

# 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(定數, integer), 실수(부동소수점)(實數, 浮動小數點, floating number), 문자열(文字列, string)

```
>>> 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
  - n: 길이
  - Index 0 ~ n-1

2\_string\_test1.py

File Edit Format Run Options Window Help

```
str = "01234567"
```

```
print(">> str = ", str)
print(">> 길이 = ", len(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
```

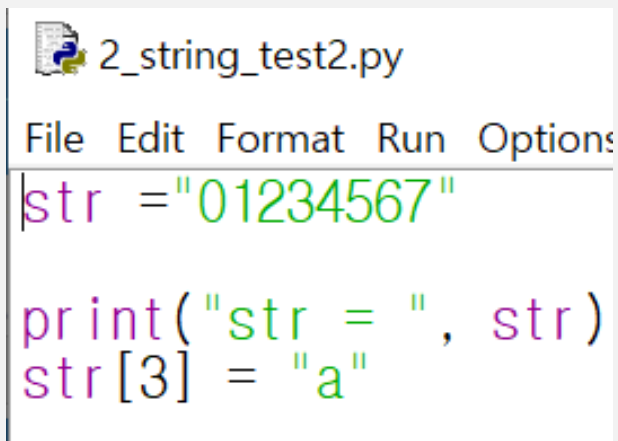
```
print(">> str[4:6] = ", str[4:6])      # 4 ~ (6-1) 번째
```

```
print(">> str[4:8] = ", str[4:8])      # 4 ~ max 번째
```

```
>> str = 01234567
>> 길이 = 8
>> 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

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

 \*2\_string\_test3.py

File Edit Format Run Options Window Help

```
# 문자열은 불변(immutable) 데이터 타입
# 변경이 불가능함.
```

```
str = "01234567"
```

```
print("1) str = ", str)
print("    id(str) : ", id(str))
```

```
#str[3] = "a"
```

```
# 새롭게 생성하여 할당함.
str = "012a4567"
```

```
print("\n2) str = ", str)
print("    id(str) : ", id(str))
```

```
1) str = 01234567
   id(str) : 2157527071536
2) str = 012a4567
   id(str) : 2157558755632
>>>
```

