



코틀린 반복문 & 조건문 & 함수

App:ple 코틀린 2차시 | 강이자 최재혁

조건문

코틀린에서의 조건문은 무엇이고,
다른 언어와 무엇이 다른 지 살펴봅시다

01. If – else문

If = 만약에, ~라면

“만약 ~라면” 이라는 의미로 작동합니다

```
if (조건식) {  
    수행 할 문장  
}
```

If = 만약에, ~라면 + else = 다른

위의 조건과 다른 조건이 성립할 때 수행하라는 의미로 작동합니다

```
else if (조건식) {  
    수행 할 문장  
}
```

else = 다른

모든 조건을 성립하지 않을 때라는 의미로 작동합니다

```
else {  
    수행 할 문장  
}
```

조건식에 들어갈 문장	<, <=, >, >=, ==, != 관계(비교) 연산자	피연산자의 대소 관계를 나타낼 수 있는 연산자 값은 Boolean으로 나오며, true 혹은 false로 이야기함
	변수 is 자료형, 변수 !is 자료형	is) 변수의 자료형이 맞다면 True, 틀리다면 False !is) is의 반대 연산자로, 반대로 작동하는 연산자임
	범위를 나타내는 연산자	1 .. 100이라고 나타내게 된다면, 1부터 100까지의 범위를 가짐

02. When문과 Switch - Case

Switch - case

제어식의 값을 case 값과 비교해 실행 문장을 결정함

Case에는 제어식에서 값이 무엇이 들어갈 지 직접 지정 해준다

```
#include <stdio.h>
```

```
Int main() {  
    int rank = 1;  
    switch(rank){  
        case 1: printf("1등"); break;  
        case 2: printf("2등"); break;  
        case 3: printf("3등"); break;  
  
        default: printf("참가");  
    }  
}
```

차례대로 위부터 내려오면서 케이스 비교

break : 해당 조건문에서 탈출하는 역할을 함
(break가 없다면 성립한 줄부터 마지막까지 실행됨)

When

제어식의 값이 성립될 때까지 모든 인수를 검사

모든 인수를 검사하였을 때, 모두 아닌 경우인

else가 반드시 필요하다 (Switch에서는 default가 필수 X)

```
fun main() {  
    val sum = 100  
    when(sum){  
        100 -> print("만점")  
        in 90..100 -> print( " A " )  
        in 80..90 -> print( " B " )  
        in 70..80 -> print( " C " )  
        in 60..70 -> print( " D " )  
        else -> print("E")  
    }  
}
```

03. 조건문의 값을 변수에 담기

If - else문

```
fun main() {  
    val a = 5  
    val b = 3  
  
    val max = if(a>b){  
        a  
    } else {  
        b  
    }  
  
    print("$max")  
}
```

_0406_01_1_pptKt x

C:\Users\user\.jdk

5

when문

```
fun main() {  
    val sum = 100  
  
    var result = when(sum){  
        100 -> "만점"  
        in 90..100 -> "A"  
        in 80..90 -> "B"  
        in 70..80 -> "C"  
        in 60..70 -> "D"  
        else -> "E"  
    }  
  
    print("$result")  
}
```

_0406_01_pptKt x

C:\Users\user\.jd

만점

Process finished

04. 직접 조건문을 사용해보자!

When

제어식의 값이 성립될 때까지 모든 인수를 검사

모든 인수를 검사하였을 때, 모두 아닌 경우인

else가 반드시 필요하다 (Switch에서는 default가 필수 X)

```
fun main() {  
    val sum = 100  
    when(sum){  
        100 -> print("만점")  
        in 90..100 -> print( " A " )  
        in 80..90 -> print( " B " )  
        in 70..80 -> print( " C " )  
        in 60..70 -> print( " D " )  
        else -> print("E")  
    }  
}
```

When문을 사용하여 **아래 코드를 구현**해보자

1. 등수를 입력받는다
입력 방식은 상관 없음

2. 입력 받은 등수에 따라 등급을 출력한다

1등급 : 1등~12등

2등급 : 13~25등

3등급 : 26~45등

4등급 : 46등~

출력을 해봤다면, 조건문 자체를 변수에 담는 것도 해보세요

반복문

코틀린에서의 변수는 무엇이고,
다른 언어와 무엇이 다른 지 살펴봅시다

01. for문

반복할 범위를 지정해줌

```
for (변수 in 시작값..종료값) {  
    //내용  
}
```

* in은 변수를 시작값부터 종료값까지
증가시키면서 코드를 반복시켜줌

```
for(변수 in 시작값 until 종료값){  
    내용  
}
```

