

02

02-8. 예외처리

Basic Syntax

# 예외처리

---

## try-catch-finally 구문으로 예외처리

```
fun main(args: Array<String>) {    try {  
    println("try top...")  
  
    val data: String = "10"  
  
    val intData: Int? = data.toInt()  
  
    println("try bottom...")  
} catch (e: Exception){  
    println("catch.....")  
  
} finally {  
    println("finally....")  
}  
}
```

실행결과

try top...

try bottom...

finally....

# 예외처리

---

## try-catch-finally 구문으로 예외처리

```
fun main(args: Array<String>) {  
    try {  
        println("try top...")  
  
        val data: String = "kkang"  
  
        val intData: Int? = data.toInt()  
  
        println("try bottom...")  
    } catch (e: Exception) {  
        println("catch.....${e.toString()}")  
    } finally {  
        println("finally....")  
    }  
}
```

### 실행결과

```
try top...  
catch.....java.lang.NumberFormatException: For input string: "kkang"  
finally....
```

# 예외처리

## catch 가 여러 개 정의된 경우

```
fun some(array: Array<Any>){
    try {
        println("try top...")
        val intData: Int= array[0] as Int
        val data: String = array[2] as String
        val data2: Int = data.toInt()
    } catch (e: ClassCastException){
        println("catch... ClassCastException")
    } catch (e: ArrayIndexOutOfBoundsException){
        println("catch... ArrayIndexOutOfBoundsException")
    } catch (e: Exception){
        println("catch... Exception... ${e.toString()}")
    }
}

fun main(args: Array<String>) {
    //cast exception
    val array= arrayOf("0", 1, "6")
    some(array);
    //index out of bound exception
    val array2= arrayOf(10,"5")
    some(array2)
    //Number format exception
    val array3= arrayOf(10, 0, "world")
    some(array3)
}
```

### 실행결과

```
try top...
catch... ClassCastException
try top...
catch... ArrayIndexOutOfBoundsException
try top...
catch... Exception... java.lang.NumberFormatException: For input
string: "world"
```

# 예외처리

## 표현식으로 이용

- try - catch 구문은 표현식으로도 사용이 가능
- 각 영역의 마지막 줄이 리턴값
- finally 부분은 표현식으로 이용되지는 않는다.

```
fun some1(arg: String): Int {  
    val parseData: Int = try {  
        println("try..top")  
        arg.toInt()  
    } catch (e: Exception){  
        println("${e.toString()}")  
        0  
    } finally {  
        println("finally...")  
        100  
    }  
    return parseData  
}  
fun main(args: Array<String>) {  
    println("${some1("10")}")  
  
    println("${some1("a")}")  
}
```

### 실행결과

```
try..top  
finally..  
10  
try..top  
java.lang.NumberFormatException: For input string: "a"  
finally..  
0
```

# 예외처리

---

## 예외 발생시키기

```
fun some(arg: Int): Int{
    if(arg < 1)
        throw Exception("parameter must be greater than zero")
    else {
        var sum=0
        for(i in 1..arg){
            sum += i
        }
        return sum
    }
}

fun main(args: Array<String>) {
    try {
        println("${some(5)}")

        println("${some(-1)}")

        println("main bottom....")
    } catch (e: Exception){
        println("Exception.... ${e.toString()}")
    }
}
```

# 예외처리

---

## Exception 클래스 정의

```
class MyException(msg: String): Exception(msg){
    val errorData: String = "some error data"
    fun errorFun(){
        println("errorFun call....")
    }
}

fun some1(){
    throw MyException("My Error...")
}

fun main(args: Array<String>) {
    try {
        some1()
    } catch (e: MyException){
        println("error message : ${e.toString()}")
        println("error data : ${e.errorData}")
        e.errorFun()
    }
}
```



# 감사합니다

단단히 마음먹고 떠난 사람은  
산꼭대기에 도착할 수 있다.  
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어  
William Shakespeare