

Equality & Ordering | Comparing Enums

■ Comparison Operators

- ◆ Enums can be compared using equality operators
- ◆ Derivable traits enable enums to be compared
 - ***PartialEq***
 - ▶ Provides equality
 - ***PartialOrd***
 - ▶ Provides ordering: greater/less than
- ◆ ***PartialOrd*** requires ***PartialEq*** to be implemented
- ◆ Usually don't need to manually implement

■ Example – *PartialEq*

```
#[derive(PartialEq)]
enum Floor {
    ClientServices,
    Marketing,
    Ops,
}

let first = Floor::ClientServices;
let second = Floor::Marketing;
if first == second {
    //..
}
```

■ Example – *PartialOrd*

```
#[derive(PartialEq, PartialOrd)]
enum Floor {
    ClientServices,
    Marketing,
    Ops,
}

fn is_below(this: &Floor, other: &Floor) -> bool {
    this < other
}
```

■ Example – *PartialOrd* w/Variant Data

```
#[derive(PartialEq, PartialOrd)]
```

```
enum Tax {  
    Flat(f64),  
    None,  
    Percentage(f64),  
}
```

```
fn smallest_amount(tax: Tax, other: Tax) -> Tax {  
    if tax < other {  
        tax  
    } else {  
        other  
    }  
}
```

```
let no_tax = Tax::None;  
let flat_tax = Tax::Flat(5.5);
```

■ Example – *PartialOrd* w/Variant Data

```
#[derive(PartialEq, PartialOrd)]
```

```
enum Tax {  
    Flat(f64),  
    None,  
    Percentage(f64),  
}
```

```
fn smallest_amount(tax: Tax, other: Tax) -> Tax {  
    if tax < other {  
        tax  
    } else {  
        other  
    }  
}
```

```
let flat_tax = Tax::Flat(4.0);  
let percent = Tax::Percentage(1.0);
```

■ Example – *PartialOrd* w/Variant Data

```
#[derive(PartialEq, PartialOrd)]
```

```
enum Tax {  
    Flat(f64),  
    None,  
    Percentage(f64),  
}
```

```
fn smallest_amount(tax: Tax, other: Tax) -> Tax {  
    if tax < other {  
        tax  
    } else {  
        other  
    }  
}
```

```
let low = Tax::Flat(5.5);  
let high = Tax::Flat(8.0);
```

■ Recap

- ◆ Enums can be sorted and compared
 - *PartialOrd* and *PartialEq* implementation required
- ◆ These traits can be used with *derive*
- ◆ Ordering respects enum variant order in the code
 - Variant data will only be considered for ordering if both enumerations are the same variant
- ◆ Manual implementation almost never needed for enums