

응용 프로그래밍

python

수업내용

- 파이썬 변수와 기본 자료형
- 사용자 입력과 출력
- 연산자

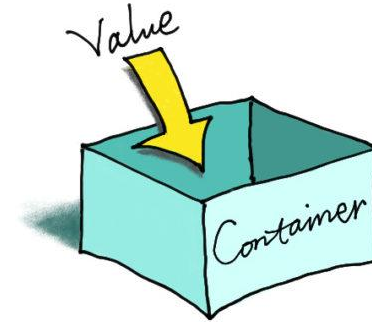
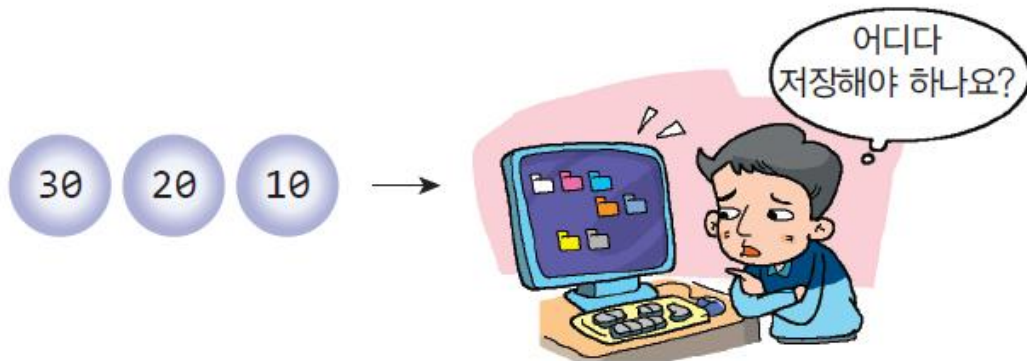
python

파이썬 변수와 기본 자료형

- 파이썬 변수
- 기본 자료형

변수(variable)?

- 변수: 데이터를 저장하기 위한 이름이 부여된 메모리 공간이다.
 - ◆ 각각의 변수는 '식별자'라는 고유의 이름을 가져야 한다.
 - ◆ 변수는 나중에 프로그램에서 변경할 수 있는 데이터를 담는 컨테이너이다.



변수들 : 선언이 필요하지 않음

- 파이썬은 변수를 사용하기 전에 **정의나 선언이 필요하지 않다**.
 - ◆ 변수를 만들고자 한다면, 변수에 값을 할당하고 변수를 사용하면 된다.
 - ◆ 변수에 값을 할당하기 위해서 단일 등호('=')를 사용한다.

```
>>> website = 'www.gachon.ac.kr'
```

```
>>> x = 100
```

```
>>> y = 10.68
```

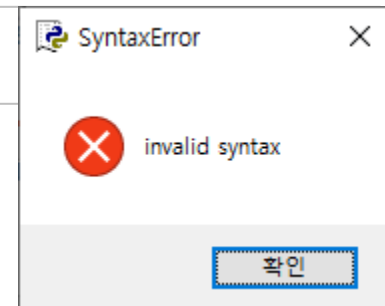
※ 파이썬에서 = 기호는 “변수에 값을 저장하라”는 의미이다.

변수명 작성 규칙

- 변수 이름은 길이에 관계없이 영문자 대/소문자(A-Z, a-z), 숫자(0-9), 그리고 밑줄 문자(_)를 조합해서 만들 수 있다.
- 변수 이름에 숫자가 포함될 수 있지만, 변수 이름의 첫번째 문자로 숫자가 사용될 수는 없다.

count, count, _count_, number3, has_W2,

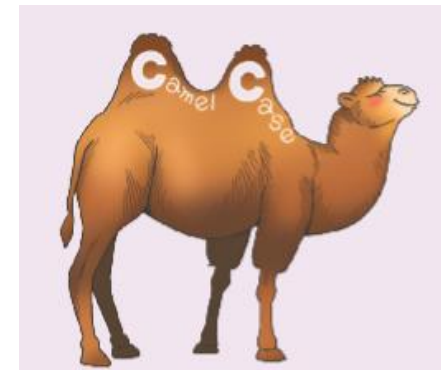
nameCount, ~~1stName~~, _count1, total



변수명 작성 규칙 중 대/소문자 구별에 관한 내용

- 변수 이름은 대/소문자를 구별한다.
 - ◆ 예: `age`, `Age` 그리고 `AGE` 는 서로 다른 3가지 변수이다.
- 변수명을 작성하려면 camelCase 표기법을 사용하세요.
 - ◆ **camelCase 표기법**은 변수의 첫 글자는 **소문자**로 시작하고 나머지 단어의 첫 글자는 **대문자**로 적는 방법이다.

