

python_기초반_2주차_수업자료

1. 숫자(number)

→ 숫자는 정수와 부동소수점으로 나뉜다.

- 정수(integer): 소수점이 없는 숫자 ex) 10, 273, 52
- 부동소수점(floating point): 소수점이 있는 숫자 ex) 10.1, 52.273, 52.0

** 주의) 숫자 52는 '정수'이고, 숫자 52.0은 '부동소수점'이다.

→ type() 함수

- type() 함수를 사용해 해당 숫자가 정수인지, 부동소수점인지 알 수 있다. 또한 변수에 저장된 숫자가 정수인지, 부동소수점인지를 알고 싶을 때도 type() 함수를 사용한다.

```
print(type(2), type(2.0))
```



```
<class 'int'> <class 'float'>
```

2. 연산

연산자 종류	의미	예시
+	덧셈 연산	print(4+3) 출력: 7
-	뺄셈 연산	print(4-3) 출력: 1
*(별표)	곱셈 연산	print(4*3) 출력: 12
/	나눗셈 연산 (a/b: a를 b로 나눔)	print(4/3) 출력: 1.333333
//	몫 연산 (a//b: a를 b로 나눈 몫)	print(4//3) 출력: 1
%	나머지 연산 (a%b: a를 b로 나눈 나머지)	print(4%3) 출력: 1
**	거듭제곱, 제곱근 연산 (a**b: a의 b제곱) (a**(1/b): a의 b제곱근)	print(4**3) 출력: 64 print(4**(1/2)) 출력: 2

→ 연산자 우선순위

- 덧셈, 뺄셈보다 곱셈, 나눗셈이 먼저 계산되며, 곱셈, 나눗셈보다 거듭제곱이 먼저 계산된다.
- 우선순위를 알고 있더라도 개발자 세계에서는 어떤 걸 먼저 계산할 지 보여주기 위해 괄호를 명시적으로 표시해 준다.

→ 절댓값 연산

- abs()안에 원하는 수를 집어넣으면 해당 숫자의 절댓값을 출력해준다.

ex) 보충자료 문제 풀어보기

3. 정수끼리의 연산 & 부동소수점끼리의 연산 & 정수-부동소수점 간의 연산

→ 정수끼리의 연산

- 덧셈, 뺄셈, 곱셈, 몫, 나머지, 거듭제곱, 제곱근 연산 결과는 정수가 나온다.
- 나눗셈 연산 결과만 부동소수점이 나온다.

→ 부동소수점끼리의 연산

- 모든 연산 결과 부동소수점이 나온다.

→ 정수와 부동소수점 간의 연산

- 모든 연산 결과 부동소수점이 나온다.

```
a = 2.0
b = 4.0
c = 6
d = 8

print(type(a+b), type(a+c), type(c+d), type(c/d))
b, c, d = (c/abs(a-p))*_c/abs(a+c)*_c/abs(c+q)*_c/abs(c,q))

<class 'float'> <class 'float'> <class 'int'> <class 'float'>
```

4. 변수 개념 심화

→ **정의(define), 선언** - 변수를 만들

→ **할당(assign)** - 변수에 값을 넣음

→ **참조(reference)** - 변수 안에 있는 값을 사용

ex) a = 3 (a라는 상자를 만들 - '정의', 상자에 3을 넣음 - '할당')

ex) a = a + 3 (a의 내용물에 3을 더한 결과를 a라는 상자에 넣음 - a의 내용물을 사용하였으므로 '참조')

ex) c = b (b의 내용물을 c라는 상자 안에 넣음 - 즉, c 상자 안에 무엇이 들어있든 간에 c 내용물은 b의 내용물로 바뀜)

5. 복합대입연산자

* 위 ex) a = a + 3에서 a라는 똑 같은 변수가 두 번 이상 쓰이는 데에 불편함을 느껴 코드를 더 줄이기 위해 만든 일종의 '코딩 트릭' 연산자

→ **+=, -=, *=, /=** 연산자 존재

ex) **a += 3은 a = a+3과 똑같은 의미**를 가진다. (복합대입연산자 여러 번 사용해서 익숙해지기! 보충자료 문제 연습)

