

Rustに触れる 3

ここでは変数の型についてより詳しく解説します。

配列

```
fn main()
{
    let int_hairetu = [1, 3, 5];

    println!("{}", int_hairetu[0], int_hairetu[1], int_hairetu[2]);
}
```

これは各値を指定する場合の宣言で、すべての値を 0 にして宣言する場合には以下のようにします。

```
fn main()
{
    let buffer = [0_u8; 256];
}
```

0 とは言いましたが 0 は 0 でも色々な型の 0 があるので **u8**（正の整数 8 bit）の 0 として要素数 256 の配列を宣言してみました。

構造体

構造体とは複数の変数を格納しひとまとめにしたもので、Rust ではさらにその構造体に専用の関数を定義することが可能です。

まずは基本的な構造体について見ていきましょう。

```
fn main()
{
    let info = Person {age : 18, favorite_num : 3.14};

    println!("Age:{},Favorite Number:{}", info.age, info.favorite_num);
}

struct Person
{
    age : i32,
    favorite_num : f32
}
```