.. 대신 until을 사용하여 종료값 이전까지
반복할 수 있음 (in은 종료값까지 반복)

```
for(변수 in 시작값..종료값 step 증가폭){  
    내용  
}
```

Step을 이용하여 증가폭을 결정

```
for(변수 in 시작값 downTo 종료값){  
    내용  
}
```

downTo 를 이용하여 시작값부터 종료값까지
감소시키면서 반복시킴

02. while문

While = ~동안

주어진 조건이 참인 동안
수행 할 문장을 수행하며, 거짓이면 빠져나간다

```
while(조건){  
    실행할 문장  
}
```

Do - While = 한 번 실행 후, ~동안

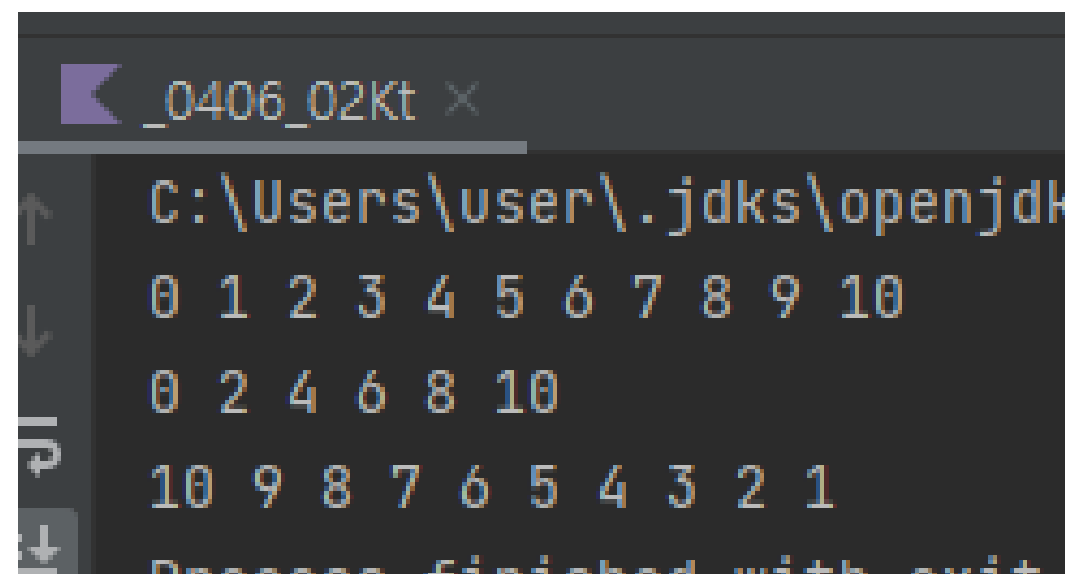
do문 안의 문장을 1회 실행 후,
주어진 조건이 참인 동안
수행 할 문장을 수행하며, 거짓이면 빠져나간다

```
do{  
    실행할 문장  
}while(조건)
```

03. 직접 반복문을 사용해보자!

반복문을 사용하여 **아래 코드를 구현**해보자

- 1) 1부터 10까지를 출력하세요
- 2) 1부터 10까지를 2 간격으로 출력하세요
- 3) 10부터 1까지 1 간격으로 반대로 출력하세요



```
_0406_02Kt ×
C:\Users\user\.jdk\openjdk
0 1 2 3 4 5 6 7 8 9 10
0 2 4 6 8 10
10 9 8 7 6 5 4 3 2 1
Process finished with exit
```

함수

코틀린에서의 함수는 무엇이고,
다른 언어와 무엇이 다른 지 살펴봅시다

01. 함수의 형식과 선언



코틀린 함수의 형식

```
fun 함수명(변수 : 자료형) : 리턴타입{
```

```
    함수 내부 코드  
    return 결과 값
```

```
}
```

이 때, 코틀린에서는 이런 식으로도 작성 가능하다

```
fun sum(a:Int, b:Int) : Int = a + b
```

함수가 한 줄로 정의될 때, 축약해 가독성 높이기 가능

02. Parameters



코틀린 함수의 형식

fun 함수명(변수 : 자료형) : 리턴타입{

함수 내부 코드
return 결과 값

}

Default parameters

아무런 값이 전달되지
않았을 경우의 기본값 설정 가능

```
fun example(speed : Int = 30){  
}
```

Required parameters

기본값이 지정되지 않은 경우,
반드시 값을 전달해주어야 함

```
fun example(speed : Int){  
}
```

자료형에 따른 변수 선언



변경여부 변수명 : 자료형 = 값 할당

형태	이름	크기 (sizeof)	사용 예시
정수형	Byte	1 byte	var num1 : Byte = 123
	short	2 byte	var num2 : Short = 123
	Int (기본)	4 byte	var num3 = 123 (10진수를 표기하려면 0x를, 2진수를 표기하려면 0b를 앞에 붙이기)
	long	8 byte	var num4 : Long = 123L (L을 통해 더 큰 메모리를 가진 변수임을 알려줘야 함)
실수형	float	4 byte	var num5 : Float = 123.5f (맨 뒤에 f를 붙여줘야 함)
	double (기본)	8 byte	var num6 = 123.4
문자형	Char (한 개의 문자)		var str1 = 'a'
문자열	String (한 개 이상의 문자)		var str2 = "String"
논리형	Boolean		var bool = true

