Introduction to Enums

This lesson will introduce you to enumeration types.

We'll cover the following

What Are Enums?

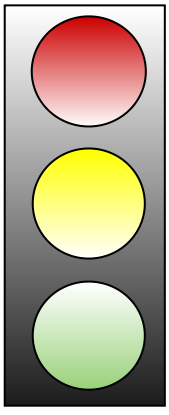
Enum is a custom data type that is composed of variants.

> Variants are values which are definite.

The key is to enumerate all possible values and select one of the values from the list.

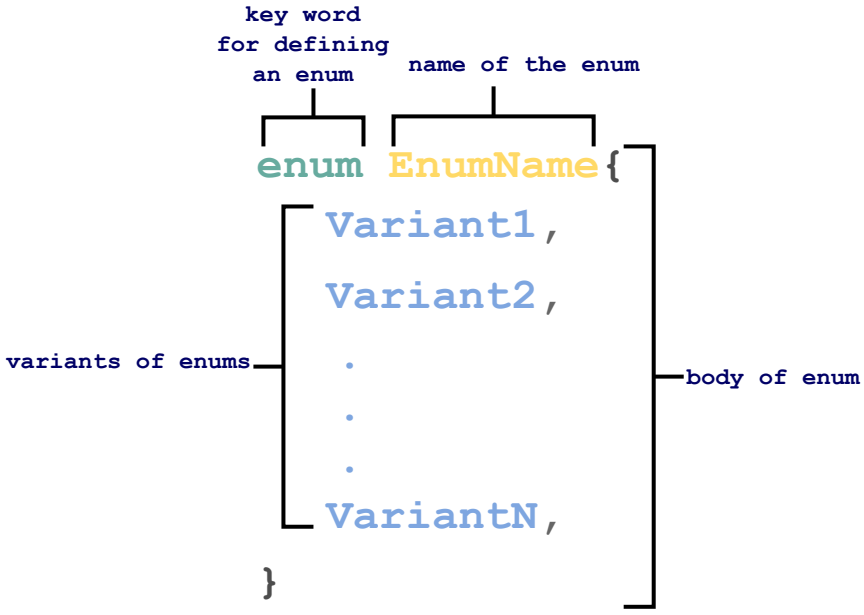
Let's consider a real life example to understand the concept of enums. The traffic signal can have only three possible states: red, yellow and green for stop, slow down, and go respectively.

Traffic Signal



Declare an Enum

Enums are declared using the `enum` keyword followed by the name of the `enum` and then the body of the enum enclosed within curly braces `{ }`. Within the curly braces, the variants of the `enum` are defined.

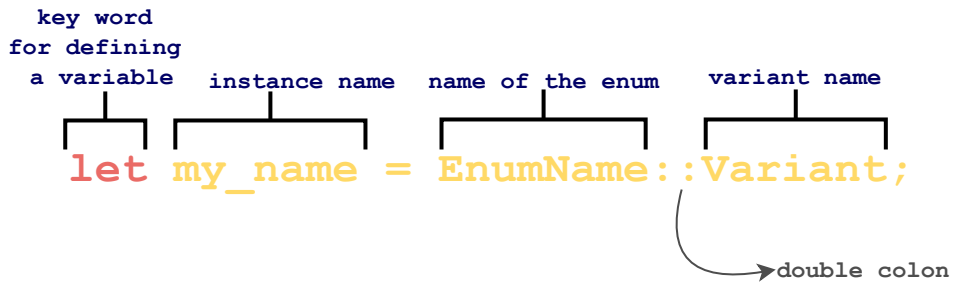


Defining an enum

> **Naming Convention**
>
> The name of the enum and it's variants are written in **CamelCase**.

Initialize an Enum

Enums are initialized using the name of the `enum` followed by a double colon(::) and then specifying the name of the variant of the enum.



