

파이썬

4강. 변수와 자료형(변수와 자료형)

1. 변수와 자료형

- 파이썬 프로그래밍에서 사용되는 대부분의 자료는 변수와 자료형이라는 용어와 관련이 있다.
- 변수는 자료를 일시적으로 보관하는 역할을 하며, 자료형은 숫자 또는 문자와 같은 자료의 유형을 의미한다.

2. 변수

- 일반적으로 변수는 필요한 자료를 일시적으로 보관하거나 처리 결과를 담을 수 있는 기억장소 역할을 한다. 다시 말해서 변수는 실제 값을 저장하는 메모리(Memory) 이름으로 할당된다. 따라서 메모리에 값이 저장된다는 것은 변수에 값이 저장된다는 의미이다. 하지만 파이썬 변수는 이와는 다소 차이가 있다. 변수에 저장될 값이 먼저 메모리에 객체(Object) 형태로 저장된 이후, 객체의 주소가 변수에 저장된다. 즉 실제값이 저장되지 않고, 실제값이 저장된 주소를 기억하는 참조형 변수가 된다.
- X변수에 3.14159를 초기화 할 경우 메모리에 3.14159가 객체로 저장된 후 메모리 주소 12345가 X변수에 할당된다. 이러한 참조형 변수는 객체의 자료형에 의해서 변수의 자료형이 결정되기 때문에 변수 선언 시 별도의 자료형을 선언하지 않는다. 한편 파이썬 변수는 방대한 자료, 함수 그리고 차트 등 다양한 유형의 객체를 담을 수 있는 막강한 역할을 한다.

2. 변수



2. 변수

- 변수명 작성 규칙
 - 의미를 파악할 수 있는 이름으로 지정하는 것이 좋다.
 - 첫 자는 영문자로 시작한다.
 - 공백이나 특수문자는 사용하지 않는다.
 - 예약어는 사용할 수 없다.
 - 두 번째 단어는 숫자, under bar(_)를 사용할 수 있다.
 - 대문자와 소문자는 서로 다른 변수로 인식한다.(대소문자 구분)
 - 두 단어를 사용하여 변수명을 지정할 경우 두 번째 단어의 첫 자는 대문자로 표기한다.(예: userName, korScore)

2. 변수

chapter02.lecture.step01_variable.py ↵

Python Console ↵

```
var1 = "Hello python"
print(var1)
print( id(var1) )
```

↵

Hello python
67239512 ↵

↵

```
var1 = 100
print(var1)
print( id(var1) )
```

↵

100 ↵
1850632928 ↵

↵

```
var2 = 150.25
print(var2)
print( id(var2) )
```

↵

150.25 ↵
64784064 ↵

↵

```
var3 = True
print(var3)
print( id(var3) )
```

↵

True
1850414288 ↵

2. 변수

- 변수와 메모리 주소(address)
 - var1, var2, var3 변수명에 할당된 값에 대해서 살펴본다. 먼저 var1 변수는 처음 "Hello python" 문자열이 할당(초기화)된 이후 새로운 값 100으로 수정되어 저장되고, var2는 150.25, var3는 True 값이 저장된다.
 - 한편 print() 함수는 변수가 참조하고 있는 메모리 객체를 콘솔에 출력한 후 자동으로 줄바꿈하는 내장함수이다.
 - 또한 id() 함수는 변수가 참조하는 객체의 주소를 콘솔에 출력하는 내장함수이다. id 함수의 출력 결과를 보면 var1 변수는 문자형 객체에서 숫자형 객체로 변경되면서 메모리의 주소도 변경되는 것을 알 수 있다.
 - 즉 같은 변수에 새로운 객체가 들어오면 기존 객체의 내용이 수정되지 않고, 새로운 객체가 생성된다는 사실을 알 수 있다.
 - 메모리에 할당된 객체의 값과 각 변수에 할당될 메모리 주소를 나타내고 있다.

메모리 객체	"Hello python" ↵	100 ↵	150.25 ↵	True ↵
메모리 주소	6723951	▶ 185063292	64784064	

2. 변수

- 식별자(Identifiers) 작성 규칙
 - 식별자는 프로그래밍 작성 과정에서 정의되는 함수명, 클래스명, 모듈명 등을 총칭하는 용어이다.
 - 식별자 작성 규칙은 변수명 작성 규칙과 마찬가지로 공백이나 특수문자는 사용하지 않고, 파이썬 예약어는 사용할 수 없다.

