

그렇다면 type과 mode의 차이는 무엇인가?

먼저, type은 R언어 자체에서 사용하는 분류이다.

R언어는 C언어를 이용해 코딩되었는데, 이 C언어 상에서 실제로 서로 구분하는 type을 나타낸다.

반면 mode는 "Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole." 이 정의한 분류 방법으로, 다른 S language 와 호환성을 가지고 있는 것이 특징이다.

이와 같이 object를 분류하는 방법이 추가로 생기게 된 이유는 R 언어가 지속적으로 긴 시간 개량되어 왔기 때문이라고 한다.

mode는 old style의 분류 방법이고, typeof 로 확인할 수 있는 type이 보다 정확한 방법의 분류이다.

type은 typeof function을 통해 확인이 가능하다.

"logical"	a vector containing logical values
"integer"	a vector containing integer values
"double"	a vector containing real values
"complex"	a vector containing complex values
"character"	a vector containing character values
"raw"	a vector containing bytes
"list"	a list
"NULL"	NULL
"closure"	a function
"special"	an internal function that does not evaluate its arguments
"builtin"	an internal function that evaluates its arguments
"environment"	an environment
"S4"	an S4 object which is not a simple object

## mode 사용의 예)

### ① numeric (숫자)

R언어는 숫자 표현에 있어 **double**을 기본으로 하고 있다. type에는 integer과 double이 있지만 numeric이라고 표현되어 있는 곳은 double이라고 바꿔도 동일한 의미라고 보면 된다. R console내 `help(numeric)`를 해보면 아래와 같이 기술되어 있다.

“ numeric is identical to double (and real). It creates a double-precision vector of the specified length with each element equal to 0.

반대로 `help(double)`을 해보면 다음과 같이 기술되어 있다.

“ double creates a double-precision vector of the specified length. The elements of the vector are all equal to 0. It is identical to numeric.

numeric의 예)

```
1 > number=10
2 > number
3 [1] 10
4 > mode(number)
5 [1] "numeric"
6 > is.numeric(number)
7 [1] TRUE
8
```

```
> number=10
> number
[1] 10
> mode(number)
[1] "numeric"
> is.numeric(number)
[1] TRUE
```

## ② character (문자)

C언어에서는 character라고 하면 문자 하나만을 의미한다. 반면 R언어에서는 문자열 또한 character라고 표현한다. 다른 언어와 마찬가지로 큰따옴표(")나 작은따옴표(')를 이용해 정의한다.

character의 예)

```
1 > ch = "Hello"
2 > ch
3 [1] "Hello"
4 > mode(ch)
5 [1] "character"
6 > is.character(ch)
7 [1] TRUE
```

```
> ch = "Hello"
> ch
[1] "Hello"
> mode(ch)
[1] "character"
> is.character(ch)
[1] TRUE
```

## ③ logical (논리, TRUE/FALSE)

일반적으로 조건문에서 사용 가능한 mode이다. TRUE와 FALSE, 두 값 중 하나를 갖는다. 혹시라도 큰따옴표(")나 작은따옴표(')와 함께 쓰인다면 character로 인식되기 때문에 주의해야 한다.

logical의 예)

```
1 > lo = TRUE
2 > lo
3 [1] TRUE
4 > mode(lo)
5 [1] "logical"
6 > is.logical(lo)
7 [1] TRUE
```

```
> lo = TRUE
> lo
[1] TRUE
> mode(lo)
[1] "logical"
> is.logical(lo)
[1] TRUE
```

+ ④ list

## mode간 변환

잘못된 연산의 예)

```
1 > "3" + "5"
2 Error in "3" + "5" : non-numeric argument to binary operator
```

```
> "3" + "5"
Error in "3" + "5" : non-numeric argument to binary operator
```

당연한 이야기이겠지만 위와 같이 mode가 character 인 경우 사칙 연산이 불가능하다. 이와 같을 때는 mode를 바꾸어주어야 할 필요가 있다.

mode 변환의 예)

```
1 > as.character(3)
2 [1] "3"
3 > as.numeric("3")
4 [1] 3
5 > as.logical("TRUE")
6 [1] TRUE
```

```
> as.character(3)
[1] "3"
> as.numeric("3")
[1] 3
> as.logical("TRUE")
[1] TRUE
```