

Type of Data

By 윤명근 / 박수현

수업목표

- Numeric, character
- String data
- Boolean data
- Changing types
- `type()`
- Type-conversion errors
- Type conversions
- 심화학습

Numeric, character

- 정수, 실수, 문자열

```
>>> a = 10
>>> b = 3.14
>>> c = 'cat'
>>> print(a, b, c)
10 3.14 cat
>>>
>>> a = 20.0
>>> c = 100
>>> print(a, b, c)
20.0 3.14 100
>>>
```

String

- 문자열 인덱스(index)와 부분문자(substring) 출력

2_string_test1.py

File Edit Format Run Options Window Help

```
str = "01234567"

print("str = ", str)

print("str[3] = ", str[3])          # (3+1) 번째

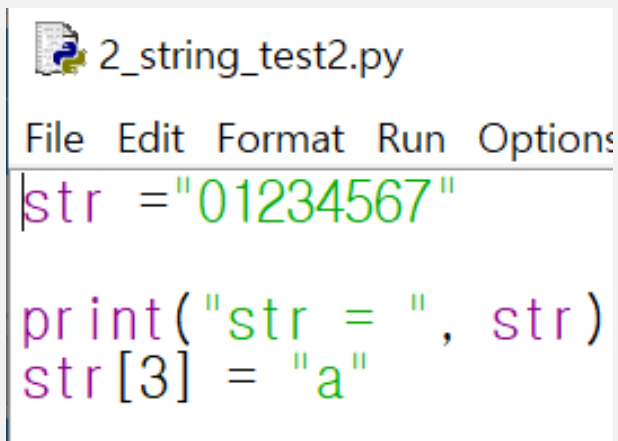
print("str[0:3] = ", str[0:3])      # 0 ~ (3-1) 번째
print("str[0:5] = ", str[0:5])      # 0 ~ (5-1) 번째

print("str[2:5] = ", str[2:5])      # 2 ~ (5-1) 번째
print("str[4:6] = ", str[4:6])      # 4 ~ (6-1) 번째
print("str[4:8] = ", str[4:8])      # 4 ~ max 번째
```

```
str = 01234567
str[3] = 3
str[0:3] = 012
str[0:5] = 01234
str[2:5] = 234
str[4:6] = 45
str[4:8] = 4567
>>>
```

String

- 문자열은 **불변(immutable)** 데이터 타입
 - 변경이 불가능함. 새롭게 생성하여 할당함.



```
2_string_test2.py
File Edit Format Run Options
str = "01234567"
print("str = ", str)
str[3] = "a"
```

```
>>>
===== RESTART: C:\W과소사\W과소사-강의예제\2_string_test2.py =====
=====
str = 01234567
Traceback (most recent call last):
  File "C:\W과소사\W과소사-강의예제\2_string_test2.py", line 4, in <module>
    str[3] = "a"
TypeError: 'str' object does not support item assignment
>>>
```

String

- 불변(immutable) 데이터 타입
 - 변경 불가능
 - Ex) 튜플(tuple), 문자열(string) 등
- 가변(mutable) 데이터 타입
 - 변경 가능
 - 리스트(list), 사전(dictionary), 집합(set) 등

Boolean Data

- Boolean
 - 참(true)과 거짓(false) 표현
 - 수학자 George Boole 발명
 - “True”는 0 이 아닌 모든 실수/정수 (non zero), null이 아닌 string
 - “False”는 0 (zero)
 - null string ("") False
 - bool() 함수 : Boolean 값 알려줌

```
>>> a = 10
>>> a < 1
False
>>> a > 9
True
>>> True + True
2
>>> bool(10)
True
>>> bool(0)
False
>>> bool("abc")
True
>>> bool("")
False
>>>
```

Boolean Data

- and, or, not
 - Boolean 문장 조합 생성
 - A and B: A와 B가 동시에 True인 경우에만 True
 - A or B: A와 B 적어도 하나가 True인 경우에 True
 - not A : A의 반대에 해당하는 bool값 생성



2.3example3.py

File Edit Format Run Options Window Help

```
print()

print ("(100 > 20) and (30==30):", (100 > 20) and (30==30))

print ("(100 > 20) or (30 !=30):", (100 > 20) or (30 !=30))

print ("not (30 !=30):", not (30 !=30))

print()
```

(100 > 20) and (30==30): True
(100 > 20) or (30 !=30): True
not (30 !=30): True

Changing types (Type casting, Type conversion)

- `int()` : 정수로 변환
- `float()` : 실수로 변환
- `bool()`: Boolean으로 변환
- `chr(숫자)`: 숫자에 해당하는 ASCII 문자로 변환
- `ord(문자)`: ASCII 문자에 해당하는 숫자로 변환

```
2.3example4.py
File Edit Format Run Opti
print (int(3.14))
print (float(10))
print (bool(100))
print (bool(-10))
print (bool(0))
print (chr(97))
print (ord('a'))
```

```
>>>
=====
=====
3
10.0
True
True
False
a
97
>>>
```

Changing types

- ASCII Table

<https://www.lookuptables.com/>

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Type()

- 변수가 가리키는 객체(object)의 타입을 알고 싶을 때 사용함

```
data_type_1.py
File Edit Format Run Options Wi
x = 3.14
y = 10
z = "3.14"
u = 'KMU Software'
v = [1, 3, "Software"]

print()
print(type(x))
print(type(y))
print(type(z))
print(type(u))
print(type(v))
```

```
<class 'float'>
<class 'int'>
<class 'str'>
<class 'str'>
<class 'list'>
>>> |
```

Type()

- 타입은 변수가 결정하는 것이 아니라 **변수가 가리키는 객체가 결정함**

```
data_type_2.py
File Edit Format Run Options W
x = 3.14
print()
print(type(x))

x = 10
print()
print(type(x))

x = 'KMU Software'
print()
print(type(x))

x = [1, 3, "Software"]
print()
print(type(x))
```

```
<class 'float'>
<class 'int'>
<class 'str'>
<class 'list'>
>>>
```

Type conversions

- Python 2.x 버전 주의 사항
 - 정수와 정수를 '/' 연산자로 나누면 정수 몫 계산
 - 실수 몫이 필요한 경우에는 나누어지는 수를 float()로 강제로 형변환시켰어야했음
 - $x=7/3$ vs $x=\text{float}(7)/3$ 또는 $x=7.0/3$
 - Python 3.x 버전에서는 '/'와 '//' 연산자 구분