

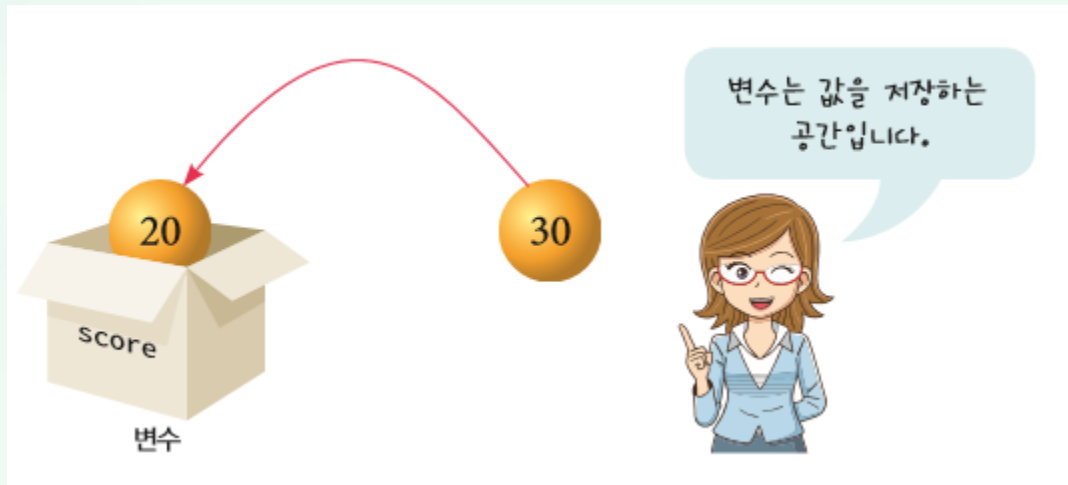
제2장 변수와 계산

1. 변수(Variable)란?

1) 변수의 의미

- 변수(variable)는 값을 저장하는 공간이다. 변수는 값을 저장하는 상자로 생각할 수 있다.
- 변수에 값을 저장했다가 필요할 때 꺼내 쓸 수 있다.
- 변수에 있는 값은 언제든지 다른 값으로 바뀔 수 있다.

예) 파이썬으로 게임을 제작한다면 사용자의 점수는 게임을 하는 도중에 변경될 것이다. 10점짜리 아이템을 획득하였다면 점수가 10점 증가되어야 한다. 이것은 이전의 점수가 어딘가에 저장되어 있어야 한다는 것을 의미한다. 이 때 변수를 사용해야 하는 것이다.



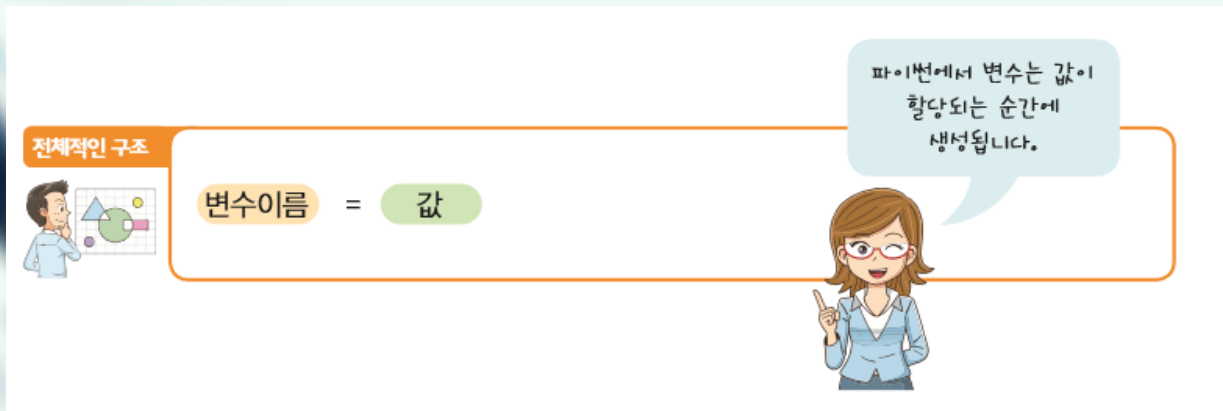
1. 변수(Variable)란?

2) 변수와 메모리

- 컴퓨터는 여러가지 값들을 저장하기 위한 메모리를 가지고 있으며 변수는 바로 이 메모리에 생성된다.

