2주차 : Python의 기초

목 차

1 자료형

1.1 자료형의 개념

- 자료형(Data Type)이란 변수나 상수의 종류를 의미한다.
- 파이썬에서 사용되는 모든 값에는 특정한 타입이 있고 타입에 따라 값들을 조합할 때 동작이 달라진다.
- 파이썬에서 다룰 수 있는 정보의 종류(자료형)는 다양하며
- 그 중에서 자주 사용하는 네 가지는

1	int 형 (정수형: integer)
1.0	float 형 (실수형: floating point)
abc	str 형 (문자열형: string)
True	bool 형 (불형: boolean)

1.2 데이터 형 확인하기

- 데이터의 종류를 확인하려면 바로 **type()** 함수를 사용한다.
- 아래 결과를 보면 <class 'int'>라고 출력되었는데 class는 일단 무시하고
- int만 보면 int는 Integer의 약자로 정수라는 의미이다

```
>>> type(100)
<class 'int'>
```

```
>>> type(3.44)
<class 'float'>
>>> type('파이썬')
<class 'str'>
>>> type(True)
<class 'bool'>
```

2 연산자

2.1 산술 연산자

- 산술 연산자는 둘 이상의 타입 값에 적용할 수 있는 기호로 재정의 연산자(overloaded operator)라고 부른다.

기호	연산자	예시	결과값
+	덧셈	12 + 3.3	15.3
-	뺄셈	6 - 12	-6
*	곱셈	8.8 * 5	44.0
/	나눗셈	11 / 2	5.5
//	몫	7 // 5	1
%	나머지	8.5 % 3.5	1.5
**	제곱	2 ** 6	64

2.2 연산자 결합 법칙

- 연산 식에서 **같은 연산자가 연속해서 나올 경우에**
  연산 순서를 왼쪽부터 취할 것인지 아니면 오른쪽부터 취할 것인지 결정하는 것

❖ **파이선에서의 연산자 결합 법칙**

연산자	결합 법칙
**(제곱)	**오른쪽 → 왼쪽**
-(음수)	왼쪽 → 오른쪽
*(곱), /(나누기), //(몫), %(나머지)	왼쪽 → 오른쪽
+(더하기), -(빼기)	왼쪽 → 오른쪽

2.3 연산자 우선순위

- 표현식에서 하나 이상의 연산자가 섞여 있을 때에는 우선 순위 규칙을 따른다.
- 그리고 괄호는 가장 높은 우선 순위를 가짐으로 괄호 안의 수식이 먼저 계산된다.
- 예를 들어 화씨를 섭씨로 변환하기 위해서는 화씨에서 32를 뺀 후 5/9를 곱하면 섭씨가 된다.

```
>>> 220 – 32 * 5 / 9
202.2222222222223
```

- 파이썬 결과는 섭씨 202.2222222222223도인데 실제로는 104.44444444444444이어야 한다.
- 곱셈과 나눗셈이 뺄셈보다 우선순위가 높아서 생긴 문제이다.
- 아래처럼 하위 표현식의 앞뒤에 **괄호를 넣으면 우선순위를 바꿀 수 있다.**

```
>>> (220 – 32) * 5 / 9
104.44444444444444
```

2.3 연산자 우선순위

연산자 우선순위	연산자	설명
1	( ) [ ] { }	괄호, 리스트, 딕셔너니, 세트 등
2	**	제곱
3	+ − ~	단항 연산자
4	* / % //	산술 연산자
5	+ −	산술 연산자
6	〈〈 〉〉	비트 시프트 연산자
7	&	비트 논리곱
8	^	비트 배타적 논리합
9	|	비트 논리합
10	〈 〉 〉= 〈=	관계 연산자
11	== !=	동등 연산자
12	= %= /= //= −= += *= **=	대입 연산자
13	Not	논리 연산자
14	And	논리 연산자
15	Or	논리 연산자
16	if ~ else	비교식

퀴즈

퀴즈 **다음 빈칸에 들어갈 단어를 채우시오.**

몫 연산자는 (　　), 나머지 연산자는 (　　), 제곱 연산자는 (　　)이다.

퀴즈

퀴즈 다음 빈칸에 들어갈 단어를 채우시오.

몫 연산자는 (//), 나머지 연산자는 (%), 제곱 연산자는 (**)이다.