`myName, myAddress, myAge, ...`



낙타 소문자 표기법(camelCase 표기법): 낙타의 등 모양처럼 중간에 획이 난 것 같이 글자의 중간에 대문자가 나오게 표기하는 방법이다.

Example

- $x+y$ 연산을 하여 변수 sum에 저장하고 sum의 값을 출력해 보세요.

```
>>> x = 8760
```

```
>>> y = 4987
```

```
>>> sum = x + y
```

```
>>> print(sum)
```

```
13747
```


- 파이썬은 변수 여러 개를 한번에 만들 수도 있다.
 - ◆ 변수명1, 변수명2 = 값1, 값2 형식으로 변수를 ,(coma)로 구분한 뒤 각 변수에 할당될 값을 지정해주면 된다.
 - ◆ 변수와 값의 개수는 동일하게 맞춰주어야 하며 나열된 순서대로 값이 할당된다.
 - ◆ 만약 변수와 값의 개수가 맞지 않으면 에러가 발생한다.

```
>>> a, b = 27, 419
```

```
>>> a
```

```
27
```

```
>>> b
```

```
419
```

```
>>> x, y, z = 10, 20
```

Traceback (most recent call last):

File "<pyshell#5>", line 1, in <module>

x, y, z = 10, 20

ValueError: not enough values to unpack
(expected 3, got 2)

- 변수 여러 개를 만들 때 값이 모두 같아도 된다면 다음과 같은 방식을 사용할 수도 있다.
 - ◆ 변수1 = 변수2 = 변수3 = 값 형식으로 변수 여러 개를 =로 연결하고 마지막에 값을 할당해주면 같은 값을 가진 변수 3개가 만들어진다.

```
>>> x = y = z = 999
```

```
>>> x
```

```
999
```

```
>>> y
```

```
999
```

```
>>> z
```

```
999
```

Exercise 1

- 변수를 사용하여 두 변수의 값을 바꾸려면 어떻게 해야 할까요?

```
>>> x = 5
```

```
>>> y = 10
```

```
>>> print(x, y)
```

```
5 10
```



```
>>> print(x, y)
```

```
10 5
```

Exercise 1: Solution

```
>>> x = 5
```

```
>>> y = 10
```

```
>>> print(x, y)
```

```
5 10
```

```
>>> x, y = y, x
```

```
>>> print(x, y)
```

```
10 5
```

■ 파이썬에서 지원하는 숫자 형식 데이터

자료형	설명	예	파이썬 사용 예
정수형 (integer)	음수, 0, 양수	1, 2, 3, 100, -9, -99, ...	data = 99
실수형 (floating point)	소수점이 포함된 실수	10.2, -9.37, 0.12, ...	data = 99.12
복소수 (complex)	허수를 표현하는 i와 정수 및 실수가 합쳐진 숫자 파이썬에서는 i를 사용하지 않고 j를 사용	2+4j, 1.4-6.91j, 0+ 31.3j, ...	data = 2 + 3j

Example

```
>>> x = 1234567890123456789
```

```
>>> print(x)
```

```
1234567890123456789
```

```
>>> y = 0.1234567890123456789
```

```
>>> print(y)
```

```
0.12345678901234568
```

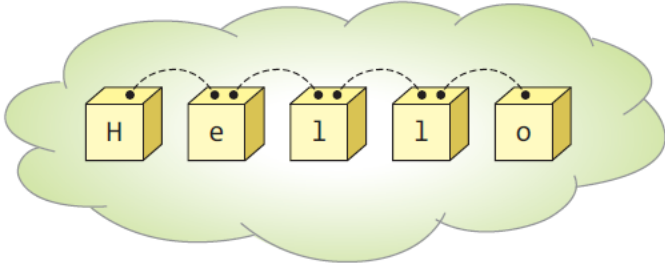
```
>>> z = 1+2j
```

```
>>> print(z)
```

```
(1+2j)
```

문자열(string)

- 문자열(string)은 문자들의 나열이다.



- 문자열은 작은 따옴표(')나 큰 따옴표(")로 정의된다.
 - ◆ 'python' 과 "python" 은 같은 문자열이다.

```
>>> 'python'
```

```
'python'
```

```
>>> "python"
```

```
'python'
```

문자열(string)

- 문자열에 작은 따옴표를 포함시키려면 문자열을 큰 따옴표로 둘러싸면 된다.
 - ◆ 예: it's python → **"it's python"**
- 파이썬에는 **문자(character) 자료형이 없다.**
 - ◆ 단일 문자를 나타내기 위해서는 길이가 1인 문자열을 사용한다.
 - ◆ 예: **'P'** , **"P"**
- 문자열도 변수에 저장될 수 있다.
 - ◆ 변수에 저장된 문자열은 **print()**를 사용해서 출력할 수 있다.
 - ◆ 또는 변수 이름만 입력하고 엔터키를 눌러도 된다.

