

## 1. 리터럴 문법에 대한 조사

클래스나, 구조체 같은 객체데이터도 상수가 될 수 있는 것이다. 단! 상수는 데이터가 변하지 않아야 한다. 그래서 객체를 상수로 지정 할 때, 객체안의 있는 데이터 까지도 변하지 않는 줄 착각 할 수 있지만, 객체 자체가 상수(메모리의 객체위치가 상수라는 의미)지, 안의 데이터들이 상수라는 의미가 아니다.

두번째로, 리터럴의 뜻은 데이터 그 자체를 뜻 한다. 그렇다면 인스턴스(클래스 데이터)가 리터럴이 될 수 있을까? 답은 아니오다. 만약 인스턴스안에 있는 값들을 변경하지 않는다면 모를까..보통의 인스턴스는 동적으로 사용하기 위해 작성되므로 리터럴이 될 수가 없다. 왜냐하면 값이 언제 바뀔지 모르는 것이므로...리터럴은 변수의 값이 변하지 않는 데이터(메모리 위치안의 값)를 의미한다.

### 1. NSArray 리터럴

```
26
27 // 이전에는
28 array = [NSArray arrayWithObjects:@"a", @"b", @"c", nil];
29
30 // 이제부터는
31 array = @[@"a",@"b",@"c"];
32
```

### 2. NSDictionary 리터럴

```
34 // 이전에는
35 dict = [NSDictionary dictionaryWithObjects:@[o1, o2, o3]forKeys:[k1, k2, k3]];
36
37 // 이제부터는
38 dict = @{ k1 : o1, k2 : o2, k3 : o3 };
```

### 3. NSNumber

```
40
41 // 이전에는
42 number = [NSNumber numberWithInt:'X'];
43 number = [NSNumber numberWithInt:12345];
44 number = [NSNumber numberWithUnsignedLong:12345ul];
45 number = [NSNumber numberWithLongLong:12345ll];
46 number = [NSNumber numberWithFloat:123.45f];
47 number = [NSNumber numberWithDouble:123.45];
48 number = [NSNumber numberWithBool:YES];
49
50 // 이제부터는
51 number = @'X';
52 number = @12345;
53 number = @12345ul;
54 number = @12345ll;
55 number = @123.45f;
56 number = @123.45;
57 number = @YES;
58
```