→ +=의 경우 문자열에서 기존 문자열에 새로운 문자열을 더할 때도 쓰일 수 있다.

ex) a = 'hello'

a += '!

print(a) #'hello!' 출력 ('hello' 문자열에 '!'가 첨가됨)

6. input() 함수

→ 코딩에서 **제일 중요하며 꼭 알아야 하는, 기본적인 함수**이다. 내가 **키보드로 원하는 데이터를 입력**하면, 컴퓨터가 해당 데이터를 화면에 표시하거나, 만약에 내가 변수를 선언하였을 경우, 변수에 저장해준다.

ex) `a = input()`이라 코드를 만들면, 내가 키보드로 직접 입력한 데이터를 a라는 상자에 저장해준다.

**** 주의!) input() 함수는 문자열을 반환해준다. 즉, 내가 아무리 키보드로 숫자를 입력했더라도, a라는 상자에 저장되는 모든 데이터는 '문자열'로 저장된다.**

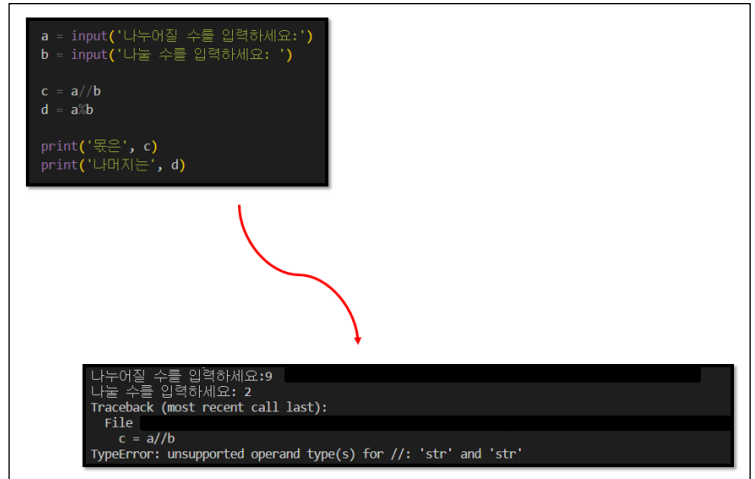
ex) → 오른쪽 코드

Q. 두 수를 입력받고, 나눗셈 결과 출력해주는 프로그램

- 코드를 보면, a와 b의 변수에는 input() 함수의 결과로 문자열이 저장되므로, 몫과 나머지 연산을 할 수 없기에 오류가 나음을 확인할 수 있다.

A. 그렇다면 저장된 문자열 string을 정수형 integer로 바꿀려면 어떻게 해야 할까?

- **type conversion 함수** 사용



7. type conversion

→ 앞서 우리는 정수형 integer, 부동소수점(소수점을 가진 number) float, 그리고 문자열 string 세 개의 data type에 대해서 배웠다. 서로 data type을 바꾸기 위해 type conversion 함수를 사용할 수 있다.

(* 꼭 기억! 문자열은 "또는 '"사이에 내용이 들어간다.)

함수 종류	함수 설명	ex)
str()	()안에 들어간 데이터를 문자열로 바꿔준다.	<code>print(str(3))</code> 결과: '3'
int()	()안에 들어간 데이터를 정수형으로 바꿔준다.	<code>print(int('3'))</code> 결과: 3
float()	()안에 들어간 데이터를 부동소수점형으로 바꿔준다.	<code>print(float(3))</code> 결과: 3.0

→ Q. 그러면 위 나눗셈 몫, 나머지 구하기 프로그램을 잘 작동시키기 위해 코드를 바꿔보면?

A. 오른쪽 그림과 같이 int()함수를 통해 문자열인 a와 b를 정수로 바꿔주었다. 정수로 바꿔주면 몫과 나머지 연산을 시행할 수 있다. 그 결과 올바른 결과 출력!

**** 반드시 보충문제 8문제를 풀어보고 다양한 연산 종류에 익숙해지기.**
추가로 저번 1주차에 배운 **indexing, slicing**과 변수 개념은 꼭 기억하기!

* 참고

→ 보충자료 답: 1. rin / 2. st / 3. 2 / 4. j / 5. 25.0 / 6. e / 7. a / 8. 164

