

# 파이썬 프로그래머를 위한 러스트 입문

윤인도

[freedomzero91@gmail.com](mailto:freedomzero91@gmail.com)

## CH2. 변수

## 값 출력하기

```
print("Hello, world!")
```

```
fn main() {  
    println!("Hello, world!");  
}
```

## 변수 선언

```
x = 1.0  
y = 10  
  
print(f"x = {x}, y = {y}")
```

파이썬 코드 실행 결과는 다음과 같습니다. 폴더를 하위 폴더인 "python"으로 이동한 다음 코드를 실행해야 합니다.

```
/code/temp/python $ python main.py  
x = 1.0, y = 10
```

변수명	타입	값
let x:	i32	= 10;

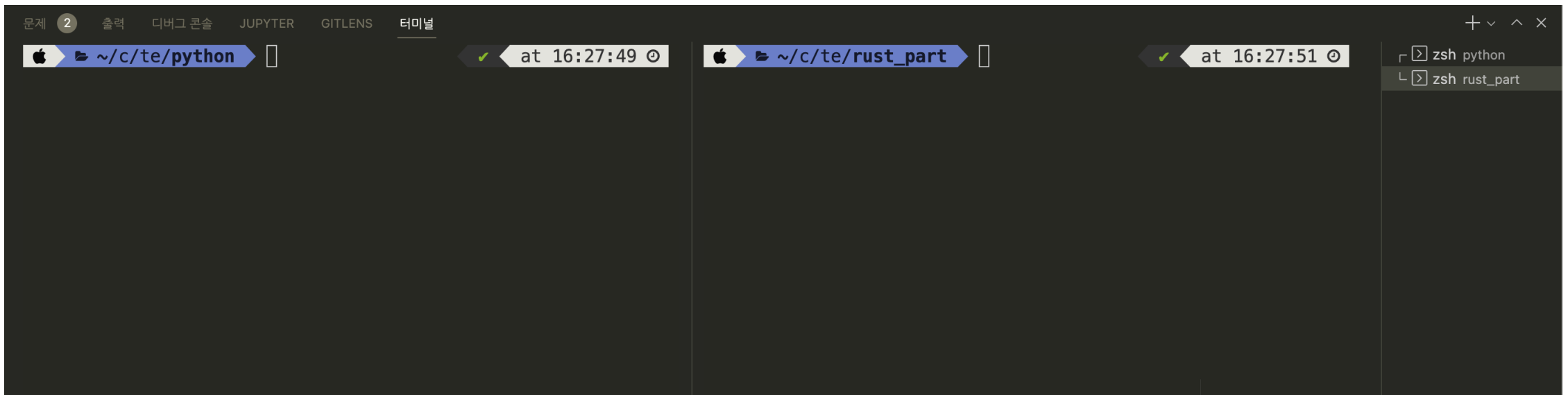
대부분의 경우에는 컴파일러가 타입을 추측해주지만, 몇몇 경우에는 직접 타입을 명시해줘야 하기도 합니다.

```
fn main() {  
    let x: f64 = 1.0;  
    let y = 10;  
  
    println!("x = {}, y = {}", x, y);  
}
```

하위 폴더인 "rust\_part" 폴더로 이동한 다음, `cargo run` 을 실행해 결과를 확인해보겠습니다.

```
/code/temp/rust_part $ cargo run  
x = 1, y = 10
```

VSCode 터미널의 분할(split) 기능을 사용하면 편리합니다.



## 작명 규칙

파이썬과 러스트의 작명 규칙은 거의 동일합니다. 변수의 경우, 둘 다 스네이크 케이스(snake case)를 사용합니다.

```
snake_case = 3
```

```
let snake_case = 3;
```



상수의 경우는 둘 다 스크리밍 스네이크 케이스(Screaming snake case)를 사용합니다.

```
SCREAMING_SNAKE_CASE = 1
```

```
const SCREAMING_SNAKE_CASE: i32 = 1;
```

## 불변성

파이썬의 변수는 언제든지 다른 타입의 값을 넣을 수 있습니다.

```
x = 1  
x = "2"  
x = 3.141592
```

```
fn main() {
    let x = 1;
    x = 2; // won't compile!
    println!("{}", x);
}
```

