Scala 내부 속성과 메소드 조회

myjlms99@gmail.com

특정 클래스 타입 조회

특정 타입 조회

scala REPL 에서 :type 명령어나 getClass 메소드로 확인하기

```
scala> val a = 1
a: Int = 1

scala> :type a
Int

scala> a.getClass
res194: Class[Int] = int
```

특정 타입 이름 조회

getClass메소드로 클래스를 가져오고 그 클래스 내부의 getName 메소드로 조회하기

```
scala> a.getClass
res194: Class[Int] = int

scala> a.getClass.getName
res195: String = int
```

타입 조회 함수 만들기

실제 내부의 타입을 조회하기 위해 match 구문을 이용해서 타입을 리턴할 수 있다.

```
scala> def typeC(a:Any) = a.getClass
typeC: (a: Any)Class[_]

scala> typeC(1)
res231: Class[_] = class java.lang.Integer

scala> def typeD(a:Any) = {
     |     a match {
     |         case a:Int => "Int"
     |     }
     | }
typeD: (a: Any)String

scala> typeD(1)
res232: String = Int
```

Array 타입 이해하기

배열을 원소 조회 및 갱신

배열을 생성하고 indexing 하려면 apply 메소드를 이용해서 조회한다. (인덱스) 를 넣고 조회하거나 갱신이 가능하다.

```
scala> val a = Array(1,2,3,4,5)
a: Array[Int] = Array(1, 2, 3, 4, 5)

scala> a(1)
res255: Int = 2

scala> a(1) = 34

scala> a
res257: Array[Int] = Array(1, 34, 3, 4, 5)

scala> a.apply(1)
res258: Int = 34
```

다차원 배열

다차원 배열의 생성은 ofDim 메소드를 이용해서 처리한다. 조회하거나 갱신할 때는 행과 열에 따라 구분해서 조회해야 한다

```
scala> val ad = Array.ofDim[Int](3,4)
ad: Array[Array[Int]] = Array(Array(0, 0, 0, 0), Array(0, 0, 0, 0), Array(0, 0,
0, 0))

scala> Array(Array(0,0,0,0), Array(0,0,0,0), Array(0,0,0,0))
res261: Array[Array[Int]] = Array(Array(0, 0, 0, 0), Array(0, 0, 0, 0), Array(0,
 0, 0, 0))

scala> ad(1)(1)
res262: Int = 0

scala> ad(1)
res263: Array[Int] = Array(0, 0, 0, 0)
```

배열을 생성해서 원소 분리

배열을 생성해서 실제 값을 하나와 나머지 값을 비교하기 head, tail은 좌측에서 우측으로 last, init 은 우측에서 좌측으로 처리한다.

```
scala> val a = Array(1,2,3,4)
a: Array[Int] = Array(1, 2, 3, 4)

scala> a.head
res201: Int = 1

scala> a.last
res202: Int = 4

scala> a.tail
res203: Array[Int] = Array(2, 3, 4)

scala> a.init
res204: Array[Int] = Array(1, 2, 3)
```

Map 메소드 처리

map 함수는 고차함수로써 내부에 함수나 메소드를 전달받아 처리한다.

```
scala> a.map(_ * 3)
res205: Array[Int] = Array(3, 6, 9, 12)

scala> a.map(_.toString)
res206: Array[String] = Array(1, 2, 3, 4)

scala> a.map(_.toDouble)
res207: Array[Double] = Array(1.0, 2.0, 3.0, 4.0)
```

배열 변환 : for yield

배열을 전체를 for yield 구문을 위해 새로운 배열로 생성한다. Map 메소드를 이용해서 처리도 가능하다.

```
scala> val a = Array(1,2,3,4,5)
a: Array[Int] = Array(1, 2, 3, 4, 5)

scala> var result = for (elem <- a) yield 2 * elem
result: Array[Int] = Array(2, 4, 6, 8, 10)

scala> a.map(_*2)
res254: Array[Int] = Array(2, 4, 6, 8, 10)
```