3) 변수 생성 과정

- 변수의 이름을 정하고 등호(=)을 사용하여 값을 저장하면 된다.
- 변수에 값을 저장하는 문장을 할당문(assignment statement) 또는 대입문이라고 칭한다.



1. 변수(Variable)란?

4) 변수의 사용 예

- 게임에서 사용자의 기본 점수가 20점이라고 한다면 아래와 같이 코딩을 하면 될 것이다.

```
>>> score = 20
```

- 위의 문장은 파이썬에게 변수 score를 생성하고 20이라는 값을 저장하라는 것이다.
- = 연산자는 같다는 의미가 아니라 변수에 값을 저장한다는 의미이다.(중요)
- 변수의 값을 출력 하려면 아래와 같이 변수 이름만 입력하고 엔터키를 누르거나, print()함수를 이용하면 된다.

```
>>> score
20
>>> print(score)
20
```

- 생성된 변수에는 얼마든지 다른 값을 저장할 수 있다.(덮어쓰는 것이다)

```
>>> score = 20
>>> score = 30
>>> socre
30
```

1. 변수(Variable)란?

4) 변수의 사용 예

- 변수에는 다른 변수의 값도 저장할 수 있다.
- 일반적으로는 변수에 어떤 수식의 값도 저장할 수가 있다.
- 사각형의 가로 길이와 세로 길이를 나타내는 변수를 만들고 사각형의 면적을 계산하는 코드는 아래와 같이 코딩하면 될 것이다.

```
>>> width = 10
>>> height = 20
>>> area = width * height
>>> print(area)
200
```

5) 변수에 문자열 저장

- 파이썬의 변수에는 정수 뿐만 아니라 문자열도 저장할 수 있다.

```
>>> str = '안녕하세요?'
>>> print(str)
안녕하세요?
```

문자열은 '...' (싱글 코테이션)이나
"..." (더블 코테이션) 안에 작성해야
한다.

1. 변수(Variable)란?

6) 변수에 실수 저장

- 파이썬의 변수에는 실수도 저장할 수 있다.

```
>>> pi = 3.141592
>>> print(pi)
3.141592
```

* 주의

- 파이썬에서 값이 할당되지 않은 변수를 사용한다면 오류가 발생한다.

```
>>> print(str)
NameError : name 'str' is not defined
>>>
```

- 파이썬에서는 변수의 값만 변경할 수 있다. 변수 x에 1을 대입할 수 있지만, 1에는 x를 대입할 수 없다.
두 번째 줄에서 1은 변수가 아니기에 오류가 발생한다.

```
>>> x = 1
>>> 1 = x
SyntaxError : can't assign to literal
```

2. 변수와 상수(constant), 주석(comment)

1) 변수가 저장하는 것

- 파이썬에서 변수는 어떤 데이터든 저장할 수 있다. 먼저 수를 저장할 수 있다. 2가지의 종류의 수를 저장할 수 있는데 하나는 정수(integer)이며, 소수점이 없는 수를 의미한다. 파이썬에서는 이것을 int라고 칭한다. 하지만 0.524와 같은 소수점이 있는 수가 필요하면 실수(floating-point)를 사용하면 된다. 이것을 float라고 칭한다. 데이터의 종류를 자료형(data type)이라고 한다. 또 변수는 "Hello"와 같은 문자열을 저장할 수 있다.
- 파이썬에서는 변수에 어떤 자료형의 값이든지 저장할 수 있다. 파이썬에서 자료형은 값과 연관되고 변수와는 연관되지 않는다.

```
>>> value = 3
```

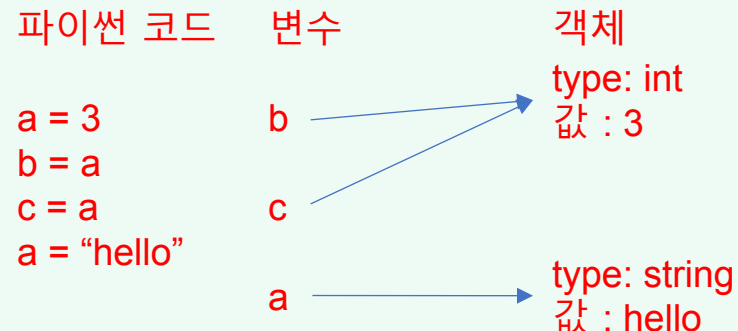
- 동일한 변수 value에 실수값을 저장할 수 있다.

```
>>> value = 10.777
```

아울러, 문자열도 저장할 수 있다.

```
>>> value = "hello"
```

파이썬이 이렇게 어떤 자료형의 데이터든지 저장할 수 있는 것은 파이썬에서는 모든 것이 객체(object)로 되어 있기 때문이다. 변수에 저장되는 것은 실제로 객체의 주소이다. 따라서 객체이기만 하면 뭐든지 변수로 가리킬 수 있는 것이다.