연산자
**(제곱)
-(음수)
*(곱), /(나누기), //(몫), %(나머지)
+(더하기), -(빼기)

3. 변수

3.0 변수

- 어떤 대상을 칭할 때,
- '전화나 문자를 주고받거나 비디오 영상을 볼 때 사용하는 직사각형의 전자 장치'보다
- '핸드폰'이라는 이름을 사용하면 명확하고 좀 더 쉽게 이해할 수 있다.
- 이와 같이 프로그래밍에서도 원하는 사건이 어떤 순서로 작동할지,
- 그 사건이 어떤 대상에게 일어나야 할지를 지칭할 때 변수를 사용해 사물을 가리킨다.
  - 프로그램에서 만들어내는 모든 사물에는 이름이 있다. 따라서 나중에 그 이름으로 그 사물을 가리킬 수 있으며 이런 이름을 변수(variable)라고 하고 이 변수를 사용해 객체를 가리킨다.
  - 그리고 하나 주의할 점은 수학 시간에 사용하는 변수는 프로그래밍의 변수와는 다르다는 것이다.
    - 수학에서 등식인 'x=1'은 같음을 뜻하지만 프로그램 코드에서의 등호(=)는 대입을 뜻한다.

3.1 할당문

- **변수는 값을 저장하는 메모리 공간이다.**
- **a = 1의 경우 a라는 이름이 붙은 객체는 1이라는 값이 할당되었으며**
- **우리는 이 값에 수학 연산을 적용할 수 있다.**

- **할당문은 변수에 값을 지정하기위해 사용된다.**
  - 할당문의 **왼쪽 : 변수**
  - 할당문의 **오른쪽 : 값**
- **할당문의 예**
  - a = 30 : a라는 변수에 30이라는 값을 할당함
  - b = 50 : b라는 변수에 50이라는 값을 할당함
  - b = a : 변수간의 할당문에서는 실제로 참조를 할당함

3.2 할당되지 않은 변수는 에러

❖ **변수는 사용되기 전에 반드시 할당 되어야 함**

다음 명령문을 순서대로 직접 실행시켜본 후 결과를 비교해보자.

1) number
2) number = 5

```
>>> number
Traceback(most recent call last):
  File "<pyshell#0>", line 1, in <module>
    number
NameError: name 'number' is not defined
>>> number = 5
>>> number
5
>>>
```

3.3 변수명 규칙

❖ 변수 이름을 결정할 때에 고려해야 할 규칙

	변수명 규칙
1	변수의 이름은 **문자, 숫자 그리고 Underscore(_)로만 이루어진다.** 다른 기호를 사용하면 구문 에러(Syntax Error)이다. (예) Money$: 구문 에러, $는 사용할 수 없다.
2	변수명은 **문자 또는 Underscore로만 시작해야 한다. 즉 숫자로 시작 하면 안된다.** (예) 3up, 7brothers : 숫자로 시작했기 때문에, 역시, 구문 에러이다.
3	**파이썬 지정단어 (Keyword, Reserved word)들은 변수명으로 사용할 수 없다.** (지정 단어 목록 참조)
4	파이썬에서는 **대문자와 소문자를 구분한다.** (예) hour 와 Hour는 다른 변수이다.
5	**공백을 포함할 수 없다**

3.4 파이썬 지정 단어 (Keyword, Reserve Word)

- **keyword를 import 하면 파이썬에서 지정한 단어들을 확인 할 수 있음**
  - 파이썬 지정 단어는 변수명으로 사용 할 수 없음에 유의하자.

keyword를 import하여 파이썬 지정 단어를 확인해 보자.

1) keyword.kwlist 명령어는 파이썬 지정 단어를 배열 형태로 나열해 준다.

2) len 함수를 사용해서 파이썬 지정 단어의 개수도 확인해 보자.

