print, Dump

```swift
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)
```

스위프트 표준 라이브러리에는 print() 함수 외에도 dump()라는 함수가 있습니다. print() 함수는 디버깅 콘솔에 간략한 정보를 출력해주는 반면, dump() 함수는 조금 더 자세한 정보를 출력해줍니다. print() 함수는 출력하려는 인스턴스의 description 프로퍼티에 해당하는 내용을 출력해주고, dump() 함수는 출력하려는 인스턴스의 자세한 내부 콘텐츠까지 출력해줍니다. 필요에 따라 print() 함수 대신에 dump() 함수를 적절히 사용하는 것도 좋습니다. print()와 dump() 함수는 '부록 B. 스위프트의 주요 함수'에서 더 살펴볼 수 있습니다.

```
BasicInformation(name: "hoon", age: 99)
▿ __lldb_expr_9.BasicInformation
  - name: "hoon"
  - age: 99
__lldb_expr_9.Person
▿ __lldb_expr_9.Person #0
  - height: 175.2
  - weight: 73.3
```

← 결과

dump는 더 자세하게 출력된다.

Int, UInt

- 정수 type으로 Int는 +, - 부호를 포함하는 정수를 뜻하고, 이 중 - 부호를 포함하지 않는 0을 포함한 양의 정수는 UInt로 표현함.
- 각각 8, 16, 32, 64 비트의 형태가 있음.
- 시스템 아키텍처에 따라 Int와 UInt의 type이 달라짐.
  - 32 bit : Int32 → Int , UInt32 → UInt
  - 64 bit : Int64 → Int, UInt64 → UInt

Int와 UInt 중 선택해 사용하기

플랫폼에 따른 Int의 최댓값(64비트 환경이라면 Int64의 최댓값) 이상 UInt의 최댓값 미만을 사용하게 되는 경우 이외에는 UInt보다 Int 타입을 사용하기 바랍니다. 예를 들어 64비트 환경이라면 Int64의 최댓값과 UInt64의 최댓값 사이의 값을 사용하는 경우 이외에 정수 타입의 값을 변수에 담지자 할 때는 Int 타입을 사용하는 것이 좋습니다. 또, 양수만 사용한다고 해서 굳이 UInt를 고집할 필요는 없습니다. 스위프트는 데이터 타입에 엄격히 엄격합니다. 같은 정수라 하더라도 Int와 UInt를 완전히 다른 타입으로 인식합니다. 따라서 코드 안에 Int와 UInt 두 타입을 모두 사용하면 (부호의 유무는 더욱지만 정수 타입의 변수끼리도 값을 교환할 때 많은 제한을 소모할 수도 있습니다.

그림 3-1 Int 타입과 UInt 타입의 범위 모식도

그림 3-1은 Int와 UInt 값의 범위를 나타낸 모식도입니다. 그림에서 굵게 표시된 영역은 Int로 표현할 수 있는 값의 범위를 초과하고, UInt의 최댓값보다는 작은 값의 범위입니다. 이 범위에 해당하는 값을 사용하고자 할 때 UInt를 사용하고, 그 외의 경우에는 Int를 사용하기를 권합니다.

```swift
import UIKit

var integer: Int = -100
let unsignedInteger: UInt = 50 // UInt 타입에는 음수값을 할당할 수 없음.
print("integer 값: \(integer), unsignedInteger 값: \(unsignedInteger)")
print("Int 최대값: \(Int.max), Int 최솟값: \(Int.min)")
print("UInt 최대값: \(UInt.max), UInt 최솟값: \(UInt.min)")

let largeInteger: Int64 = Int64.max
let smallUnsignedInteger: UInt8 = UInt8.max
print("Int64 최댓값: \(largeInteger), UInt8 최댓값: \(smallUnsignedInteger)")

let tooLarge: Int = Int.max + 1 // Int의 표현 범위를 초과하므로 오류를 냄.
let cannotBeNegative: UInt = -5 // UInt는 음수가 될 수 없으므로 오류를 냄.

integer = unsignedInteger // 오류! 스위프트에서 Int와 UInt는 다른 타입임.  Cannot assign value of type 'UInt' to t...
integer = Int(unsignedInteger) // Int 타입의 값으로 할당해주어야 함.
```

```
integer 값: -100, unsignedInteger 값: 50
Int 최대값: 9223372036854775807, Int 최솟값: -9223372036854775808
UInt 최대값: 18446744073709551615, UInt 최솟값: 0
Int64 최댓값: 9223372036854775807, UInt8 최댓값: 255
```

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