

Fundamentals | Struct Update Syntax

■ Struct Instantiation

- ◆ Structs may have many fields to set during creation
 - Lots of code
- ◆ **Default** can be used to set the default values
 - Sometimes one or two fields may need to have non-default values
 - ▶ Possible mutability, lots of boilerplate

■ Setup

```
struct Particle {  
    color: (u8, u8, u8),  
    alpha: u8,  
    size: (u32, u32),  
    position: (i32, i32),  
    velocity: i32,  
    direction: f32,  
}
```

```
impl Default for Particle {  
    fn default() -> Self {  
        Self {  
            color: (255, 0, 255),  
            alpha: 255,  
            size: (100, 100),  
            position: (0, 0),  
            velocity: 0,  
            direction: 0.0,  
        }  
    }  
}
```

■ Without Struct Update

```
let mut particle = Particle::default();  
particle.alpha = 127;  
let particle = particle;
```

■ Struct Update w/Default

```
let particle = Particle {  
    alpha: 127,  
    ..Particle::default()  
};
```

■ Struct Update w/Other Struct

```
let red_particle = Particle {  
  color: (255, 0, 0),  
  ..Particle::default()  
};  
  
let fast_particle = Particle {  
  velocity: 10,  
  ..red_particle  
};
```

Recap

- ◆ Struct update syntax allows structs to be easily instantiated
- ◆ Can be used with
 - *default*
 - Another struct of the same type

```
let s = Struct {  
    field: value,  
    ..Struct::default()  
};
```