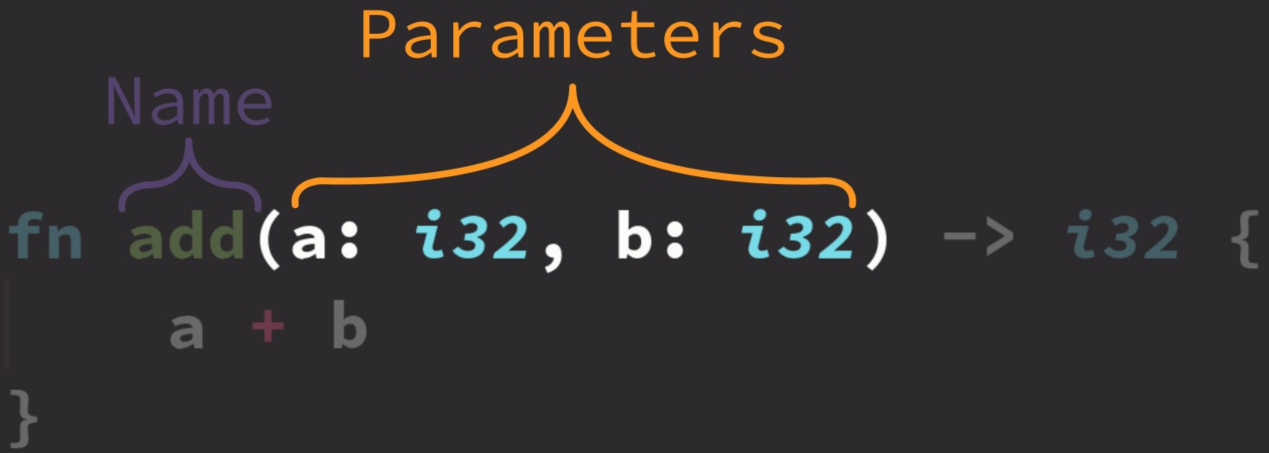
```
fn add(a: i32, b: i32) -> i32 {
    a + b
}
```

Anatomy of a function

$\overbrace{\texttt{add}}^{\text{Name}}$

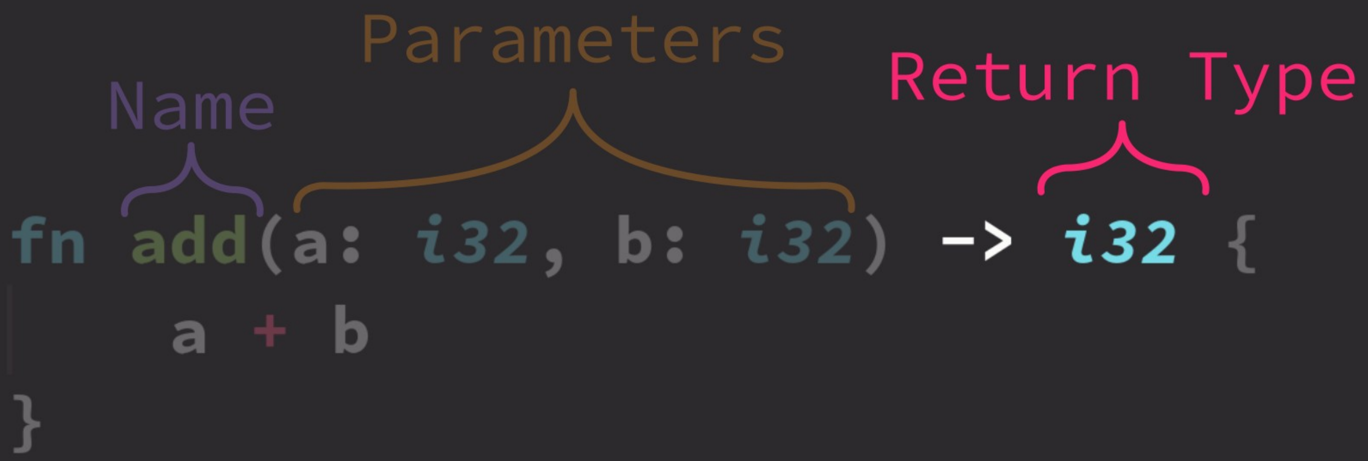
```rust
fn add(a: i32, b: i32) -> i32 {
    a + b
}
```

Anatomy of a function



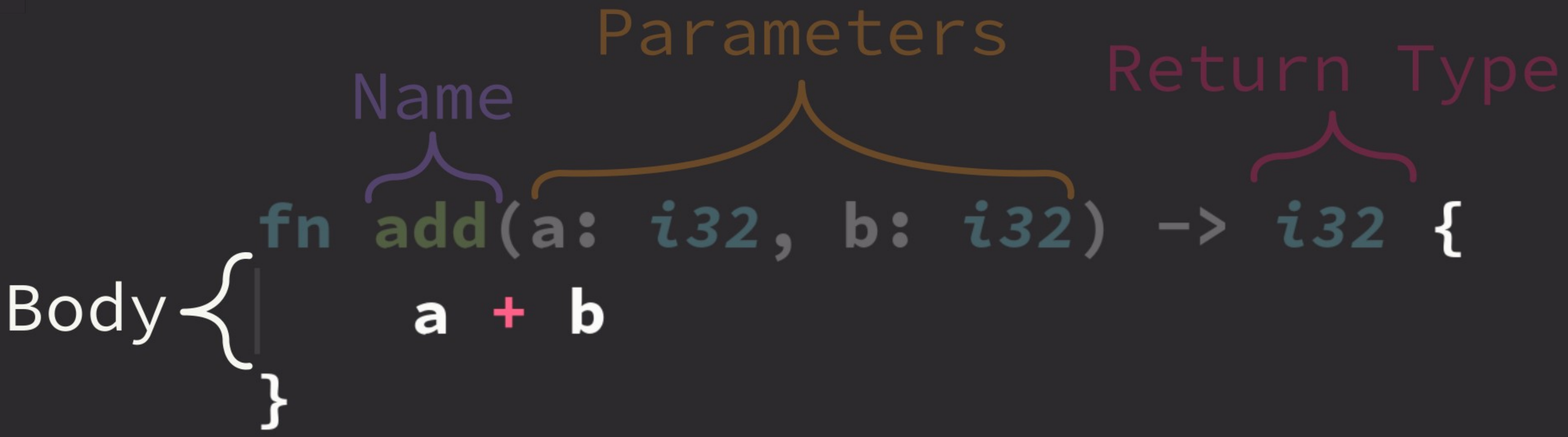
```
fn add(a: i32, b: i32) -> i32 {
    a + b
}
```

Anatomy of a function



```rust
fn add(a: i32, b: i32) -> i32 {
    a + b
}
```

Anatomy of a function



```rust
fn add(a: i32, b: i32) -> i32 {
    a + b
}
```

Using a function

```rust
fn add(a: i32, b: i32) -> i32 {
    a + b
}

let x = add(1, 1);
let y = add(3, 0);
let z = add(x, 1);
```

Recap

- Functions encapsulate functionality
- Useful to organize code
- Can be executed by “calling” the function
- Parameters determine what data a function can work with
- Optionally “returns” data
  - Data sent back from the function