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', '__peg_parser__', 'and',
'as', 'assert', 'async', 'await', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except',
'finally', 'for', 'from', 'global', 'if', 'import',
'in', 'is', 'lambda', 'nonlocal', 'not', 'or',
'pass', 'raise', 'return', 'try', 'while', 'with',
'yield']
>>> len(keyword.kwlist)
36
```

3.3 변수명 규칙

❖ **변수변수명은 의미 있게 만들어져야 함**

- **역할에 맞는 변수명**으로 만들어야, 프로그램의 검토, 협업, 공유 시 도움이 됨
- 다음 예제를 통해, **의미 있는 변수 명**의 **차이**를 확인하자.

예제) 3개의 시험 성적이 주어졌을 때, 시험 성적의 총합과 평균을 구하여라.
수학: 26점, 영어: 54점, 역사: 96점

의미 없는 변수 명 예시

```
a = 26
b = 54
c = 96
d = a + b + c
f = d / 3
```

의미 있는 변수 명 예시

```
math = 26
english = 54
history = 96
sum = math + english + history
average = sum / 3
```

퀴즈

퀴즈 아래의 '변수명'이 올바른지 ○/×로 표현해 보자.

① True (×)

② 3apples (×)

③ elif (×)

④ new_score (○)

⑤ Brother (○)

퀴즈

퀴즈 아래의 '변수명'이 올바른지 ○/×로 표현해 보자.

① True (×)
② 3apples (×)
③ elif (×)
④ new_score (○)
⑤ Brother (○)

3.5 여러 변수에 값 할당(Multiple Assignments)

❖ 여러 변수에 값을 할당 시 변수와 값의 개수가 일치해야 함

- 여러 변수에 값 할당 연습

다음 명령문을 직접 실행시킨 후 결과를 비교해보자.

1) number_1, number_2 = 550
2) number_1, number_2 = 2, 4, 8
3) number_1, number_2 = 7, 14

```
>>> number_1, number_2 = 550
Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
    number_1, number_2 = 550
TypeError: cannot unpack non-iterable int object
>>> number_1, number_2 = 2,4,8
Traceback (most recent call last):
  File "<pyshell#19>", line 1, in <module>
    number_1, number_2 = 2,4,8
ValueError: too many values to unpack (expected 2)
>>> number_1, number_2 = 7,14
>>> number_1
7
>>> number_2
14
>>>
```

에러

에러

퀴즈

퀴즈 다음 중 오류가 발생하는 것은 무엇인가?

① h1 = 100

② h1, h2 = 100, 200

③ h1, h2, h3 = 100, 200, 300

④ h1, h2 = 100

퀴즈

퀴즈 다음 중 오류가 발생하는 것은 무엇인가?

① h1 = 100

② h1, h2 = 100, 200

③ h1, h2, h3 = 100, 200, 300

④ h1, h2 = 100

3.6 변수에 변수 할당

- **변수에 변수를 할당하여 사용할 수 있다.**

✓ 단, 할당문의 오른쪽에 변수가 올 때는 반드시 먼저 값을 할당 받은 후 할당문의 오른쪽에 변수를 사용해야 한다.

- **다음 명령문을 직접 실행시켜본 후 결과를 비교해 보자.**

```
number_3 = number_4
number_4 = 550
number_3 = number_4
```

에러, number_4에 값 없음

```
>>> number_3 = number_4
Traceback (most recent call last):
 File "<pyshell#2>", line 1, in <module>
        number_3 = number_4
NameError: name 'number_4' is not defined
>>> number_4 = 550
>>> number_3 = number_4
>>> number_4
550
```

3.6 복합 대입 연산자 +=, -=, *=, /=, //= , %=

- 복합 대입 연산자는 연산과 할당을 동시에 표현하는 것으로
- '+=, -=, *=, /=, //=, %=' 등이 있다.
- 복합 대입 연산자의 의미는 다음과 같다.

복합 대입 연산자	의미
a += 1	a = a + 1
a -= 1	a = a – 1
a *= 1	a = a * 1
a /= 1	a = a / 1
a //= 1	a = a // 1
a %= 1	a = a % 1
a **= 1	a = a ** 1

4. 입력문과 출력문을 위한 함수

4.1 입력문을 위한 함수

- Python에서 자주 사용되는 입력 함수는 2가지 이다.

변수이름=input()	사용자로부터 입력을 받는다.
변수이름=input('문자열')	'문자열'에 해당하는 내용을 출력 후 사용자로부터 입력을 받는다.

- input()함수는 모든 것을 문자열로 입력받으며,
- 아래의 코드는 사용자에게 이름을 입력받아 출력하는 코드를 의미한다.

