

SWIFT Type Casting

Bill Kim(김정훈) | ibillkim@gmail.com

목차

What Is A Type?

Type Casting

Checking Type

Type Casting for Any and AnyObject

References

What Is A Type?

Swift에서는 타입을 통하여 변수 데이터 및 객체에 대한 형태를 표현할 수 있도록 해준다.

- 타입 예시

Int : 정수형 숫자 타입

Double : 실수형 숫자 타입

String : 문자형 타입

UIView : 애플에서 제공하는 기본 UI를 표현하기 위한 View 객체

단 Swift는 다른 언어와 달리 해당 변수 및 객체에 대한 형태가 유추가 가능하면 타입에 대해서 생략이 가능하다.

var age = 20 // Int 타입에 대해서 유추 가능하므로 타입 생략

Type Casting

Type Casting이란 원래 가지고 있던 타입에서 다른 형태의 타입으로 변환하는 것을 뜻한다.

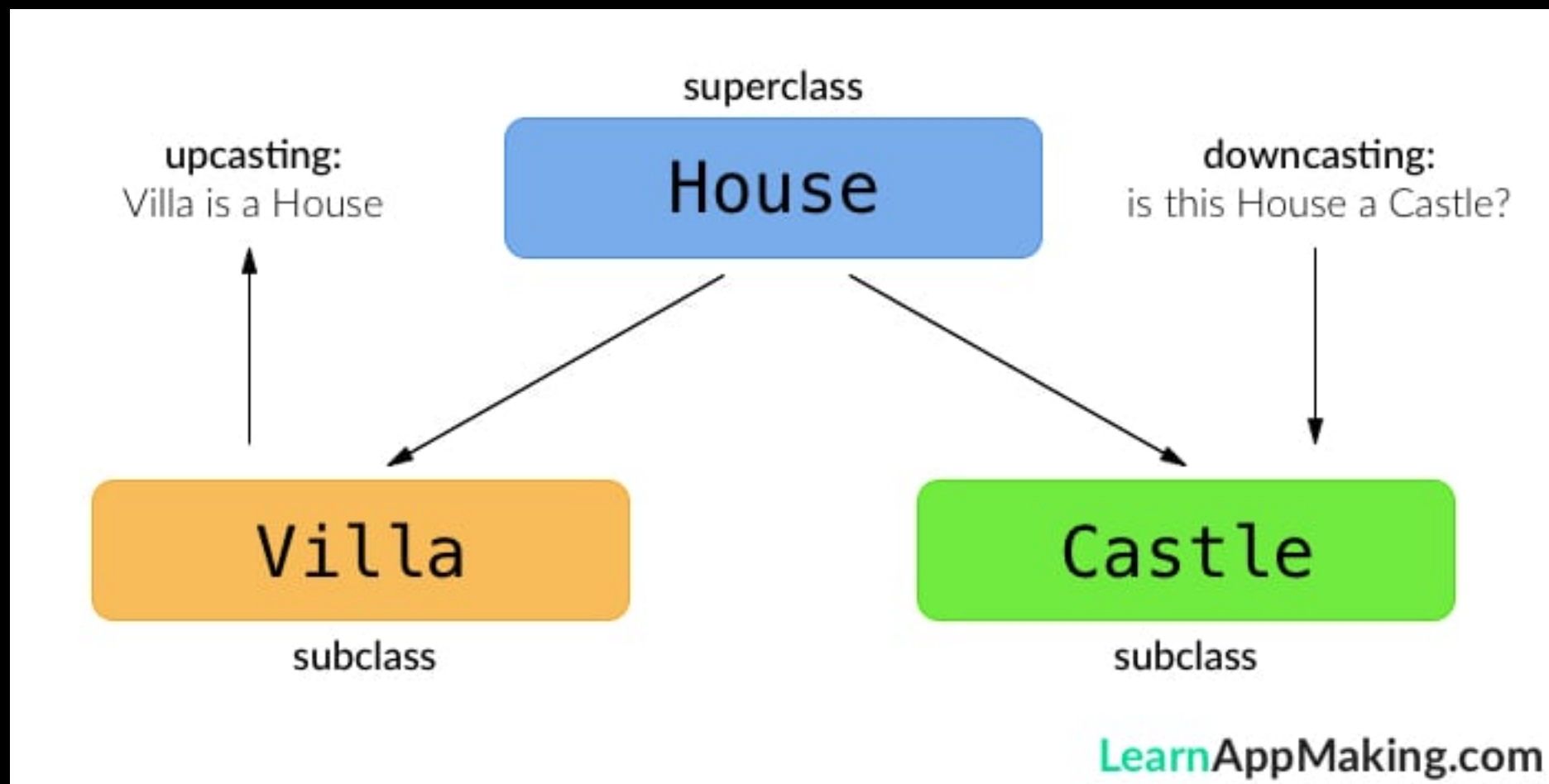
- 변수 타입 캐스팅 예시

```
var value1:Int = 20 // Int 타입에 대해서 유추 가능하므로 타입 생략  
var value2:Float = Float(value1); // Swift는 기본 자료형에 대해서 별도의 타입 캐스팅 함수를 제공한다.
```

```
print(value2) // 20.0
```

Type Casting

객체에 대한 타입 캐스팅은 객체 상속과 관련이 있으며 아래의 그림과 같은 형태로 타입 캐스팅을 할 수 있습니다.