2. 변수와 상수(constant), 주석(comment)

2) 변수의 이름의 규칙

- 의미 없는 이름을 사용하는 것보다는 의미 있는 이름을 사용하는 것이 바람직하다.
- 소문자와 대문자는 서로 다르게 취급된다.
- 변수의 이름은 영문자와 숫자, 밑줄(_)로 이루어진다.
- 변수의 이름 중간에 공백이 들어가면 안 된다. 단어를 구분하려면 밑줄(_)을 사용 한다.



```
value  
_count  
boxVolume  
numberOfPictures  
King3
```



```
2nd_base (x)  
money# (x)
```

개발자들은 변수의 이름을 지을 때, 낙타체를 많이 사용한다. 낙타체란 변수의 첫 글자는 소문자로, 나머지 단어의 첫 글자는 대문자로 적는 방법이다. 예를 들면, myNewCar처럼 첫 'm'은 소문자로, 나머지 단어들의 첫 글자는 대문자로 표기한다.

2. 변수와 상수(constant), 주석(comment)

3) 상수(constant)

- 상수(constant)는 한번 값이 결정되면 절대로 변경되지 않는 변수를 말한다.
- 권장사항은 상수의 이름은 통상 대문자로 작성하는 것이 관례적이다.

```
TAX_RATE = 0.35  
PI = 3.141592  
MAX_SIZE = 100
```

- 아래 소스에서는 왜? 0.35라 적지 아니하고 TAX_RATE으로 코딩을 할까? 이걸 2가지의 이유로 볼 수가 있다.
첫 번째는 TAX_RATE으로 코딩하면 가독성적인 면이 향상되며, 두 번째로는 만약 세율이 바뀌면 상수값인 TAX_RATE의 값만 변경하면 유지보수에 도움이 많이 되는 이유이다.

```
TAX_RATE = 0.35  
tax = 1000 * TAX_RATE  
income = 1000 - tax  
print("현재의 세율은 ", TAX_RATE*100, "%입니다")
```

2. 변수와 상수(constant), 주석(comment)

4) 주석(comment)

- 주석(comment)은 소스 코드에 붙이는 설명글이다.
- 주석은 프로그램이 하는 일을 설명한다.
- 주석은 프로그램의 실행 결과에 영향을 끼치지 아니한다.
- 주석은 반드시 있어야 하는 부분은 아니다.
- 아울러, 컴파일러가 주석은 무시하면 주석에 대한 기계어 코드를 전혀 생성하지 아니한다.
- 파이썬에서는 # 을 적으면 주석이 된다.

```
# 사각형의 가로 길이  
width = 10
```

```
# 사각형의 세로 길이  
height = 20
```

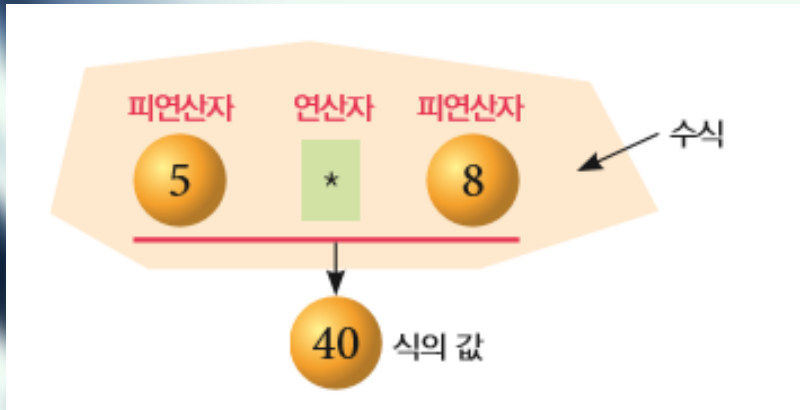
```
# 사각형의 면적 계산  
area = width * height
```

3. 수식과 연산자

1) 수식(expression)

- 수식이란 피연산자들과 연산자의 조합을 의미한다.
- 연산자(operator)는 연산을 나타내는 기호를 의미한다.
- 피연산자(operand)는 연산의 대상이 되는 것을 의미한다.