※ 파이썬에서는 따옴표(" ")가 없으면 무조건 변수 이름이라고 생각한다. 따라서 변수가 아니라 문자열이라는 것을 표시하기 위해서는 따옴표를 붙여야 한다.

Example

```
>>> mystr = "it's python"
```

```
>>> print(mystr)
```

```
it's python
```

```
>>> name = "김가천"
```

```
>>> address = '성남시 수정구 성남대로 1342'
```

```
>>> print(name)
```

```
김가천
```

```
>>> print(address)
```

```
성남시 수정구 성남대로 1342
```

문자열 연산하기: 더해서 연결하기(+)

- + 연산은 모든 변수가 문자열이라면 문자열 연결하기를 실행한다.
 - ◆ 숫자 + 숫자 = 덧셈
 - ◆ 문자열 + 문자열 = 문자열 연결하기
 - ◆ 숫자 + 문자열 = 오류 발생
- + 연산은 숫자와 문자열에 함께 사용할 수 없다.

Example

```
>>> a=27
>>> b=419
>>> print(a+b) # 숫자+숫자
446
>>> mystr1 = 'Hello'
>>> mystr2 = 'python'
>>> print(mystr1 + ',' + mystr2) # 문자열 + 문자열
Hello,python
```

```
>>> print(a + mystr1) # 숫자 + 문자열
```

Traceback (most recent call last):

File "<pyshell#28>", line 1, in <module>

print(a + mystr1)

TypeError: unsupported operand type(s)
for +: 'int' and 'str'

문자열 연산하기 : 반복 연산 (*)

- * 연산을 사용하면 동일한 문자열을 반복시켜서 새로운 문자열을 생성할 수 있다.

```
>>> message = '파이썬!'
```

```
>>> print(message * 3)
```

```
파이썬!파이썬!파이썬!
```

100과 "100"의 차이

- 100과 "100"은 컴퓨터에서는 아주 다르게 취급한다.
 - ◆ 100 → 정수
 - ◆ "100", '100' → 문자열

```
>>> print(100+200)
```

```
300
```

```
>>> print('100'+ '200')
```

```
100200
```

100+200을 하면 (정수+정수) 형태가 되어서 덧셈이 가능하다.
하지만 "100"+"200"은 문자열과 문자열끼리 합하는 것이기 때문에 그냥 2개의 문자열이 연결된다.

문자열: 인덱싱(indexing)

- 문자열 안에 저장된 문자들은 0부터 시작하는 번호가 매겨져 있다. 이 번호를 이용하여 원하는 문자를 추출할 수 있다.
- 이 번호를 인덱스(index)라고 한다.
- 파이썬은 0부터 숫자를 센다.
- 문자열 안에 저장된 문자들을 인덱싱하기 위해서 대괄호([])를 사용한다.

문자열: 인덱싱(indexing)

- 예: 'Oh my god!' 문자열의 인덱스 분석은 다음과 같다:

문자열	O	h		m	y		G	o	d	!
인덱스 (왼쪽→오른쪽)	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
인덱스 (오른쪽→왼쪽)	[-10]	[-9]	[-8]	[-7]	[-6]	[-5]	[-4]	[-3]	[-2]	[-1]

- ◆ 첫번째 영문자 O의 인덱스는 0으로 시작하고, 마지막 항목인 ! 기호는 인덱스가 9이다.
- ◆ 문자열 뒤에서부터 읽어 오기 위해서는 음수(-)기호를 붙인다.
- ◆ 즉 문자열의 가장 오른쪽 마지막 항목인 !기호가 -1이다.

Example

```
>>> mystr = 'Oh my God! I won the lottery!'
```

```
>>> print(mystr[1])
```

```
h
```

```
>>> print(mystr[0])
```

```
O
```

```
>>> print(mystr[-1])
```

```
!
```

```
>>> print(mystr[-12])
```

```
t
```


문자열 중 일부 추출하기: 슬라이싱(slicing)

- 슬라이싱은 문자열의 인덱스 번호를 이용하여 원하는 문자열을 추출하는 것이다.
- 슬라이싱을 하기 위해 대괄호([]) 사이에 콜론(:)을 사용한다.
 - ♦ **[시작 번호:끝 번호(끝번호에 해당하는 것은 제외)]**

```
>>> mystr = 'Oh my God! I won the lottery!'
```

```
>>> print(mystr[0:4])
```

```
Oh m
```

- ♦ mystr[0:4]를 지정하면 끝 번호 4에 해당하는 y는 포함되지 않기 때문에, mystr[0], mystr[1], mystr[2], mystr[3]에 해당되는 문자열 'Oh m'만 출력된다.

