

Fundamentals | Type Annotations

■ Type Annotations

- ◆ Required for function signatures
- ◆ Types are usually inferred
- ◆ Can also be specified in code
 - Explicit type annotations

■ Example – Basic

```
fn print_many(msg: &str, count: i32) { }
```

```
enum Mouse {  
    LeftClick,  
    RightClick,  
    MiddleClick,  
}
```

```
let num: i32 = 15;
```

```
let a: char = 'a';
```

```
let left_click: Mouse = Mouse::LeftClick;
```

■ Example – Generics

```
let numbers: Vec<i32> = vec![1, 2, 3];  
let letters: Vec<char> = vec!['a', 'b'];  
let clicks: Vec<Mouse> = vec![  
    Mouse::LeftClick,  
    Mouse::LeftClick,  
    Mouse::RightClick,  
];
```

■ Recap

- ◆ Type annotations are mostly optional within function bodies
 - Occasionally required if compiler cannot infer the type
- ◆ Can be specified when using *let* bindings