
변수와 메모리할당

2주차_01

한 동 대 학 교
김경미 교수

변수(Variables)란 무엇인가?

- **프로그램에서 기억공간이 필요할 때 사용한다**
 - 프로그래머가 붙여준 이름이 있는 기억공간
 - 하드웨어의 메모리를 할당 받아 사용
 - 값을 기억하고 있다
 - 2 는 정수형, 3.14159 는 실수형 값
 - “Hello, World!” 은 문자열
 - 다양한 데이터 형을 표현할 수 있다

언제 변수를 활용하는가?

- **뭔가 기억 시켜 놓을 만한 요인이 생겼을 때**
 - 몇 번 실행 했는지 확인하기 위하여 count 필요할 때
 - 사용자에게 입력 받은 값을 저장해 놓고 싶을 때
 - 일정한 간격으로 숫자를 출력하려고 하는데, 그 간격을 기억해 놓아야 할 때
 - 특정한 조건을 지정하여 저장 할 때
 - 반복문을 사용하는데 반복 횟수를 저장해야 할 때

변수명(Variable Name)

- 문자와 숫자등을 저장하는 공간
- 변수명은 문자로 시작되어야 한다
 - 사용 가능한 변수명
 - myname, name, age, height
 - price_of_tea, student_no, identification_code
 - 나이, 이름, 소속기관
 - 한글 변수명도 변수명으로 사용 가능
 - 하지만, 다른 컴퓨터 언어와 호환되는 경우 한글 변수명 사용 불가
 - 사용 불가능한 변수명
 - 100_name, class, break, False

키워드 (Reserved Word)

- 다음의 이름들은 예약된 단어로 파이썬 문법에서 사용하기 때문에
- 변수명으로 사용될 수 없다

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

변수 사용 예제

```
>>> fred = 100
```

```
>>> print(fred)
```

```
100
```

```
>>> print('fred')
```

```
fred
```

```
>>> fred = 200
```

```
>>> print(fred)
```

```
200
```

변수 사용 예제

```
>>> value = 100
```

```
>>> num_coins = 15
```

```
>>> num_coins * value
```

```
1500
```

```
>>> 15 * 100
```

```
1500
```

```
>>> 15 * value
```

```
1500
```

변수 사용 예제

```
>>> message = "John"
```

```
>>> print(message)
```

```
John
```

```
>>> print("Hello!", "John")
```

```
Hello, John
```

```
>>> score = 78
```

```
>>> print(message , score)
```

```
John 78
```

```
>>> print("message" , "score")
```

```
message score
```

```
>>> print( "abba" * 3)
```

```
abbaabbaabba
```


변수의 데이터 형(DataType)

데이터 형 성격	데이터 형	설명
정수형	int	양의정수, 0, 음의정수
실수형	float	실수로 나타나는 수, -5.1234
문자열형	string	텍스트로 표시하는 글자모음, “한동대”
부울형	boolean	True, False로만 표현 가능
리스트형	list	여러 개의 데이터 들을 모은 것 [1,2,3,4,5] ['apple', 'banana', 'citrus', 'lime'] ['kim', 1, 10, 'park', 5.123]

데이터 형, int

- 값
 - -3, -2, -1, 0, 1, 2, 3, 4, 5, ...
 - 정수 리터럴(literal)은 다음과 같음: 1, 45, 43028030
 - 십표(자릿수 표현)나 마침표(소수점 표현)가 없음
- 연산자
 - +, -, *, /, //, **, 단항 연산자 -
- 원칙
 - int 값에 대한 연산식 결과는 int이다

정수형 사용 예제

```
>>> price = 125
```

```
>>> price * 4
```

```
500
```

```
>>> price = price + 25
```

```
>>> print(price)
```

```
150
```

```
>>> print(price * 10)
```

```
1500
```

```
>>> print(price)
```

```
150
```

정수형 사용 예제

```
>>> price = input("가격을 입력하세요 : ") # input value = 125
```

```
>>> price * 4
```

```
'125125125125'
```

```
>>> price = int(price)
```

```
>>> print(price)
```

```
125
```

```
>>> price * 4
```

```
500
```

데이터 형, float

- 값

- 실수 (근사치)
- 파이썬에서는 “.”을 포함한 숫자를 실수형으로 취급
- 소수점 사용하지 않은 숫자는 정수형으로 취급

- 연산자

- +, -, *, /, **, 단항연산자 -
- 실수형(float)와 정수형(int)은 서로 다른 의미를 가질 수 있음
- 1.0/2.0 의 결과는 0.5

실수형 사용 예제

```
>>> price = 30.5
```

```
>>> price * 4
```

```
122.0
```

```
>>> price = price + 20
```

```
>>> print(price)
```

```
50.5
```

```
>>> print(price * 10)
```

```
505.0
```

```
>>> print(price)
```

```
50.5
```

실수형 사용 예제

```
>>> price = input("가격을 입력하세요 : ") # input value = 30.5
```

```
>>> price * 4
```

```
'30.530.530.530.5'
```

```
>>> price = float(price)
```

```
>>> print(price)
```

```
30.5
```

```
>>> price * 4
```

```
122.0
```

데이터 형, string

- 문자열(String) 형
 - 따옴표 안의 문자열의 나열
 - 이중 따옴표: "Hello World!"
 - 따옴표: 'Hello World!'
 - 연산자: +
 - 연결(Concatenation)은 문자열에 대해서만 적용
 - "ab" + "cd"의 결과는 "abcd"

데이터 형, string

- 문자열(String)은 인덱스화됨(Indexed)
 - $s = \text{'abcd'}$
 - $s[0]$ 은 'a', $s[2]$ 은 'c'
 - $s1 = s[0] + s[3]$
 - $s1$ 에는 'ad' 가 기억된다
 - 부분 문자열 추출 가능
 - $s[1:] \rightarrow \text{'bcd'}$
 - $s[:2] \rightarrow \text{'ab'}$
 - $s[:3] \rightarrow \text{'abc'}$
 - $s[1:3] \rightarrow \text{'bc'}$

문자열 사용 예제

```
>>> s1 = 'The Brave'
```

```
>>> s1
```

```
'The Brave'
```

```
>>> s2 = 'The Beauty'
```

```
>>> s3 = s1 + 'gets' + s2
```

```
>>> s3
```

```
'The Brave gets The Beauty'
```

```
>>> s3[4 : 9]
```

```
'Brave'
```

데이터 형, bool

- **boolean 또는 bool 형**
 - True 와 False로 결과가 나타난다
- **연산자**
 - not, and, or
- **정수형 또는 실수형 값들 비교할 때 사용**
 - $i < j, i \leq j, i \geq j, i > j$
 - $i == j, i != j$

데이터 형, list

- 여러 개의 데이터들을 모은 것

- 집합적인 형태
- Index로 각 요소의 변수명을 지칭

```
>>> fruits = [ 'apple', 'banana', 'blueberry', 'lime' ]
```

```
>>> fruits[2]
```

```
'blueberry'
```

```
>>> numbers = [ 1,2,4,7,11,16,22,29 ]
```

```
>>> print(numbers)
```

```
[ 1, 2, 4, 7, 11, 16, 22, 29 ]
```

요약

- 변수의 개념 이해
- 변수는 언제 활용하는가?
 - 뭔가 기억 시켜 놓을 만한 요인이 생겼을 때
- 변수의 데이터형
 - 정수형
 - 실수형
 - 문자형
 - 부울형
 - 리스트형

감사합니다

2주차_01 변수와 메모리할당