2. 변수

chapter02.lecture.step01_variable.py ↵

Python Console ↵

```
# 예약어 확인 ↵
```

```
import keyword # 모듈 импорт ↵
```

```
python_keyword = keyword.kwlist
```

```
print(python_keyword) ↵
```

```
print( type(python_keyword) )
```

```
print( len(python_keyword) ) ↵
```

```
['False', 'None', 'True', 'a↵  
nd', 'as', 'assert', 'async↵  
, 'await', 'break', 'class↵  
, 'continue', 'def', 'del',↵  
'elif', 'else', 'except', 'f↵  
inally', 'for', 'from', 'glo↵  
bal', 'if', 'import', 'in',↵  
'is', 'lambda', 'nonlocal',↵  
'not', 'or', 'pass', 'raise↵  
, 'return', 'try', 'while',↵  
'with', 'yield'] ↵
```

```
<class 'list'>
```

```
35 ↵
```

2. 변수

- 파이썬 예약어 확인
 - 예약어를 확인하기 위해서 'keyword'라는 모듈(module)을 가져와야한다. 모듈이란 프로그래밍에 서 자주 사용하는 수식이나 기능 등을 파이썬 파일(*.py)로 저장해 놓은 것을 말한다. 이러한 모듈은 파이썬 설치 시 자동으로 만들어진다.
 - 'import keyword' 문장은 keyword.py라는 파일의 내용을 현재 작성 중인 모듈로 가져온다는 의미 이다. 'python_keyword = keyword.kwlist' 문장은 keyword 모듈에 포함된 kwlist 변수의 값을 가져와 python_keyword 변수에 할당한다.
 - 나머지 3개의 print 함수는 가져온 예약어를 출력하고, 변수의 자료형과 크기를 출력하고 있다. 파이썬에서는 하나의 변수에 여러 개의 값을 저장할 경우 'list'라는 자료구조에 배열 형식으로 저장된다. 그리고 배열 원소의 길이는 len()함수에 의해서 확인할 수 있다.

3. 자료 (data)

- 프로그램이 처리할 수 있는 모든 것
- 프로그램은 자료를 처리하기 위한 모든 행위

4. 자료형 (data type)

- 자료를 기능과 역할에 따라 구분한 것
- 문자열 (string) : 메일 제목, 메시지 내용 등
- 숫자 (number) : 물건의 가격, 학생의 성적 등
- 불 (boolean) : 친구의 로그인 상태 등

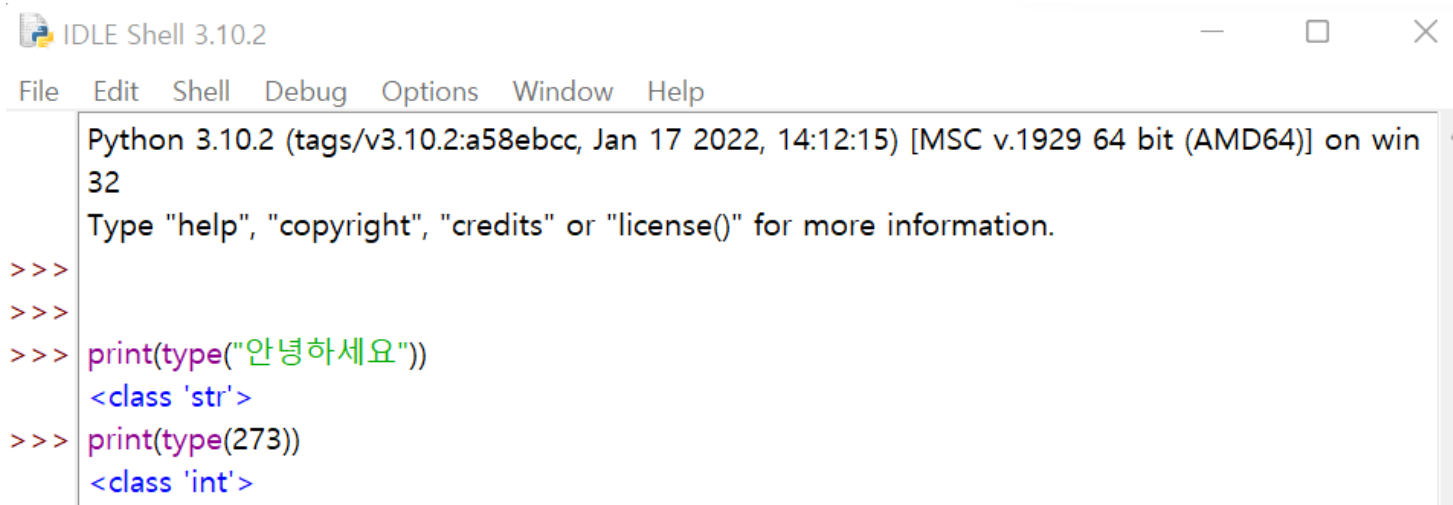
유형(Type)	값(Value)	예(Sample)
숫자형(Numeric)	정수, 실수	125, 125.123
문자형(Character)	문자, 문자열	'홍길동', "홍길동", ""홍길동""
논리형(Logical)	참, 거짓	True, False

