# Enums and Match Control Flow Operator

This lesson will get you acquainted with enums and match flow operators.

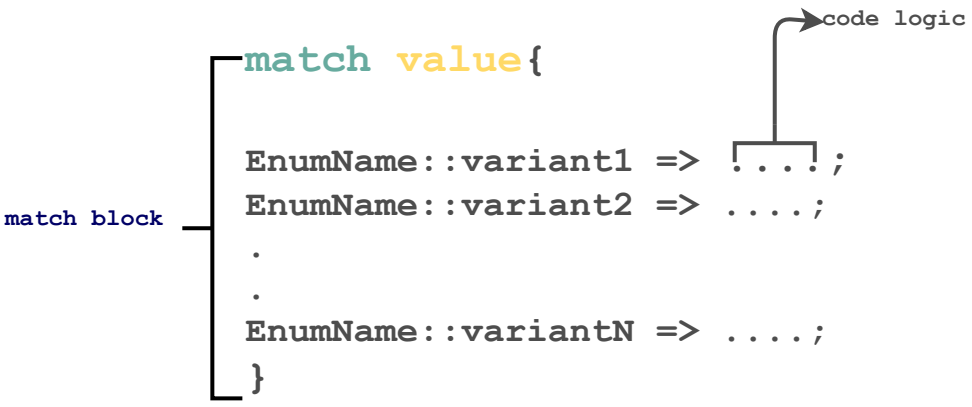
**We'll cover the following**



The `match` statement can be used to **compare values** within an `enum`. The `match` statement can be written within a `main` function or any other user-defined function.

## Syntax #

The match statement can be written within a function be it `main` or any other **user-defined function**.



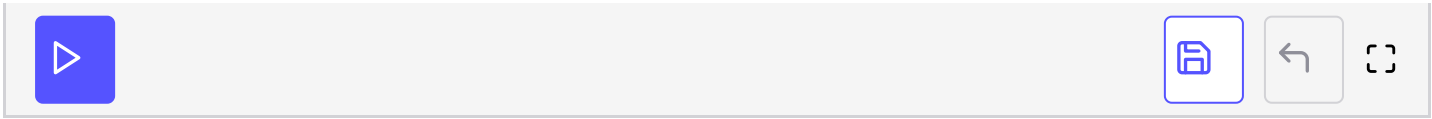

Defining a match statement within a function

## Example #

The example below makes use of a `match` statement within a `print_direction` function.

```
enum KnightMove{
   Horizontal,Vertical
}
```

```
// print function
fn print_direction(direction:KnightMove) {
   // match statement
   match direction {
      //execute if knight move is horizontal
      KnightMove::Horizontal => {
         println!("Move in horizontal direction");
      },
       //execute if knight move is vertical
      KnightMove::Vertical => {
         println!("Move in vertical direction");
      }
   }
}
fn main() {
   // invoke function `print_direction`
   let knight1 = KnightMove::Horizontal;
   let knight2 = KnightMove::Vertical;
   print_direction(knight1);
   print_direction(knight2);
}
```



## Explanation

- `main` **Function**

  The body of the `main` function is defined from **line 18 to line 24**.

  - On **line 20**, a variable `knight1` is initialized with the value `KnightMove::Horizontal`.

- On **line 21**, a variable `knight2` is initialized with the value `KnightMove::Vertical`.

- On **line 22**, `print_direction` function is invoked with `knight1` passed as a parameter.

- On **line 23**, `print_direction` function is invoked with `knight2` passed as a parameter.

- `enum`

  - On **line 1**, `enum` `KnightMoves` is defined.
  - On **line 2**, **variants** of enum `Horizontal` and `Vertical` are defined.

- `print_direction` **Function**

The function is defined from **line 5 to line 17**.

- The function has a `match` construct.
  - `match` takes a parameter `direction`
  - **On line 9**, checks if the `direction` matches with the `KnightMove::Horizontal`. If it does, it prints “Move in horizontal direction”.
  - **On line 13**, checks if the direction matches `KnightMove::Vertical`. If it does, it prints “Move in vertical direction”.

---

Let’s learn about enums and structs in the next lesson.