

Working With Data | Tuples

■ Tuples

- ◆ A type of “record”
- ◆ Store data anonymously
 - No need to name fields
- ◆ Useful to return pairs of data from functions
- ◆ Can be “destructured” easily into variables

Example

```
enum Access {  
    Full,  
}
```

```
fn one_two_three() -> (i32, i32, i32) {  
    (1, 2, 3)  
}
```

```
let numbers = one_two_three();  
let (x, y, z) = one_two_three();  
println!("{:?}", "{:?}", x, numbers.0); // 1  
println!("{:?}", "{:?}", y, numbers.1); // 2  
println!("{:?}", "{:?}", z, numbers.2); // 3
```

```
let (employee, access) = ("Jake", Access::Full);
```

■ Recap

- ◆ Allow for anonymous data access
- ◆ Useful when destructuring
- ◆ Can contain any number of fields
 - Use *struct* when more than 2 or 3 fields