4. 자료형 (data type)

- 자료를 알아야 하는 이유
- 자료를 자료형에 맞게 모으고 처리 과정을 거쳐 더 큰 자료형을 점차 만들어 나가면 프로그램이 완성됨
- 가장 기본 단위로서의 자료의 의미와 쓰임새를 아는 것은 매우 중요

4. 자료형 (data type)

- 자료의 형식
 - type() 함수로 확인
 - str : 문자열
 - int : 정수



```
IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>>>
>>> print(type("안녕하세요"))
<class 'str'>
>>> print(type(273))
<class 'int'>
```

4. 자료형 (data type)

- 객체의 자료형에 따라서 처리하는 방법이 달라진다. 예를 들면 숫자형은 통계 처리나 차트를 그리고, 문자형은 자연어 처리, 논리형은 조건식 등에서 사용된다.
- 따라서 객체의 자료형 확인을 빈번하게 발생 한다.
- `type()` 함수는 변수의 자료형을 확인할 때 이용하는 내장함수로 문자열은 `<class 'str'>`
- 정수형은 `<class 'int'>`
- 실수형은 `<class 'float'>`
- 논리형은 `<class 'bool'>` 형식으로 콘솔에 출력된다.
- 기본 자료형 뿐만 아니라 다양한 객체의 자료형도 확인할 수 있다.

4. 자료형 (data type)

chapter02.lecture.step01_variable.py ↵

Python Console ↵

```
var1 = "Hello python" ↵  
print(var1) ↵  
print( type(var1) ) ↵
```

```
Hello python ↵  
<class 'str'> ↵
```

```
var1 = 100 ↵  
print(var1) ↵  
print( type(var1) ) ↵
```

```
100 ↵  
<class 'int'> ↵
```

```
var2 = 150.25 ↵  
print( type(var2) ) ↵
```

```
<class 'float'> ↵
```

```
var3 = True ↵  
print( type(var3) ) ↵
```

```
<class 'bool'> ↵
```


4. 자료형 변환

- 변수가 참조하는 객체의 자료형을 다른 자료형으로 변환하는 방법에 대해서 알아본다

chapter02.lecture.step01_variable.py ↵

Python Console ↵

```
# 실수 -> 정수 ↵
a = int(10.5) ↵
b = int(20.42) ↵
add = a + b ↵
print('add = ', add) ↵
↵
# 정수 -> 실수 ↵
a = float(10) ↵
b = float(20) ↵
add2 = a + b ↵
print('add2 = ', add2) ↵
↵
# 논리형 -> 정수 ↵
print( int(True) ) # 1 ↵
print( int(False) ) # 0 ↵
↵
# 문자형 -> 정수 ↵
st = '10' ↵
print( int(st) ** 2 ) # 제곱 연산 ↵
```

add = 30 ↵

add2 = 30.0 ↵

1 ↵

0 ↵

100 ↵

4. 자료형 변환

- int() 함수는 실수 인자를 대상으로 정수형으로 변환하는 역할을 하고, float() 함수는 정수 인자를 대상으로 실수형으로 변환하는 역할을 한다.
- 또한 int() 함수에 논리형의 인자를 넣어주면 1 또는 0으로 변환된다. True의 비율이나 False의 비율을 계산할 경우 이러한 방식을 이용한다.
- 만약 문자형 자료를 인수로 넣으면 숫자형으로 변환된다.
- 현재 st='10'은 문자형 객체를 갖고 있기 때문에 int() 함수에 의해서 정수형 숫자로 변환되기 때문에 제공(**)에 의해서 100으로 계산이 가능하다.

THANK YOU