Type Casting

Upcasting : 자식(Subclass)에서 부모(Superclass)로 캐스팅
Downcasting : 부모(Superclass)에서 자식(Subclass)으로 캐스팅

```
class House {  
    var windows:Int = 0  
  
    init(windows:Int) {  
        self.windows = windows  
    }  
}  
  
class Villa: House {  
    var hasGarage:Bool = false  
  
    init(windows:Int, hasGarage:Bool) {  
        self.hasGarage = hasGarage  
        super.init(windows: windows)  
    }  
}  
  
class Castle: House {  
    var towers:Int = 0  
  
    init(windows:Int, towers:Int) {  
        self.towers = towers  
        super.init(windows: windows)  
    }  
}
```

Type Casting

```
// Upcast(Castle -> House), as House 부분은 생략 가능
let house:House = Castle(windows: 200, towers: 4) as House

// Output: error: value of type 'House' has no member 'towers'
print(house.towers)

// Output: 200
print(house.windows)

// Downcast(House -> Castle)
let castle:Castle = house as! Castle

// Output: 4
print(castle.towers)
```

Type Casting

특정 클래스 타입의 인스턴스가 어떤 타입의 인스턴스인지 확인하는 용도로 아래와 같이 확인할 수 있습니다.

as! : 특정 타입이라는 것이 확실할 경우에 사용
as? : 특정 타입이 맞는지 확신할 수 없을 때 사용

```
// as!를 사용하여 House 객체로의 Upcast를 무조건 성공할거라고 확신함
let house:House = Castle(windows: 100, towers: 10) as! House

// as?를 사용하여 Downcast 실행, 실패할 경우 else로 진입
if let castle1 = house as? Castle {

}
else
{
    print("Downcast 실패")
}
```


Checking Type

is 연산자를 이용해 특정 인스턴스의 타입을 확인할 수 있습니다.

```
let castle:Castle = Castle(windows: 100, towers: 10)
let villa:Villa = Villa(windows: 10, hasGarage: true)
```

```
// castle은 Castle 객체입니다.
```

```
if castle is Castle {
    print("castle은 Castle 객체입니다.")
} else {
    print("castle은 Castle 객체가 아닙니다.")
}
```

```
// villa는 Castle 객체가 아닙니다.
```

```
if villa is Castle {
    print("villa는 Castle 객체입니다.")
} else {
    print("villa는 Castle 객체가 아닙니다.")
}
```

Type Casting for Any and AnyObject

Swift에서는 아래의 특별한 두가지 타입을 제공합니다.

Any : 함수 타입을 포함해 모든 타입을 나타내는 타입

AnyObject : 모든 클래스 타입의 인스턴스를 나타내는 타입

```
var things = [Any]()

things.append(0.0) // 0.0 is a double value.
things.append(42) // 42 is integer value.
things.append(3.14159) // 3.14159 is a positive double value.
things.append("hello") // "hello" is a string value.
things.append(Castle(windows: 200, towers: 4)) // test.Castle object is castle type instance.

for thing in things {
    switch thing {
    case 0.0 as Double:
        print("0.0 is a double value.")
    case let someInt as Int:
        print("\(someInt) is integer value.")
    case let someDouble as Double where someDouble > 0:
        print("\(someDouble) is a positive double value.")
    case let someString as String:
        print("\(someString) is a string value.")
    case let castle as Castle:
        print("\(castle) object is castle type instance.")
    default:
        print("something else")
    }
}
```

Type Casting for Any and AnyObject

```
var thingsObject = [AnyObject]()
```

```
thingsObject.append("error") // 객체 타입이 아니라서 컴파일 에러 발생  
thingsObject.append(Villa(windows: 10, hasGarage: true))
```

References

[1] 타입캐스팅 (Type Casting) : <https://jusung.gitbook.io/the-swift-language-guide/language-guide/18-type-casting>

[2] [Swift]Type Casting 정리 : <http://minsone.github.io/mac/ios/swift-type-casting-summary>

[3] Swift) Type Casting : <https://zeddios.tistory.com/265>

[4] [Swift4] 타입캐스팅 (Type Casting) : <https://flyburi.com/594>

[5] 스위프트 타입 캐스팅 swift type casting : <https://studyhard24.tistory.com/75>

References

[6] [swift] Type Casting : <https://zetal.tistory.com/entry/swift-기초문법-20-타입캐스팅Type-casting>

[7] [Swift 3] 타입 변환 (Type Casting) : <https://beankhan.tistory.com/170>

[8] [Xcode / Swift] Type Casting(형변환) 관련 | is, as, as?, as! 차이점 : <https://macinjune.com/all-posts/web-developing/swift/xcode-swift-type-casting형변환-관련-차이점/>

[9] Type Casting In Swift Explained : <https://learnappmaking.com/type-casting-swift-how-to/>

Thank you!