

02. 타입

정수형 타입

- 파이썬의 정수 는 크기 에 구애 없이 모든 숫자를 담을 수 있다.
- . 정수 타입 의 변수 a 에 2의 100승 의 결과를 할당 후 출력할 수 있다.

```
02_type[타입].py X
Chap01_intro > 02_type[타입].py > ...
1 a = 2 ** 100
2 print(a)
3
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/02_type[타입].py

D:\Python>

- . 음수 의 자리수 도 구애 없이 할당 할 수 있다.
- > 2 의 100 승 결과에서 3의 200승 결과 를 뺀 결과를 표현

```
02_type[타입].py X
Chap01_intro > 02_type[타입].py > ...
1 a = 2 ** 100
2 b = 3 ** 200
3 print(a-b)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/02_type[타입].py

D:\Python>

진법의 표현

- 컴퓨터가 사용하는 진수 의 종류

진법	접두	사용가능한 숫자	사용예시
2진법	0x	0,1	0b11
8진법	0o	0~7	0o10
16진법	0b	0~9 / a~f	0xB

```
02_type[타입].py X
Chap01_intro > 02_type[타입].py
1 print(0b11)
2 print(0o10)
3 print(0xb)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/02_type[타입].py

3
8
11

. 2진수 11 을 10진수로 표현

10진수	0	1	2	3
2진수	0	1	10	11

. 8진수 10 을 10진수로 표현

10진수	0	1	2	3	4	5	6	7	8	9
8진수	0	1	2	3	4	5	6	7	10	11

. 16진수 b 를 10진수로 표현

10진수	0	1	2	3	4	5	6	7	8	9	10	11
16진수	0	1	2	3	4	5	6	7	8	9	a	b

- 다른 진법으로 숫자 표현

. Hex, oct, bin

```

28  # # 다른 진법으로 숫자 표현
29  print(bin(3)) # 10진수 3 을 2 진법으로
30  print(oct(8)) # 10진수 8 을 8진법으로
31  print(hex(11)) # 10진수 11 을 16진법으로
32
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirstPython\0.Source\python\debugpy\adapter\python\python-2023.16.0\pythonFiles\lib\python\debugpy\python\0.Source\Python\02_type[타입].py" "
0b11
0o10
0xb

```

실수 형

- 소수점 을 가진 숫자 데이터 유형

. e12 : 10 의 12승 (1000000000000)

> 9.46 * 10 ** 12

```

58  light = 10 # 정수형
59  light = 10.23393 # 실수형
60
61  # 실수 형의 다른 표현
62  light = 9.46e12 # e12 : 10 의 12승 (1000000000000)
63  print(light) # 9.46 * 10 ** 12
64  print(9.46 * 10**12)
65
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirstPython\0.Source\Python> cmd /C "D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirstPython\0.Source\python\debugpy\adapter\python\python-2023.16.0\pythonFiles\lib\python\debugpy\python\0.Source\Python\02_type[타입].py" "
9460000000000.0
9460000000000.0

```

< 결과가 같은 것을 확인 할 수 있다. >

문자열

- 일련의 문자를 따옴표 로 감싸 나열해 둔 형식
- . 쌍따옴표 또는 홑따옴표 로 표현 가능

```
02_type[타입].py
Chap01_intro > 02_type[타입].py > ...
1 strValue = "안녕하세요"
```

- 확장열
- . 문자열을 나타내는 따옴표 안에서 동작하는 기능.
- > \w(역슬러쉬) 로 나타 낸다.

확장열	설명
\n	개행
\t	탭
\"	큰따옴표
\'	작은따옴표
\\	\ 를 문자로

> 사용 예시

```
02_type[타입].py x
Chap01_intro > 02_type[타입].py
1 print("안녕하세요 \"파이썬\" 문법 중\n확장열 에 관한 예시 입니다.")
2

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/02_type[타입].py
안녕하세요 "파이썬" 문법 중
확장열 에 관한 예시 입니다.
```

- . \w 를 많이 사용하는 문자열
- > r"문자열" 로 \w\w 를 여러 번 쓰지 않고 표현 할 수 있다.

