



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?

Fewer branches -> Fewer lines -> Cleaner logic.

How?

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

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

Or match pattern of enum:

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

Or match pattern of tuple:

```
if let Some(point @ (x, y)) = pair {  
    println!("Got point {:?} with x = {}, y = {}", point, x, y);  
}
```

Or match pattern of field in struct:

```
struct User {  
    name: String,  
    age: u8,  
}  
if let u @ User { age: 18..=30, .. } = user {  
    println!("{:?} is a young user", u.name);  
}
```

Call For Action! Find one place in your code to use the @ operator, open a PR, and make your TL proud!