위 코드를 실행해보면 다음과 같은 에러가 발생합니다.

```
error[E0384]: cannot assign twice to immutable variable `x`
--> src/main.rs:3:5
 2 |     let x = 1;
   |     -
   |     |
   |     first assignment to `x`
   |     help: consider making this binding mutable: `mut x`
 3 |     x = 2; // won't compile!
   |     ^^^^^ cannot assign twice to immutable variable
```

```
let mut x = 1;
```

컴파일러의 조언에 따라 수정된 코드를 아래와 같이 작성하고 실행해봅시다.

```
fn main() {  
    let mut x = 1;  
    x = 2;  
    println!("{}", x);  
}
```

값을 바꾸고자 하는 변수에는 `mut` 키워드로 가변성을 부여해야 합니다.

## 새도잉

한번 선언한 불변 변수의 값을 변경하는 것은 불가능하지만, 변수 자체를 새로 선언하는 것은 가능합니다. 이렇게 변수 이름을 재사용해서 새로운 변수를 다시 선언하는 것을 새도잉(shadowing)이라고 합니다.

```
fn main() {  
    let x = "5";  
  
    let x = 6; // x is redeclared as 6  
  
    println!("The value of x is: {}", x); // 6  
}
```

# 타입

C언어 계열과 마찬가지로, Rust는 타입이 존재합니다. 러스트의 원시 타입(primitive type) 목록은 다음과 같습니다.

이름	타입
8비트 정수	<code>i8</code>
16비트 정수	<code>i16</code>
32비트 정수	<code>i32</code>
64비트 정수	<code>i64</code>
128비트 정수	<code>i128</code>
아키텍처	<code>isize</code>

다음 코드를 VSCode에 붙여넣으면 아래 그림과 같이 타입이 추론되는 것을 볼 수 있습니다.

```
fn main(){  
    let x = 1;  
    let y = 1.0;  
    println!("{}", x, y);  
}
```

```
fn main(){  
    let x: i32 = 1;  
    let y: f64 = 1.0;  
    println!("{}", x, y);  
}
```

변수의 타입을 다른 타입으로 바꾸는 타입 변환(Casting)도 가능합니다. 파이썬에서는 타입 이름을 바로 사용해 타입 변환을 수행합니다.

```
x = 1.2  
y = int(x)  
print(f"{x} -> {y}");
```

실행결과

```
1.2 -> 1
```



러스트에서는 아래와 같이 `as` 키워드를 사용하면 됩니다.

```
fn main() {  
    let x: f64 = 1.2;  
    let y = x as i32;  
    println!("{}", x, y);  
}
```

실행결과

```
1.2 -> 1
```

# 상수

상수(constant)란, 한 번 선언되면 값이 바뀌지 않는 변수를 의미합니다. 먼저 파이썬에서 상수를 다음과 같이 선언해 보겠습니다.

```
THRESHOLD = 10

def is_big(n: int) -> bool:
    return n > THRESHOLD

if __name__ == '__main__':
    print(THRESHOLD)
    print(is_big(THRESHOLD))

    THRESHOLD = 5
```

```
const THRESHOLD: i32 = 10;

fn is_big(n: i32) -> bool {
    n > THRESHOLD
}

fn main() {
    println!("{}", THRESHOLD);
    println!("{}", is_big(5));
}
```

실행결과

```
10
false
```

값이 불변이기 때문에 `THRESHOLD = 5;` 와 같이 새로운 값을 할당하게 되면 오류가 발생합니다.

```
...  
    THRESHOLD = 5;  
...
```

실행결과

```
--> src/main.rs:11:15  
11 |     THRESHOLD = 5;  
   |     ^  
   | cannot assign to this expression
```

컴파일러가 친절하게 상수 `THRESHOLD`에는 새로운 값을 할당할 수 없다고 알려주게 됩니다. 실행하기 전 편집기 안에서도 빨간 줄로 해당 코드에 문제가 있음을 알려주기 때문에 문제를 빠르게 찾고 해결할 수 있습니다.

```
4      n > THRESHOLD
5  }
6
7  ▶ Run ()
8  fn m
9
10     invalid left-hand side of assignment rustc(E0070)
11     main.rs(11, 5): cannot assign to this expression
12     문제 보기   빠른 수정을 사용할 수 없음
13     THRESHOLD = 5;
14 }
```



