인자 추론

map 함수는 익명함수를 전달을 받을 경우 인자에 대한 타입추론도 가능하지만 인자를 추론해서 사용할 수 있다

```
scala> a.map((x:Int) => x*3)
res215: Array[Int] = Array(3, 6, 9, 12)

scala> a.map((x) => x*3)
res216: Array[Int] = Array(3, 6, 9, 12)

scala> a.map(_ * 3)
res217: Array[Int] = Array(3, 6, 9, 12)
```

Foreach 메소드

foreac 메소드는 내부의 인자들을 하나씩 검색해 주는 역할을 한다. 인자로 받은 함수를 가지고 처리해서 결과를 보여준다.

```
scala> a.foreach(println)
1
2
3
4
```

타입 변환 메소드

배열을 집합과 리스트 타입으로 변환도 가능하다.

```
scala> a
res219: Array[Int] = Array(1, 2, 3, 4)

scala> a.toSet
res220: scala.collection.immutable.Set[Int] = Set(1, 2, 3, 4)

scala> a.toList
res221: List[Int] = List(1, 2, 3, 4)
```

정렬하기

sorted와 reverse 메소드를 이용해서 정렬 처리를 할 수 있다.

```
scala> a
res223: Array[Int] = Array(1, 2, 3, 4)

scala> a.reverse
res224: Array[Int] = Array(4, 3, 2, 1)

scala> val b = a.reverse
b: Array[Int] = Array(4, 3, 2, 1)

scala> b
res225: Array[Int] = Array(4, 3, 2, 1)

scala> b.sort
sortBy   sortWith   sorted

scala> b.sorted
res226: Array[Int] = Array(1, 2, 3, 4)
```

메소드 체인 사용

메소드를 연속해서 부르기

메소드 결과값이 객체가 나오면 필요한 메소드를 이용해서 호출해서 메소드를 연결해서 하나의 결과를 나올 때까지 처리한다

```
scala> b
res228: Array[Int] = Array(4, 3, 2, 1)

scala> b.sorted.map(_ *3)
res229: Array[Int] = Array(3, 6, 9, 12)

scala> b.sorted.map(_ *3).filter(_ % 2==0)
res230: Array[Int] = Array(6, 12)
```

내부 속성과 메소드 조회

내부 정보 조회

인스턴스에 점연산를 치고 tab을 누르면 내부의 속성과 메소드가 나온다.

```
scala> val a = 1
a: Int = 1

scala> a.hashCode
res196: Int = 1

scala> a.
!=    >             floatValue      isValidInt      to              toRadians
%     >=            floor           isValidLong     toBinaryString  toShort
&     >>            getClass        isValidShort    toByte          unary_+
*     >>>           intValue        isWhole         toChar          unary_-
+     ^             isInfinite      longValue       toDegrees       unary_~
-     abs           isInfinity      max             toDouble        underlying
/     byteValue     isNaN           min             toFloat         until
<     ceil          isNegInfinity   round           toHexString     |
<<    compare       isPosInfinity   self            toInt
```

필드 이름 조회

하나의 함수를 만들어서 내부 클래스의 필드를 조회해서 출력한다.

```
scala> def dirF[T](v:T) {
     |   v.getClass.getFields.map(_.getName).toSet.toList.foreach(println)
     | }
dirF: [T](v: T)Unit

scala> dirF[Int](100)
MIN_VALUE
SIZE
BYTES
MAX_VALUE
TYPE
```

메소드 이름 조회

하나의 함수를 만들어서 내부 클래스의 메소드를 조회해서 출력한다.

```
scala> def dir[T](v:T) {
     |   v.getClass.getMethods.map(_.getName).toSet.toList.foreach(println)
     | }
dir: [T](v: T)Unit

scala> dir[Int](100)
toHexString
compareUnsigned
notify
wait
valueOf
divideUnsigned
doubleValue
equals
bitCount
rotateRight
reverseBytes
remainderUnsigned
min
decode
notifyAll
longValue
```

Int 내부 조회

Int로 생성한 것을 처리하면 연산자 메소드를 제외한 메소드들이 보인다.

```
scala> dir(a)
toHexString
compareUnsigned
notify
wait
valueOf
divideUnsigned
doubleValue
equals
bitCount
rotateRight
reverseBytes
remainderUnsigned
min
decode
notifyAll
longValue
numberOfLeadingZeros
shortValue
compareTo
reverse
toUnsignedString
max
```