

**Working With Data** | *Result*

# Result

- ◆ A data type that contains one of two types of data:
  - “Successful” data
  - “Error” data
- ◆ Used in scenarios where an action needs to be taken, but has the possibility of failure
  - Copying a file
  - Connecting to a website

# ■ Definition

```
enum Result<T, E> {  
    Ok(T),  
    Err(E)  
}
```

# Example

```
fn get_sound(name: &str) -> Result<SoundData, String> {  
    if name == "alert" {  
        Ok(SoundData::new("alert")),  
    } else {  
        Err("unable to find sound data".to_owned())  
    }  
}
```

```
let sound = get_sound("alert");  
match sound {  
    Ok(_) => println!("sound data located"),  
    Err(e) => println!("error: {:?}", e),  
}
```

# Recap

- ◆ *Result* represents either success or failure
  - *Ok(variable\_name)*
    - ▶ The operation was completed
  - *Err(variable\_name)*
    - ▶ The operation failed
- ◆ Useful when working with functionality that can potentially fail
- ◆ Use *Result<T,E>* when working with results