배열 & Lists

코틀린에서의 배열은 무엇이고,
다른 언어와 무엇이 다른 지 살펴봅시다

01. 배열과 리스트

컬렉션 (Lists)	배열
<p>크기가 고정되지 않고 임의의 개수를 담을 수 있는 동적 배열</p> <p>컬렉션으로는 List, Map, set이 존재함</p> <p>동적으로 컬렉션을 사용하기 위해 자료형 앞에 "Mutable"을 붙임</p>	<p>저장될 수 있는 배열의 크기를 정해 놓고 사용함</p> <p>중간에 개수를 추가하거나 제거할 수 없음</p> <p>배열 요소는 변경이 가능함</p>

02. 배열 – arrayOf()

```
var dayArray = arrayOf("MON", "TUE", "WED", "THU", "FRI", "SAT", "SUN")
```

arrayOf() 함수를 사용해 배열 선언을 하고 값을 선언할 수 있음

```
var dayArray = intArrayOf(0, 1, 2, 3, 4, 5)
```

같은 자료형만 포함하도록 정의할 수도 있음

```
dayArray[0] = "monday"  
dayArray.set(0, "월요일싫어")
```

배열의 값을 변경하는 방법에는 두 가지가 존재함

```
var number = dayArray[0]  
var number = dayArray.get(1)
```

배열의 값에 접근하는 방법에도 두 가지가 존재함

02. 리스트 – mutableListOf()

```
var mutableList = mutableListOf("MON", "WED")
```

arrayOf() 함수를 사용해 배열 선언을 하고 값을 선언할 수 있음

```
var dayOfWeek = mutableList.get(1)
```

리스트에 저장된 값을 사용함

```
mutableList.removeAt(1)
```

리스트에 저장된 값을 제거시킴

```
mutableList.set(1, "FRI")
```

리스트에 저장된 값을 수정시킴

과제

배운 내용을 활용하여
과제를 수행 해봅시다

숫자야구 만들기 [조건 추가 ver.]



과제 요구 사항

01. 사용자가 프로그램을 실행시켰을 때, (화면 기본 구성)

- ☆ 사용자가 맞출 랜덤 숫자 3자리를 뽑는다, "숫자를 입력하세요 : "라고 출력한 뒤, 사용자에게 입력을 받을 수 있도록 한다
- ☆ 사용자가 수를 입력하거나, 랜덤 숫자 3자리를 뽑을 때 숫자가 겹치지 않도록 한다

02. 사용자가 입력했을 때,

- ☆ 입력을 받은 뒤, 사용자의 숫자에 따라 []S []B의 형식으로 코드를 내보낸다
- ☆ 3S인 경우(위치와 숫자 모두 동일) "게임에서 승리하였습니다, 숫자 : ____" 를 내보낸다 (이 때, 숫자는 랜덤 생성 숫자를 출력)
- ☆ 3S이 아닌 경우, 사용자가 다시 숫자를 맞출 수 있도록 한다
- ☆ 사용자가 10번 안에 맞추지 못한 경우, 사용자가 패배하며 10번 안에 맞춘 경우 사용자가 승리한다 (횟수 출력)

03. 과제 제출 방법 안내

숫자야구 게임 만들기가 오늘 과제입니다

아래 구글 폼으로 들어가 코틀린 파일을 모두 압축하여 제출 시
과제 제출로 인정하도록 하겠습니다 (제출 기한 : 04/13 수업 전까지)

구글 폼 링크

<https://forms.gle/Liogap1XMTfscYbx7>

과제에 대한 질문이 있을 경우

디스코드 : 재혁 #2293, 인스타 : @jaehyeok3017,
애플파이 12,13기 단체 카톡을 이용하여 질문을 하시면 됩니다.

04. 코틀린 랜덤 수 생성 코드

```
import java.util.Random
```

```
// random수는 변하지 않을 것이므로 val로 선언
```

```
val random = Random()
```

```
val num = random.nextInt(10) // 0~9 사이의 랜덤 수
```



코틀린 2차시 끝!

위 페이지에서 제시하였던
제출 기한까지 반드시 제출해주세요!