Example

```
>>> mystr = 'Oh my God! I won the lottery!'
```

```
>>> print(mystr[6:10])
```

```
God!
```

```
>>> print(mystr[11:])
```

```
I won the lottery!
```

```
>>> print(mystr[:])
```

```
Oh my God! I won the lottery!
```

`mystr[시작 번호: 끝 번호]`에서 끝 번호를 생략하면 시작 번호부터 그 문자열의 끝까지 출력한다.

`mystr[시작 번호: 끝 번호]`에서 시작 번호와 끝 번호를 생략하면 그 문자열의 처음부터 끝 번호까지 출력한다.

- 파이썬은 하나의 변수에 데이터를 저장하는 대입과정에서 변수에 대한 자료형식이 지정된 후, 다시 동일한 변수에 다른 데이터 객체가 대입되면 변수의 형식이 변환된다.

```
>>> var = 23.561
```

```
>>> print(var)
```

```
23.561
```

```
>>> var = "Now I'm a string"
```

```
>>> print(var)
```

```
Now I'm a string
```

Quiz

- 아래의 파이썬 프로그램 코드에 오류가 있는지 살펴 보세요. 만약, 오류가 있다면 어떤 부분에 오류가 있는지 설명하세요.

```
>>> print('I' + 'have' + 12 + 'Python' + 'books')
```

Quiz: Solution

파이썬에서 '+' 연산자는 "**문자와 문자를 연결**"하는 연산자입니다.
그러므로 문자와 숫자를 연결하는데 사용할 수 없습니다.

Exercise 2

- 다음 파이썬 코드의 실행 결과는 무엇일까요?

```
>>> print('python' + ' ' + 'programming')
>>> print('welcome to python!' * 2)
>>> myStr = 'Python is free and open source.'
>>> print(myStr[10:14], myStr[19:23])
```

Exercise 2: Solution

python programming

Welcome to python!Welcome to python!

free open

python

사용자 입력과 출력

- 사용자 입력, input()
- 출력, print()

- 사용자가 입력한 값을 변수에 저장하려면 어떻게 해야 할까요?
 - ◆ 정답: `input()`을 사용하면 돼요.
- `input()`은 (...)안의 메시지를 화면에 출력하고 사용자의 입력을 기다린다. 사용자가 무엇인가를 입력하면 `input()`이 종료된다.
- `input()`은 입력되는 모든 것을 문자열로 취급한다.

사용자로부터 정수 입력 받기

- 사용자로부터 2개의 정수를 입력 받아 덧셈을 한 후 결과를 출력하려면 어떻게 해야 할까요?
 - ◆ 정답: `input()`은 사용자의 입력을 무조건 문자열 형태로 반환하기 때문에 이를 **정수로 변환**하려면 `input()`의 반환 값을 **`int()`로 감싸야** 한다.
- `int()`는 문자열을 정수로 변환한다.

```
x = int(input('첫번째 정수를 입력하세요:'))
```

메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 입력 받는다.

사용자가 입력한 문자열을 숫자로 변환한다.

Example

```
#ex2-1.py
```

```
x=int(input('첫번째 수를 입력하세요:'))  
y=int(input('두번째 수를 입력하세요:'))  
sum=x+y  
print('sum=', sum)
```

```
첫번째 수를 입력하세요:8760  
두번째 수를 입력하세요:4597  
sum= 13357
```

생성자	설명	선언 형태
int()	다른 타입의 값을 정수형(int)로 변환한다.	<pre> >>> int(3.5) 3 >>> int(-9.51) -9 >>> int('529') 529 </pre>
float()	다른 타입의 값을 실수형(float)로 변환한다.	<pre> >>> float(7) 7.0 >>> float(-1) -1.0 >>> float('3.141592') 3.141592 </pre>

Exercise 3

- 사용자로부터 3개의 숫자를 입력 받아서 평균을 계산하고 결과를 출력하세요.