# 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'))
```

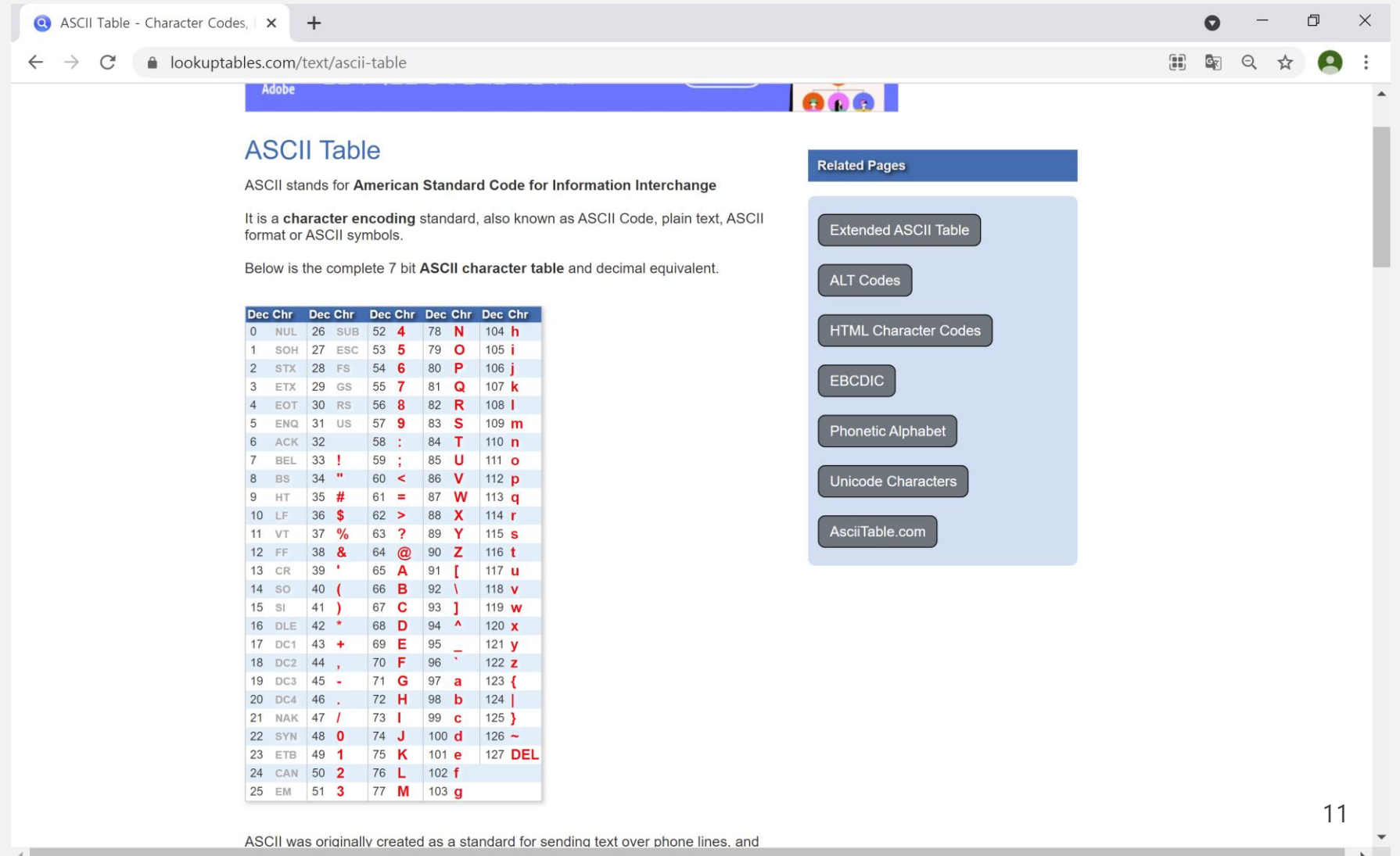
```
>>>
=====
=====
3
10.0
True
True
False
a
97
>>>
```

# Changing types

- ASCII Table

- ASCII :  
American Standard Code  
for Information Interchang

<https://www.lookuptables.com/text/ascii-table>



ASCII Table

ASCII stands for **American Standard Code for Information Interchange**

It is a **character encoding** standard, also known as ASCII Code, plain text, ASCII format or ASCII symbols.

Below is the complete 7 bit **ASCII character table** and decimal equivalent.

Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr
0	NUL	26	SUB	52	4	78	N	104	h
1	SOH	27	ESC	53	5	79	O	105	i
2	STX	28	FS	54	6	80	P	106	j
3	ETX	29	GS	55	7	81	Q	107	k
4	EOT	30	RS	56	8	82	R	108	l
5	ENQ	31	US	57	9	83	S	109	m
6	ACK	32		58	:	84	T	110	n
7	BEL	33	!	59	;	85	U	111	o
8	BS	34	"	60	<	86	V	112	p
9	HT	35	#	61	=	87	W	113	q
10	LF	36	\$	62	>	88	X	114	r
11	VT	37	%	63	?	89	Y	115	s
12	FF	38	&	64	@	90	Z	116	t
13	CR	39	'	65	A	91	[	117	u
14	SO	40	(	66	B	92	\	118	v
15	SI	41	)	67	C	93	]	119	w
16	DLE	42	*	68	D	94	^	120	x
17	DC1	43	+	69	E	95	_	121	y
18	DC2	44	,	70	F	96	`	122	z
19	DC3	45	-	71	G	97	a	123	{
20	DC4	46	.	72	H	98	b	124	
21	NAK	47	/	73	I	99	c	125	}
22	SYN	48	0	74	J	100	d	126	~
23	ETB	49	1	75	K	101	e	127	DEL
24	CAN	50	2	76	L	102	f		
25	EM	51	3	77	M	103	g		

Related Pages

- Extended ASCII Table
- ALT Codes
- HTML Character Codes
- EBCDIC
- Phonetic Alphabet
- Unicode Characters
- AsciiTable.com

ASCII was originally created as a standard for sending text over phone lines. and

# 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 가리키고 있는 객체 (변수에 저장된 값 객체)가 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 형변환 시켰어야했음
    - $x=7/3$  vs  $x=\text{float}(7)/3$  또는  $x=7.0/3$
  - Python 3.x 버전에서는 '/'와 '//' 연산자 구분