

## Rustに触れる 2

---

ここでは条件文について取り上げます。

### if文

他の言語でも使ったことのあるだろうif文です。

```
fn main()
{
    let mut value = 100;

    if value == 100
    {
        println!("True");
    }
    else
    {
        println!("False");
    }

    // 「True」

    value += 100;

    if value == 100
    {
        println!("True");
    }
    else
    {
        println!("False");
    }

    // 「False」
}
```

C言語と違う点はまず条件文を ( ) で囲う必要がないということです。

またif文では**bool型**の値を直接扱うことができます。

```
fn main()
{
    let mut value = true;

    if value
    {
        println!("True");
    }
    else
    {
        println!("False");
    }

    // 「True」

    value = false;

    if value
    {
        println!("True");
    }
    else
    {
        println!("False");
    }

    // 「False」
}
```

関数の結果を用いて以下のように使うこともできます。

```
fn main()
{
    if check_value(1)
    {
        println!("value is bigger than 0.");
    }
    else
    {
        println!("value is smaller than 0.");
    }
    // 「value is bigger than 0.」

    if check_value(-1)
    {
        println!("value is bigger than 0.");
    }
    else
    {
        println!("value is smaller than 0.");
    }
}
```

```
    {
        println!("value is smaller than 0.");
    }

    // 「value is smaller than 0.」
}

fn check_value(value : i32)->bool
{
    value > 0
}
```

まず**`check_value`**関数では戻り値がboolであるのに対して条件文をセミコロン無しで書くことでboolを返すことを実現しています。

そしてif文の条件には関数を入れることで結果をコントロールすることができます。

次にelseや論理和、論理積です。

```
if a > 0 || b > 0
{
    // どちらかTrueでOK
}
else if a > 0 && b > 0
{
    // どちらもTrueでなければならない
}
```

## match文

Rust特有でmatchという文があります。C言語ではswitchに近い気がします。

```
fn main()
{
    let v = 1;

    match v
    {
        0=>println!("value is 0"),
        1=>println!("value is 1"),
        _=>{
            println!("value is not 0 or 1.");
        }
    }
}
```

**`match`**では比較したいものを入れてそれが取りうるすべての結果に対して考察することができます。例えば今回はint型のvという変数が入力となっているので「0だったらこう」とか「1だったらこう」とか特有の結

果も記述できますが、「それ以外」だった場合として「`_`」が条件として記述されてます。

また`match`を使うときの注意点として取りうるすべての値に対して結果を用意しないとエラーがでます。

以下は他の型の値で取り扱ってみた例です。

```
fn main()
{
    let v = 123.456;

    match v
    {
        0.0=>println!("value is 0"),
        1.0=>println!("value is 1"),
        145.3=>println!("value is 145.3"),
        _=>{
            println!("value");
        }
    }
}
```