예) 수식 $(5 * 8)$ 에서 5와 8은 피연산자이며, *는 연산자에 해당한다.



3. 수식과 연산자

2) 산술 연산자

- 산술 연산자는 기본적인 산술 연산인 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지 연산을 실행하는 연산자이다.
- 연산자의 종류는 아래의 그림과 같다.

연산자	기호	사용예	결과값
덧셈	+	$7 + 4$	11
뺄셈	-	$7 - 4$	3
곱셈	*	$7 * 4$	28
나눗셈	//	$7 // 4$	1
나눗셈	/	$7 / 4$	1.75
나머지	%	$7 \% 4$	3

주의할 점은 앞서 실습에서도 확인해본 바 있지만 나눗셈은 항상 실수로 계산된다는 것을 알고 있다. 하여 나눗셈을 하여 정수값을 결과로 얻고자 한다면 //를 사용하도록 한다.



3. 수식과 연산자

3) 지수 계산하기

- 지수(power)를 계산하려면 ** 연산자를 사용한다.
- 예를 들어 3의 3승을 구하기 위한 수식은 아래와 같다.

```
>>> 3 ** 3  
27
```

- 수학에서처럼 지수 연산자는 다른 연산자들보다 높은 우선순위를 가진다.

4) 나머지 계산하기

- 나머지 연산자 %는 생각보다 많이 사용되는 중요한 연산자이다.
- $x \% y$ 는 x 를 y 로 나누어서 남은 나머지를 반환한다.
- 예를 들어 9를 2로 나누면 몫은 4이고 나머지는 1이 된다.
- 나머지 연산은 모듈러스(modulus)연산이라고 불린다.
- % 연산자는 퍼센트와는 전혀 관련이 없다.

```
>>> 9 % 2  
1
```

4. 연산자의 우선순위

1) 연산자 우선순위

- 여러 연산자들이 하나의 수식에 섞여 있는 경우, 연산자의 우선순위에 따라서 연산된다.
- 산술 연산자의 경우는 아래와 같다.

- ① 지수(**)
- ② 곱셈, 나눗셈, 나머지
- ③ 덧셈, 뺄셈

- 일반적인 수식의 경우, 왼쪽에서 오른쪽으로 계산된다.

```
>>> 1 + 2 * 3
7
>>> 4 - 40 - 3
-39
```

- 하지만, 이것저것 우선순위 따질 필요없이 괄호를 사용하는 것이 제일 좋은 방법이다. 그 이유는 괄호를 추가하면 코드가 보기 쉬워지며, 연산자의 우선순위를 생각할 필요도 없다는 것이다.

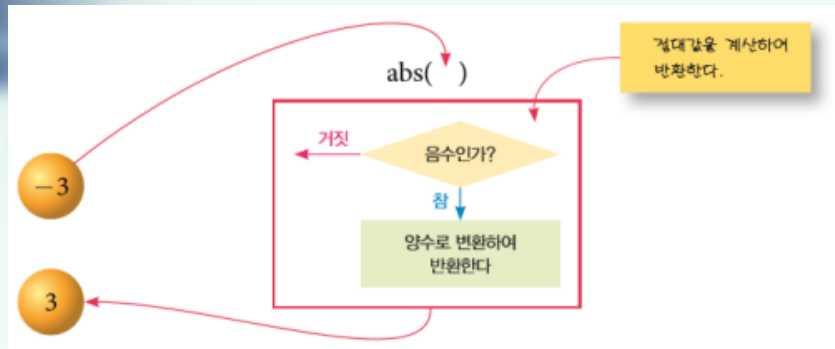
```
>>> 1 + (2 * 3)
7
```

5. 함수(function) 호출

1) 함수 호출

- 함수란 특별한 기능을 담당하는 명령어들의 모임이다.
- 지금까지 print(), input(), int() 함수를 사용해왔다. 파이썬이 기본으로 제공하는 내장 함수는 상당히 많다.
- 계산과 관련된 함수들을 확인하고 넘어가도록 하자.