```
02_type[타입].py x
Chap01_intro > 02_type[타입].py
1 print(r"C:\User\Python")
2

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/02_type[타입].py
C:\User\Python
```

< 파일 경로 등을 문자열로 나타내고자 할때 용이하다 >

- 장문의 문자열을 표현
- . " 로 문자열을 나타 내기에는 다소 불편한 점이 있다.

```
02_type[타입].py x
Chap01_intro > 02_type[타입].py
1 print(" 파이썬 에 관한 설명을 문자열로 나타내고 싶은데.. 너무 길어서 \n확장열을 쓰니 한줄에 표현이라 너무길고 ")
2 print(" 줄을 바꾸니 매번 다른줄에 print() 쓰고.. \" 로 열고 닫고\" 넘나 귀찮아...")
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/02_type[타입].py
파이썬 에 관한 설명을 문자열로 나타내고 싶은데.. 너무 길어서
확장열을 쓰니 한줄에 표현이라 너무길고
줄을 바꾸니 매번 다른줄에 print() 쓰고.. " 로 열고 닫고" 넘나 귀찮아...
```

. 삼겹 따옴표[""] 을 사용해 장문의 문자열을 간단히 표현 할 수 있다.

```
02_type[타입].py X
Chap01_intro > 02_type[타입].py > ...
1 Message = """ 파이썬 에 관한 설명을 문자열로 나타내고 싶은데.. 너무 길어서
2 ["\"\"\""] 을 사용 하였더니
3 여러줄을 편리하게 나타낼수 있게 되었습니다 """
4
5 print(Message)
6
```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/02_type[타입].py

파이썬 에 관한 설명을 문자열로 나타내고 싶은데.. 너무 길어서
["\"\"\""] 을 사용 하였더니
여러줄을 편리하게 나타낼수 있게 되었습니다.

. 삼겹 따옴표 내에 문자열 \ 를 사용하면 문자열을 개행 하지 않고 연이어 표현 할 수 있다.

```
02_type[타입].py X
Chap01_intro > 02_type[타입].py > ...
1 Message = """ 파이썬 에 관한 설명을 문자열로 나타내고 싶은데.. 너무 길어서 \
2 ["\"\"\""] 을 사용 하였더니 \
3 여러줄을 편리하게 나타낼수 있게 되었습니다. """
4
5 print(Message)
6
```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/02_type[타입].py

파이썬 에 관한 설명을 문자열로 나타내고 싶은데.. 너무 길어서 ["\"\"\""] 을 사용 하였더니 여러줄을 편리하게 나타낼수 있게 되었습니다.

- 문자 코드 와 문자 (문자 하나 문자열 은 문자의 모음)

. ord() 와 chr()

* 아스키 코드 표 참조 <https://stepbystep1.tistory.com/10>

```
02_type[타입].py X
Chap01_intro > 02_type[타입].py > ...
1 a = ord('a')
2 b = chr(97)
3 print(a)
4 print(b)
5 print(chr(0x61)) # print (hex(97))
6 print(chr(0b1100001)) # print (bin(97))
7
```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/02_type[타입].py

97
a
a

- > ord() : 문자를 아스키 10진수 아스키코드 로 나타낸다.
- > chr() : 아스키 코드 번호 를 문자 로 나타낸다.

제어 문자		공백 문자		구두점		숫자		알파벳		
10진	16진	문자	10진	16진	문자	10진	16진	문자	16진	문자
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63

* 아스키 코드 와 유니코드

. <https://code-lab1.tistory.com/233>

. 컴퓨터 로 사용할수 있는 문자(한글,영어,숫자,특수문자....) 등을 표준화 하여
정의를 해 둔 부호 체계

진위 형

- 참과 거짓을 0 과 1 로 표현하고 싶은데 표현력이 좋은 키워드(True / False) 로 할당
- 참 또는 거짓의 상태를 표현하는 데이터 타입
- True 또는 False 의 값을 가진다 (첫 글자는 반드시 대문자로 표현한다.)

