



## BLOCK NONCE #10: Binding in Pattern Matching

### What?

The @ operator lets you bind a name to a value while matching it against a pattern.

### Why?

Cleaner code -> Cleaner logic.

### How?

```
if let Some(v) = value {  
    if (1..=10).contains(&v) { println!("Value {v} is in range"); }  
}
```

```
if let Some(v @ 1..=10) = value {  
    println!("Value {v} is in range");  
}
```

You can also match an enum!

```
match result {  
    Err(e @ RecoverableError { .. }) =>  
        println!("Failing due to recoverable error: {e:?}"),  
    Err(e @ FatalError { .. }) =>  
        println!("Fatal error! Shutting down: {e:?}"),  
    Ok(_) =>  
        println!("Success!"),  
}
```

A pattern inside a tuple!

```
if let (x @ 1..i32::MAX, y @ 1..i32::MAX) = pair {  
    println!("Point is in the first quadrant with x = {x} and y = {y}");  
}
```

A part of a slice!

```
fn product_non_zeroes(nums: &[i32]) -> i32 {  
    match nums {  
        [] => 1,  
        [0, ref rest @ ..] => product_non_zeroes(rest),  
        [x, ref rest @ ..] => x * product_non_zeroes(rest),  
    }  
}
```

**Call For Action!** Find one place in your code to make it cleaner using the @ operator, send a PR, and make your TL proud!