① abs()함수 : 절대값 계산



② round()함수 : 소수 첫째 자리에서 반올림

```
>>> round(1.278)
1
>>> round(1.888)
2
```

5. 함수(function) 호출

1) 함수 호출

- ③ max()함수 : 매개변수 값들 중 최대값을 반환

```
>>> max(10, 50)  
50
```

- ④ min()함수 : 매개변수 값들 중 최소값을 반환

```
>>> min(10, 50)  
10
```

- ⑤ sqrt() 함수 : 제곱근을 구함

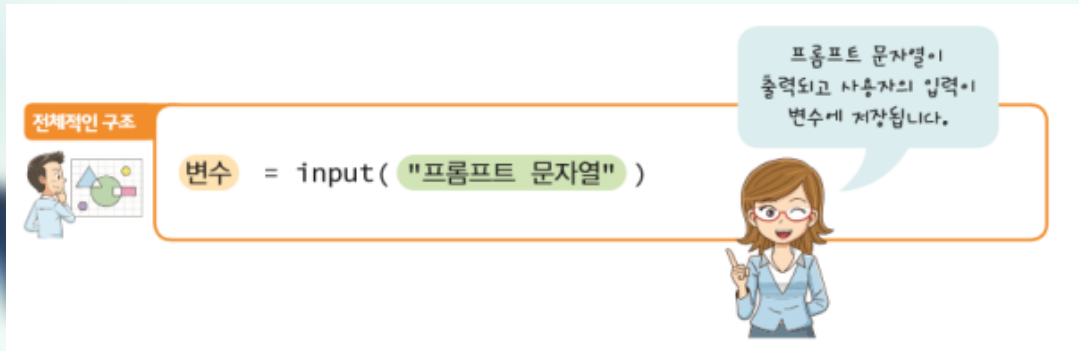
```
>>> from math import *           # 한번만 하면 된다.  
>>> sqrt(4.0)  
2.0
```

sqrt()와 같은 수학적 함수들은 math라고 하는 라이브러리에 저장되어 있다. 파이썬에서는 이것을 모듈(module)이라고 부른다.
math라이브러리에는 sin(), cos(), log() 등등의 다양한 함수가 존재하여 수학적 계산을 쉽게 할 수 있게 해준다.

6. input 함수()

1) input()함수

- 사용자로부터 키보드에서 입력을 받는 함수이다.



- input()이 호출되면 프로그램의 실행은 사용자가 입력할 때까지 잠시 중단된다. 사용자가 입력을 마치고 엔터키를 누르면 프로그램의 실행이 다시 시작된다.
- input()의 프롬프트 문자열은 화면에 출력되며 사용자에게 안내를 하는 용도로 사용되는 것이 바람직하다.

① 문자열 입력

```
name = input("이름이 무엇인가요? ")
print("만나서 반갑습니다. " + name + "씨!")
age = input("나이는요? ")
print("네, 그러면 당신은 이미 " + age + " 살이시군요, " + name + "씨!")
```

6. input 함수()

1) input()함수

```
이름이 무엇인가요? 홍길동  
만나서 반갑습니다. 홍길동씨!  
나이는요? 21  
네, 그러면 당신은 이미 21 살이시군요, 홍길동씨!
```

② 숫자 입력(변경 전)

```
x = input("첫 번째 정수: ")  
y = input("두 번째 정수: ")  
sum = x + y  
print("합은 ", sum)
```

```
첫 번째 정수: 10  
두 번째 정수: 20  
합은 1020 ← 문자열로 간주하여 서로 합침!
```

6. input 함수()

1) input()함수

② 숫자 입력(변경 후)

```
x = int(input("첫 번째 정수: "))  
y = int(input("두 번째 정수: "))  
sum = x + y  
print("합은 ", sum)
```

```
첫 번째 정수: 10  
두 번째 정수: 20  
합은 30
```

앞서 강의에서 설명한 바 있지만 재차 복기를 하도록 하자. input()함수는 사용자가 입력한 값을 문자열로 변수에 저장하기 때문에 10, 20을 문자열로 인식하여 1020이 나오는 것이다. 하여, 사용자로 정수를 입력 받기 위해서는 int()함수 즉, 문자열을 숫자로 바꾸는 함수를 사용하여야지만 올바른 결과를 얻을 수 있다.



감사합니다.