
Chapter 1

자료형과 연산자





주석문

- 주석은 프로그램의 실행에는 영향이 없다.
- 코드 작성 후 코드의 이해를 도우는 설명을 첨언하는 기능
- 프로그램 수정 및 변경 시 시간과 노력을 절감할 수 있다.
- 한 줄 주석처리 시에는 '#' 을 사용하여 주석 처리! (색이 변한다.)
- #의 오른쪽 문장만 주석으로 간주됨.

```
12  
13 # 한줄만 주석처리 할 때는 # 기호를 쓰고 뒤에 내용을 작성  
14
```

변수

- 변수란 데이터를 저장하기 위한 공간에 이름을 붙인 것.
- 매번 값을 지정한다면 코드를 작성하거나 수정하기 어렵다.
- 자료를 일관성 있게 사용하고 관리하기 위해 이름(Identifier)를 부여해
- 다른 변수나 자료와 구분해서 사용할 수 있게 함.
- 하나의 변수에는 하나의 값만 저장이 가능.
- 변수의 값은 실행 중에 언제든지 변경 가능.
- 선언되지 않은 변수는 사용이 불가.
- 변수에 이름을 부여하고 값을 저장하면 연산 또한 가능.

```
1 a = 10
2 print(a)
```

10

```
1 int a = 10
```

File "<ipython-input-4-392d75864529>", line 1

```
int a = 10
```

^

SyntaxError: invalid syntax

변수에 값 할당

```
11 abc = 10 + 9
12 print(abc)
13
14 abc = 100
15 print(abc)
16
17 print("-----")
18
19 n1 = 15
20 n2 = 3
21 result = n1 * n2
22 print(result)
23
24 # 선언되지 않은 변수는 사용 불가!
25 # print(number)
26 print("number")
27 # result = n3 + n4
28
29 n1 = n1 + 3
30 print(n1)
31 result = n1 * n2
32 print(result)
33
```

- 변수의 값은 언제든지 바뀔 수 있다. (11~15)
- 변수에 값을 저장하고 연산이 가능하다. (19~22)
- 선언되지 않은 변수는 사용이 불가하다. (24~27)
- 변수를 이용해 또다른 변수를 선언할 수 있다. (29 ~ 32)

변수 이름 규칙

- 변수의 이름은 문자, 숫자, 밑줄(`_`(underscore))문자를 포함
- 변수는 숫자로 시작할 수 없음
- 변수에 사용하는 문자는 대소문자를 구분
- 공백, 문장부호, 특수문자(밑줄 문자만 유일하게 가능) 등은 사용 불가
- 파이썬의 예약어(예: `class`, `def`)는 사용할 수 없음
- 사용 중인 내장 함수나 모듈 이름(예: `id`, `list`, `print`)등은 사용 지양

바른 예		잘 못된 예	
year2000	숫자 포함 가능	2000year	숫자로 시작 못함
Class	대/소문자를 구분함	class	예약어 임
member_name	두 단어이면 _로 연결	member name member.name	.과 공백 등 특수문자를 포함할 수 없음
print_	_를 붙여 사용할 것을 권장함	print	문법 오류는 없지만 권장하지 않음

데이터 타입

- 정수(integer)를 지정하는 키워드는 `int`
- 부동소수점(실수, floating-point)를 지정하는 `float` 또는 `double`
- 논리(boolean)을 지정하는 `boolean`
- 문자와 숫자로 이루어진 문자열(string)을 지정하는 `string`

- 자료형의 확인 - `type()`

1	<code>type(120)</code>
---	------------------------

`int`

1	<code>type(2147483648)</code>
---	-------------------------------

`int`

1	<code>type(3.141592)</code>
---	-----------------------------

`float`

1	<code>type(1.23456e3)</code>
---	------------------------------

`float`

데이터 타입

- 수치형 타입 중 정수형(int) 타입
- 양수, 음수의 정수값을 표현하며 소수점 이하는 표현할 수 없다.
- python에서는 메모리가 허용하는 한 무수히 많은 정수값 저장이 가능하다.
- 2진수, 8진수, 16진수도 저장이 가능하다. 각각 0b, 0o, 0x값을 붙인다.
- 타 진법으로 정수를 출력하려면 bin(), oct(), hex() 함수들을 이용한다.