```
>>> name = input()
Gildong
>>> name
'Gildong'
>>>
```

```
>>> name = input('What is your first name? ')
What is your first name? Gildong
>>> name
'Gildong'
>>>
```

4.2 input() 함수의 자료형 변환

❖ **input() 함수의 입력 결과는 '문자열'이다.**

- 산술 계산을 하기 위해서는 문자열을 정수 또는 실수 자료형으로 변환하여야 가능
  - int() : 문자열 → 정수
  - float() : 문자열 → 실수

```
>>> a = input("입력:")
입력:50
>>> print("자료형:",type(a))
자료형: <class 'str'>
>>> print(a+100)
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    print(a+100)
TypeError: can only concatenate str (not "int")
to str
```

int() →

```
>>> a = input("입력:")
입력:50
>>> print("자료형:",type(a))
자료형: <class 'str'>
>>> print(int(a)+100)
>>> 150
```

float() →

```
>>> a = input("입력:")
입력:50
>>> print("자료형:",type(a))
자료형: <class 'str'>
>>> print(float(a)+100)
>>> 150.0
```

4.3 출력문을 위한 함수

- 입력을 위한 input() 함수가 있다면, 출력을 위한 print() 함수도 존재한다.
- Python에서 자주 사용되는 출력 함수는 3가지 이다.

print('문자열')	'문자열'을 화면에 출력해준다.
print(변수이름)	변수(변수이름)에 해당하는 값을 화면에 출력해준다.
print('문자열', 변수이름)	'문자열'과 변수(변수이름)에 해당하는 값을 연속해서 화면에 출력해준다.

#print( ) 함수 안의 글자에 ' '(따옴표)가 씌워진 것은 str(문자형)의 type을 의미

4.3 출력문을 위한 함수

```
>>> print('True False')
True False
>>>
```

```
>>> number1 = 36
>>> number2 = 42
>>> sum = number1 + number2
>>> print(sum)
78
>>>
```

```
>>> math = 74
>>> english = 87
>>> science = 80
>>> print('수학, 영어, 과학 점수 : ', math, english, science)
수학, 영어, 과학 점수 : 74 87 80
```

4.4 print() 함수를 이용한 양식문자

❖ 'print( )' 함수를 이용하여 문자열, 변수, 숫자로 이루어진 구조의 내용을 출력할 때 양식 문자를 이용하면 좀 더 단순한 형태의 구조로 표현하고 출력할 수 있다.

양식 문자	표현 내용	비고
%d	정수(십진수)	Decimal(0~9)
%f	실수(소수점)	Floating point number
%g	정수 혹은 실수	소수점의 여부에 따라 정수, 실수 자동 표시
%s	문자열	String
%c	문자	Character
%o	8진수	Octal number(0~7)
%x	16진수	Hexa number(0~9, A~F)

4.4 print() 함수를 이용한 양식문자

❖ 양식문자를 사용하지 않은 예시

```
name = "홍길동"
age = 21
weight = 58.7
print("내 이름은", name, "입니다.")
print("나는 ", age, "살 입니다.")
print("나의 몸무게는", weight, "kg입니다.")
```

❖ 양식문자를 사용한 예시

```
name = "홍길동"
age = 21
weight = 58.7
print("내 이름은 %s입니다."%name)
print("나는 %d살 입니다."%age)
print("나의 몸무게는 %f kg입니다."%weight)
```

```
kor = 90
eng = 80
print("국어는 %d점이고 영어는 %d점입니다."%(kor, eng))
국어는 90점이고 영어는 80점입니다.
```

```
x = 7
y = 8
print("%d와 %d를 곱한 값은 %d점입니다."%(x, y, x*y))
7과 8을 곱한 값은 56입니다.
```

퀴즈

❖ **다음 중 오류가 발생하는 것은 무엇인가?**

다음 빈칸에 들어갈 단어를 채우시오.

① 문자열 또는 실수를 정수로 변환해 주는 함수는 ( ) 이다.

② input( ) 함수는 기본적으로 모든 값을 ( )로 입력받는다.

③ 정수를 표시하는 양식 문자는 ( )이고 실수는 ( )이고 문자열은 ( )이다.

퀴즈

❖ **다음 중 오류가 발생하는 것은 무엇인가?**

퀴즈 **다음 빈칸에 들어갈 단어를 채우시오.**

① 문자열 또는 실수를 정수로 변환해 주는 함수는 (int()) 이다.

② input( ) 함수는 기본적으로 모든 값을 (문자)로 입력받는다.

③ 정수를 표시하는 양식 문자는 (%d)이고 실수는 (%f)이고 문자열은 (%s)이다.

감사합니다.