

03. const static

const1.rs

```
1 fn main()
2 {
3     let size0 = 10;
4
5     const SIZE1 : u32 = 10;
6
7     // const SIZE2 = 10; // error
8
9     // const size3 : u32 = 10; // 경고
10
11     // size0 = 20; // error
12     // SIZE1 = 20; // error
13 }
14
```

const

핵심 정리

- 컴파일 시간 상수
 - ⇒ let 대신 **const** 사용
 - ⇒ 반드시 타입을 지정해야 한다.
 - ⇒ 상수명은 대문자(권장)
- immutable vs const
 - ⇒ 공통점 : 초기화된 이후로 값을 변경할 수 없다.
 - ⇒ 차이점

const	컴파일 시간에 값을 알 수 있어야 한다.
immutable	실행시간에 결정된 값으로도 초기화 가능

const2.rs

```
1 fn main()
2 {
3     let mut user_input = 10;
4
5     let n1 : i32 = 10;
6     let n2 : i32 = user_input;
7
8     const C1 : i32 = 10;
9     // const C2 : i32 = user_input; //error
10 }
```

const

```
rustc const2.rs -A dead_code -A unused_variables -
A unused_assignments -o out.exe
warning: variable does not need to be mutable
--> const2.rs:3:6
3 |     let mut user_input = 10;
  |     ^^^^^^^^^^^^^^^^^
  |
  | help: remove this `mut`
  |
= note: `#[warn(unused_mut)]` on by default
warning: 1 warning emitted
[ rustc compiling success !! run out.exe ]

> 
```



```
static2.rs  static
1 //let global = 10;
2
3 static GLOBAL : i32 = 10;
4
5 fn main()
6 {
7     let local = 10;
8 }

rustc static2.rs -A dead_code -A unused_variable
-A unused_assignments -o out.exe
error: expected item, found keyword `let`
--> static2.rs:1:1
1 | let global = 10;
  | ^^^ consider using `const` or `static` instead of `let` for global variables

error: aborting due to previous error

[ Error !!, rustc fail to compile. ]

> rc static2.rs
```