1	a = 10
2	type(a)

int

데이터 타입 float

- 실수형 데이터 타입, float point.
- 실수형 자료에는 10진수형 소수점 표현 방식과 지수형 표현 방식을 사용할 수 있다.
- 정수형과 정수형의 나눗셈의 결과는 **실수형**으로 출력된다.

1	<code>c = 3.5</code>
2	<code>type(c)</code>

float

1	<code>3/2</code>
---	------------------

1.5

1	<code>type(2/2)</code>
---	------------------------

float



데이터 타입 boolean

- 논리형 데이터 타입은 명제가 참일 경우 **True**, 거짓일 경우 **False**값을 가집니다.
- 관계(비교)연산자는 좌항과 우항을 비교하여 논리값(True,False)값을 도출합니다.

연산자 미리보기

연산자	표현식	연산자의 의미	예제	결과
<	$A < B$	A는 B보다 작으면 True 아니면 False	$5 < 7$	True
>	$A > B$	A는 B보다 크면 True 아니면 False	$5 > 7$	False
<=	$A \leq B$	A는 B보다 작거나 같으면 True 아니면 False	$3 \leq 5$	True
>=	$A \geq B$	A는 B보다 크면 True 아니면 False	$3 \geq 5$	False
==	$A == B$	A는 B와 같으면 True 아니면 False	$3 == 5$	False
!=	$A != B$	A는 B와 같지 않으면 True 아니면 False	$3 != 5$	True

데이터 타입 string

- 문자열(String)은 일련의 문자들을 따옴표 (", ')에 담아 나열한 문자 데이터의 집합형태 입니다.
- 따옴표 안에 어떤 형태의 데이터가 들어가도 모두 문자열로 인식합니다.
- 전 세계의 모든 문자를 저장할 수 있으며, 저장 길이에 제한이 없습니다.

```
1 name = "JinKyoung"  
2 address = '서울시 강남구'  
3 print(name, address)
```

JinKyoung 서울시 강남구

```
1 text = '''이렇게 작성하면  
2 줄 바꿈도 그대로 적용해서  
3 여러 줄의 문자를 작성할 수 있습니다.'''
```

```
1 print(text)
```

이렇게 작성하면
줄 바꿈도 그대로 적용해서
여러 줄의 문자를 작성할 수 있습니다.

탈출문자

- 문자열 내에서 특별한 의미를 갖는 문자들은 **역슬래시(\)**를 이용하여 이스케이프(escape) 문자를 사용

문자	의미
\n	줄 바꿈
\t	탭
\r	리턴(행의 첫 번째 열로 돌아옴)
\0	널(null)
\\	\ 문자 표시
\'	'(홀따옴표) 문자 표시
\"	"(겹따옴표) 문자 표시

```
1 print("Hello\\World")
```

Hello\\World

```
1 print("Hello'World")
```

Hello'World

```
1 print("Hello\"World")
```

Hello"World

출력

- Python의 표준 출력함수 `print()`
- 괄호 안에 출력하고 싶은 변수나 상수, 수식 등을 적으면 콘솔창에 텍스트 형식으로 출력을 실행합니다.

```
10 value = 1234
11 name = "강감찬"
12 print(value)
13 print(name)
14 print(5678)
15 print("홍길동")
16 print(value * 3)
17
```

Console

<terminated>

1234

강감찬

5678

홍길동

3702

- 변수나 상수값, 문자열, 변수를 참조한 연산을 시행하면 문자값으로 출력합니다.