Initialize an enum

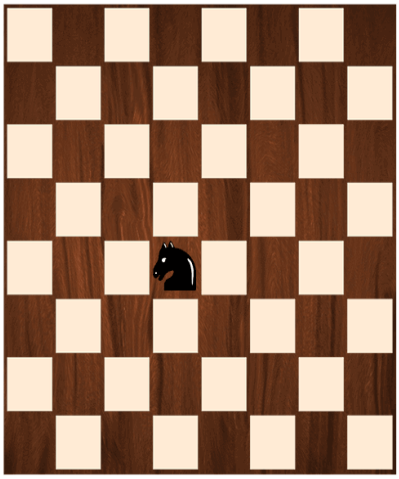
> **Note:** To print the values of an enum, write `#[derive(Debug)]` at the

beginning of the program code. Use the **debug trait{:?}** for printing the variants.

Example

The following example declares an `enum` named `KnightMoves`.

> **Note:** To keep things simple, two variants are mentioned. However, a Knight in chess can move be in four directions. It moves to a square that is two squares away horizontally and one square vertically, or two squares vertically and one square horizontally



Attribution: Wikimedia Commons

```
// make this `enum` printable with `fmt::Debug`.
#[derive(Debug)]
enum KnightMove{
   Horizontal, Vertical
}
fn main() {
   // use enum
   let horizontal_move = KnightMove::Horizontal;
   let vertical_move = KnightMove::Vertical;
   // print the enum values
   println!("Move 1: {:?}", horizontal_move);
   println!("Move 2: {:?}", vertical_move);
}
```

Explanation

- **`main` Function**

  The body of the `main` function is defined from **line 6 to line 13**.

  - On **line 8** and **line 9**, `enum` is initialized.
  - On **line 11** and **line 12**, the values of `enum` are printed.

- On **line 2** `#[derive(Debug)]` is declared which helps to print the values of the enum

enum.

- `enum`
  - On **line 3**, `enum` `KnightMove` is defined
  - On **line 4**, **variants** of enum `Horizontal` and `Vertical` are defined.

```
#[derive(Debug)]
enum KnightMove{
   Horizontal,Vertical
}
fn main() {
   let horizontal_move = KnightMove::Horizontal;
   let vertical_move = KnightMove::Vertical;
   println!("{:?}",horizontal_move);
   println!("{:?}",vertical_move);
}
```

Output:


```
#[derive(Debug)]
enum KnightMove{
   Horizontal,Vertical
}
fn main() {
   let horizontal_move = KnightMove::Horizontal;
   let vertical_move = KnightMove::Vertical;
   println!("{:?}",horizontal_move);
   println!("{:?}",vertical_move);
}
```

Output:

```
#[derive(Debug)]
enum KnightMove{
   Horizontal,Vertical
}
fn main() {
   let horizontal_move = KnightMove::Horizontal;
   let vertical_move = KnightMove::Vertical;
   println!("{:?}",horizontal_move);
   println!("{:?}",vertical_move);
}
```

Output:


```
#[derive(Debug)]
enum KnightMove{
   Horizontal,Vertical
}
fn main() {
   let horizontal_move = KnightMove::Horizontal;
   let vertical_move = KnightMove::Vertical;
   println!("{:?}",horizontal_move);
   println!("{:?}",vertical_move);
}
```

Output:

```
Horizontal
```



```
#[derive(Debug)]
enum KnightMove{
   Horizontal,Vertical
}
fn main() {
   let horizontal_move = KnightMove::Horizontal;
   let vertical_move = KnightMove::Vertical;
   println!("{:?}",horizontal_move);
   println!("{:?}",vertical_move);
}
```

Output:

```
Horizontal
Vertical
```

```
#[derive(Debug)]
enum KnightMove{
    Horizontal,Vertical
}
fn main() {
    let horizontal_move = KnightMove::Horizontal;
    let vertical_move = KnightMove::Vertical;
    println!("{:?}",horizontal_move);
    println!("{:?}",vertical_move);
}end of program code
```

Output:

```
Horizontal
Vertical
```


Quiz

Test your understanding of basics of `enums` in Rust.

Quick Quiz on Basics of Enums!

Q

Suppose you have defined an `enum`

```
enum Move{
  Left, Right, Top, Bottom
}
```

How would you call variant `Left` in the `main` function?

Retake Quiz

Now that you have learned about enums a bit, let’s move on to learn more about enums and data types.