첫번째 수를 입력하세요:245

두번째 수를 입력하세요:893

세번째 수를 입력하세요:359

세 수의 평균은 499.0

Exercise 3: Solution

```
#ex2-2.py
```

```
x=int(input('첫번째 수를 입력하세요:'))
```

```
y=int(input('두번째 수를 입력하세요:'))
```

```
z=int(input('세번째 수를 입력하세요:'))
```

```
avg=(x+y+z)/3
```

```
print('세 수의 평균은', avg)
```

출력: print()

- print()는 (...)안에 주어진 것들을 출력하는 기능을 한다.
 - ◆ print()를 사용하여 변수의 값을 출력한다.
- print() 안에 출력하고 싶은 변수들과 문자열을 콤마(,)로 분리하여 나열하면 된다.

```
x=int(input('첫번째 수를 입력하세요:'))  
y=int(input('두번째 수를 입력하세요:'))  
sum=x+y  
print(x, '과', 'y', '의 합은', sum, '이다.')
```

첫번째 수를 입력하세요:56

두번째 수를 입력하세요:75

56과 75의 합은 131이다.

.format()을 사용한 출력

- 파이썬에서는 문자열을 만들 때 간단한 문자열 포매팅(string formatting)을 제공한다.
- 문자열 포매팅은 **중괄호({ })**안에 포매팅을 지정하고 **.format()**에 값을 넣는다.

```
print('{ }'.format(값))
```


Example

```
#ex2-3.py
```

```
x=int(input('첫번째 수를 입력하세요:'))
```

```
y=int(input('두번째 수를 입력하세요:'))
```

```
sum=x+y
```

```
# print(x, '과', y, '의 합은', sum, '이다.')과 문자열 포매팅은  
동일한 기능을 한다.
```

```
print('{}과 {}의 합은 {}이다.'.format(x,y,sum))
```

첫번째 수를 입력하세요:56

두번째 수를 입력하세요:75

56과 75의 합은 131이다.

Exercise 4

- 사용자로부터 이름과 나이를 입력 받아서 사용자가 100살이 되는 연도를 출력하세요.

이름을 입력하세요: 김가천
나이를 입력하세요: 21
김가천씨는 2098년에 100살이네요!

Exercise 4: Solution

```
#ex2-4.py
```

```
name = input('이름을 입력하세요: ')\nage = int(input('나이를 입력하세요: '))
```

```
year = 2020-age+100
```

```
print('{}씨는 {}년에 100살이네요!' .format(name, year))
```

python

연산자

- 산술 연산자
- 할당 연산자

기본 연산자	예제	결과
+ (더하기)	>>> 47 + 15	62
- (빼기)	>>> 47 - 15	32
* (곱하기)	>>> 47 * 15	705
/ (나누기)	>>> 47 / 15	3.1333333333333333
// (나눗셈의 몫을 정수로 계산)	>>> 47 // 15	3
% (나머지)	>>> 47 % 15	2
** (제곱수)	>>> 47 ** 3	103823

Example

- 초단위로 시간을 입력 받아서 몇 분 몇 초인지 계산해 보세요.

```
#ex2-5.py
```

```
time = int(input('초단위로 시간을 입력하세요: '))  
minutes = time // 60  
seconds = time % 60  
print('{}초는 {}분 {}초입니다.'.format(time, minutes, seconds))
```

초단위로 시간을 입력하세요: 230

230초는 3분 50초입니다.

기본 연산자	예제	의미
=	$x = 5$	$x = 5$
+=	$x += 5$	$x = x + 5$
-=	$x -= 5$	$x = x - 5$
*=	$x *= 5$	$x = x * 5$
/=	$x /= 5$	$x = x / 5$
%=	$x \% = 5$	$x = x \% 5$
//=	$x //= 5$	$x = x // 5$
**=	$x ** = 5$	$x = x ** 5$

Example

```
>>> x=1000  
>>> print(x)  
1000  
>>> x+=173  
>>> print(x)  
1173  
>>> x**=2  
>>> print(x)  
1375929
```


우선 순위	연산자
1	()
2	**
3	*, /, %, //
4	+, -
5	<, <=, >, >=
6	==, !=
7	and, or, not

Exercise 5

- 움직이는 물체의 운동 에너지를 계산하세요.
 - ◆ 물체의 운동 에너지 = $\frac{1}{2} * \text{무게} * (\text{속도})^2$
 - ◆ 사용자로부터 물체의 무게와 속도를 실수형으로 입력 받으세요.

물체의 무게를 입력하세요(킬로그램): 12.75

물체의 속도를 입력하세요(미터/초): 40

물체는 10200.0의 에너지를 가지고 있다.

Exercise 5: Solution

```
#ex2-6.py
```

```
mass = float(input('물체의 무게를 입력하세요(킬로그램): '))  
velocity = float(input('물체의 속도를 입력하세요(미터/초): '))  
  
energy = 0.5 * mass * velocity **2  
print('물체는 {}의 에너지를 가지고 있다.'.format(energy))
```

Q&A