출력

- `print()` 함수의 속성 `separator`
- `separator`는 `sep`이라고 약칭하며 구분자라고 부릅니다.
- `sep` 속성의 기본값은 " " 공백 문자열이 지정되어 있으며 만약 변경하고 싶다면
- `sep` 속성을 직접 작성하여 변경합니다.

```
33 print("-----")
34 print(dog, cat, "좋아~~", sep=" ")
35 print(dog, cat, "좋아~~", sep=",")
36 # 멍멍이=>야옹이=>좋아~~
37 print(dog, cat, "좋아~~", sep="=>")
38 # 멍멍이야옹이좋아~~
39 print(dog, cat, "좋아~~", sep="")
40
41 print("-----")
42
```

```
Console
<terminated> print_ex.py [C:\Users\stephen\AppData\Local\
-----
멍멍이 야옹이 좋아~~
멍멍이,야옹이,좋아~~
멍멍이=>야옹이=>좋아~~
멍멍이야옹이좋아~~
-----
```

출력 print()

- Print() 함수의 속성값 end
- End속성은 데이터 출력 이후 맨 끝에 포함할 문자를 지정하는 용도입니다.
- 기본값으로는 “\n”의 값이 지정되어 있어 줄 개행 명령을 시행합니다.
- 그러므로 Python의 print 함수를 출력하면 항상 기본으로 줄 개 행을 하여 출력합니다.
- Ex) print(“Hello”, “nice to meet you”, sep=“!!”, end=“^^”)

```
1 name = "홍길동"
2 age = 20
```

```
1 print(name, '님의 나이는 ', age, '세입니다.', sep='')
```

홍길동님의 나이는 20세입니다.

출력 format()

인덱스 : format() 함수의 인수 중에서 해당자리에 출력할 인수의 인덱스

자릿수 : 변수의 값을 출력할 최대 자릿수를 지정.

타입 : 출력 형식을 지정. 'd'는 10진 정수, 'f'는 실수(부동소수점), 's'는 문자열을 의미.
숫자 형식 'b'는 2진수, 'o'는 8진수, 'x'는 16진수로 출력.

```
1 print('{}님의 나이는 {}입니다.'.format("홍길동", 20))
```

홍길동님의 나이는 20입니다.

```
1 print('출력 : {2}, {1}, {0}'.format(10, 20, 30))
```

출력 : 30, 20, 10

출력 format()

```
1 a = 12345
```

```
1 print('출력: [{}], [{}:10], [{}:3]'.format(a, a, a))
```

출력: [12345], [12345], [12345]

```
1 ▾ print('출력: [{}:d], [{}:f], [{}:b], [{}:o], [{}:x]'.format(a, a, a, a, a))
2
```

출력: [12345], [12345.000000], [11000000111001], [30071], [3039]

```
1 ▾ print('출력: [{}], [{}:f], [{}:15f], [{}:10.2f], [{}:20.10f]'.format(a, a, a, a, a))
2
```

출력: [12345], [12345.000000], [12345.000000], [12345.00], [12345.000000000000]

d: 10진 정수
f: 실수
b: 2진수
o: 8진수
x: 16진수

출력 format()

문자열의 포맷 코드는 s

{:width.precision} 형식 : {전체자릿수.출력할 문자열의 개수}

```
1 b = 'Hello World'
```

```
1 print('출력: [{}], [{:s}], [{:20}], [{:5}]'.w  
2      .format(b, b, b, b))
```

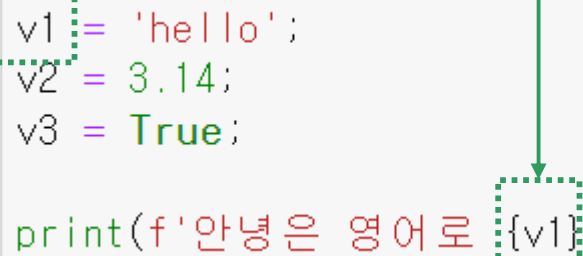
출력: [Hello World], [Hello World], [Hello World], [Hello
World]

```
1 print('출력: [{}], [{:.5}], [{:10.5s}'].format(b, b, b))
```

출력: [Hello World], [Hello], [Hello]

출력 f-string

```
v1 = 'hello';  
v2 = 3.14;  
v3 = True;  
  
print(f'안녕은 영어로 {v1}, 원주율은? {v2}, 진실은 영어로 {v3}');
```





Chapter 1

수고하셨습니다