```
02_type[타입].py x
Chap01_intro > 02_type[타입].py > ...
1 a = 5
2 b = a == 5
3 print(b)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

Python>C:/Users/MasterD/AppData/Local/Pr
True
```

- > 1행 : a 에 5 를 할당
- > 2행 : a 가 5 인지 비교 (참이면 True / 거짓이면 False) 후 결과를 b 에 할당
* == : 비교 연산자 참 / 거짓을 판단 하기 위한 연산 문법
- > 3행 : b 를 출력

- None (타언어에서는 Null 이라고 하며 Python 에서 는 Null 을 None 로 표현한다)
- None 은 아무런 값이 할당 되지 않은 상태를 말한다.
- > ex) 아파트 청약 후 아직 만들어 지지 않은 아파트의 세대주

```
185 a = None
186 print(a == None)
187 print(a)
188
189
190 a = ''
191 print(a == None)
192 print(a , '<--- 아무값도 없어요' , sep = '')
193

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirstPython\0.Source\Python>
cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe c:\Us
/./././debugpy\launcher 62461 -- "d:\MyDoc\11.EDUCATION\1. Master Source\15.Pytho
True
None
False
<--- 아무값도 없어요
```

- >185행 : a 의 상태를 None 로 할당
- >186행 : a 의 상태가 None 인지 판단결과 출력
- >187행 : a 를 출력
- >190행 : a 에 아무것도 없는 문자를 할당
- >191행 : a 가 None 인지 판단결과 출력
- >192행 : a 를 출력

컬렉션 (다양한 컬렉션은 추후 확인)

- 리스트
- 동일한 데이터 유형의 여러 값을 하나의 집합으로 묶어 서 특정 변수 에 할당하는 형태
- 데이터를 변경 할 수 있다.

```
리스트_변수명 = ['값1', '값2', '값3', '값4']
```

리스트				
데이터	값1	값2	값3	값4
위치(index)	0	1	2	3

* Index 시작점은 0 부터 시작한다.

- 각각의 데이터가 등록 된 위치 의 주소를 Index 라 한다.
- > Values 라는 문자열 리스트에 담긴 값 중 0 번째 Index 에 있는 값을 출력

```
02_type[타입].py x
Chap01_intro > 02_type[타입].py > ...
1
2 Values = ['값1', '값2', '값3', '값4']
3 print(Values[0])

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/pyth
값1
```


> Values 의 0 Index 의 값을 '가압1' 로 변경 후 출력

```
02_type[타입].py
Chap01_intro > 02_type[타입].py > Values
1 Values = ['값1', '값2', '값3', '값4']
2 print(Values[0])
3 Values[0] = '가압1'
4 print(Values[0])
5

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/pyt
값1
가압1
```

> 3행 : 0 위치의 데이터 를 '가압1' 로 변경

데이터	가압1
위치(index)	0

- 튜플 (Tuple)

- 리스트 와 유사한 형태 이나 값을 수정 할 수 없다.
- 튜플의 데이터 집합은 ' () ' 로 할당한다.

```
02_type[타입].py x
Chap01_intro > 02_type[타입].py > ...
1 Values = ('값1', '값2', '값3', '값4')
2 print(Values[0])
3 Values[0] = '가압1'
4 print(Values[0])
5

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python
값1
Traceback (most recent call last):
  File "d:\Python\Chap01_intro\02_type[타입].py", line 3, in <module>
    Values[0] = '가압1'
    ~~~~~^~
TypeError: 'tuple' object does not support item assignment
```

> 튜플은 값을 할당 할수 가 없어요...

실습

- 정수 리스트 1, 2, 3, 4, 5 를 만들고 첫번째 값과 3번째 값의 합을 나타내 보세요

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Pyt
4
```