



print, Dump

```
import UIKit

struct BasicInformation {
    let name: String
    var age: Int
}

var hoonInfo: BasicInformation = BasicInformation(name: "hoon", age: 99)

class Person {
    var height: Float = 0.0
    var weight: Float = 0.0
}

let hoon: Person = Person()
hoon.height = 175.2
hoon.weight = 73.3

print(hoonInfo)
dump(hoonInfo)

print(hoon)
dump(hoon)

BasicInformation(name: "hoon", age: 99)
↳ _lldb_expr_9.BasicInformation
- name: "hoon"
- age: 99
_llldb_expr_9.Person
↳ _lldb_expr_9.Person #
- height: 175.2
- weight: 73.3
```

스위프트 표준 라이브러리에는 `print()` 함수 외에도 `dump()`이라는 함수가 있습니다. `print()` 함수는 디버깅 콘솔에 간단한 정보를 출력해주는 반면, `dump()` 함수는 조금 더 자세한 정보를 출력해줍니다. `print()` 함수는 출력하는 인스턴스의 `description` 프로퍼티에 해당하는 내용을 출력해주고, `dump()` 함수는 출력하는 인스턴스의 내부 콘텐츠까지 출력해줍니다. 필요에 따라 `print()` 함수 대신에 `dump()` 함수를 적절히 사용하는 것도 좋습니다. `print()`과 `dump()` 함수는 'B'로 B. 스위프트의 주요 함수에서 더 살펴볼 수 있습니다.

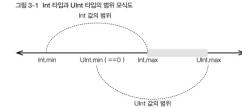
결과
dump는 더 자세하게 출력된다.

Int, UInt

- 정수 `type`으로 `Int`는 +,-부호를 포함하는 정수를 뜻하고, 이 중 -부호를 포함하지 않는 0을 포함한 양의 정수는 `UInt`로 표현됨.
- 각각 8, 16, 32, 64 비트의 형태가 있음.
- 시스템 아키텍처에 따라 `Int`와 `UInt`의 `type`이 달라짐.
 - L **32bit** : `Int32 → Int`, `UInt32 → UInt`
 - 64bit** : `Int64 → Int`, `UInt64 → UInt`

Int와 UInt은 어떻게 사용하기

클래스에 따른 Int의 최대값은 64비트. 원래라면 Int164의 최대값 이상 Int의 최대값을 사용하게 되는 경우 아래에는 UInt보다 Int 타입을 사용해야 합니다. 예를 들어 64비트 원래라면 Int64가 최대값 UInt64의 최대값보다 큼. 하지만 그걸 이용하는 경우 아래와 같은 예제가 있어야 되니 잘 하는 Int 타입을 사용하는 게 좋습니다. 예전에는 Int 타입을 사용하는 경우 아래와 같은 예제를 사용했습니다. 스위프트에서는 Int 타입을 사용하는 경우 아래와 같은 예제를 사용합니다. 원래 Int 타입을 원래 대체적으로 인터파스터에서 그 예제와 같은 예제를 유도로 사용하는 경우는 대체로 원래 대체로 인터파스터에서 그 예제와 같은 예제를 유도로 사용할 수 있습니다.



[그림 3-1] Int 타입과 UInt 타입의 범위 차이
↑ Int 범위
↑ UInt 범위

[그림 3-1]은 Int와 UInt 모두의 범위 차이입니다. 그림에서 크게 표시된 범위는 Int로 표현할 수 있는 모든 범위를 초과하고 UInt의 최대값보다는 작은 값은 범위를 넘습니다. 이 범위에 해당하는 값은 사용하고자 할 UInt를 사용하고 그 외의 경우에는 Int를 사용하기를 권장합니다.

```
1 import UIKit
2
3 var integer: Int = -100
4 let unsignedInteger: UInt = 50 // UInt 타입에는 음수값을 할당할 수 없음.
5 print("integer 값: \(integer), unsignedInteger 값: \(unsignedInteger)")
6 print("Int 최대값: \(Int.max), Int 최소값: \(Int.min)")
7 print("UInt 최대값: \(UInt.max), UInt 최소값: \(UInt.min)")
8
9 let largeInteger: Int64 = Int64.max
10 let smallUnsignedInteger: UInt8 = UInt8.max
11 print("Int64 최대값: \(largeInteger), UInt8 최대값: \(smallUnsignedInteger)")
12
13 let tooLarge: Int = Int.max + 1 // Int 타입 범위를 초과하므로 오류를 낸다.
14 let cannotBeNegative: UInt = -5 // UInt는 음수가 될 수 없으므로 오류를 낸다.
15
16 integer = unsignedInteger // 오류! 스위프트에서 Int와 UInt는 다른 타입임. ✎ Cannot assign value of type 'UInt' to ...
17 integer = Int(unsignedInteger) // Int 타입의 것으로 할당해주어야 함.
✎
```

19

```
integer 값: -100, unsignedInteger 값: 50
Int 최대값: 9223372036854775807, Int 최소값: -9223372036854775808
UInt 최대값: 18446744673799551615, UInt 최소값: 0
Int64 최대값: 9223372036854775807, UInt8 최소값: 255

▶
error: MyPlayground.playground:16:11: error: cannot assign value of type 'UInt' to type 'Int'
integer = unsignedInteger // 오류! 스위프트에서 Int와 UInt는 다른 타입임.
```