Python 시작

최석재
lingua@naver.com

값 출력하기와 주석 넣기

- 가장 기본적인 함수인 print() 함수의 사용방법을 알아본다
- 숫자, 수식, 문자열 등 다양한 내용을 출력할 수 있다
- # 표시를 하면 주석, 즉 메모를 입력할 수 있다

```
print(2020)                 # 숫자 2020을 출력한다
print(3+2)                  # 수식 3+2의 연산결과를 출력한다
print("안녕하세요~~")        # 문자열을 출력한다
```

변수와 변수 출력

- 변수를 사용하면 기억하기 어려운 값을 쉽게, 그리고 재사용할 수 있다
- 예를 들어, pi 값을 저장하면 다음과 같이 사용할 수 있다

```
pi = 3.14159265359
print(pi * 2)                    # 6.28318530718
```

- 3.14159265359 라는 복잡한 내용대신 'pi'라고 하는 사람이 이해하기 쉬운 이름을 사용하여 접근할 수 있다
- 같은 내용을 간결한 'pi'라고 하는 이름으로 재사용할 수 있다

- 변수 사이에 연산도 가능하고, 수정도 가능하다

```
apple = 10                                  # 한 개의 값을 하나의 변수에 저장
lemon, banana = 20, 50                      # 두 개의 값을 두 개의 변수에 저장
fruit = apple + lemon + banana              # 세 개의 변수가 가지고 있는 내용을 더하여 저장

print("apple:", apple)
print("banana:", banana)
print("fruit:", fruit)

apple = 30                                  # 변수의 값을 수정
print("apple:", apple)
```

변수 이름 짓기

- 변수의 이름을 지을 때는 몇 가지 규칙이 있다

① 변수명이 숫자로 시작할 수는 없다
② 변수명은 띄어 쓸 수 없다
③ 대문자와 소문자는 구분된다

④ 밑줄을 잘 활용한다 (user_id, user_year, user_age, …)
⑤ 숫자를 잘 활용한다 (user1, user2, user3, …)

사용 불가 표현

- !, @, #, $, %, ^, * 와 같은 특수문자는 변수명으로 사용 불가
- 파이썬에서 지정해 놓은 아래의 명령어도 사용 불가

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

기본 연산자

- 파이썬의 기본 연산자로는 아래와 같은 것들이 있다

기호	연산자	의미
x + y	더하기	x와 y를 더한다
x - y	빼기	x에서 y를 뺀다
x * y	곱하기	x와 y를 곱한다
x / y	나누기	x를 y로 나눈다
x ** y	지수승	x의 y 승을 구한다
x % y	나머지	x를 y로 나눈 나머지를 구한다

```
add = 3+25
print("add:", add)

sub = 45 - 13
mul = 9*9

div = 3/2
print("div:", div)

exp = 10 ** 3                    # 10의 3승
print("exp:", exp)

mod = 101 % 10                   # 나머지 연산자
print("mod:", mod)
```

```
add: 28
div: 1.5
div_int: 1
exp: 1000
mod: 1
```

문자열 맛보기

- 값을 작은 따옴표 혹은 큰 따옴표로 묶어주면 문자열이 된다
- 작은/큰 따옴표로 시작했으면 작은/큰 따옴표로 끝나야 한다

```
print('작은 따옴표 사용')
print("큰 따옴표 사용")

city = 'Seoul'
city2 = '서울'
도시 = "한글 도시"                    # 한글로도 변수명을 만들 수 있다

print(city)
print(city2)
print(도시)
```

여러 줄 처리

- 여러 줄로 처리해야 하는 경우를 다루어 본다

```
num1 = 10; num2 = 20; num3 = 30;
print(num1 + num2 + num3)
```

※ 세미콜론을 사용하면 여러 줄의 내용을 한 줄에 작성할 수 있다

```
sum = 1 + 2 + 3 + ₩
    4 + 5 + 6 + ₩
    7 + 8 + 9
print('sum:', sum)
```

※ 줄을 넘겨서 내용을 작성하려면 줄의 끝에 연결 문자인 역슬래시(₩)를 입력한다

```
sum2 = (1 + 2 + 3 +
    4 + 5 + 6 +
    7 + 8 + 9)
print("sum2:", sum2)
```

※ 괄호를 사용하면 연결 문자를 사용하지 않아도 된다

내장 함수 맛보기

- 파이썬은 자주 사용하는 연산을 내장 함수로 만들어 두었다
- 이러한 함수들은 어느 곳에서나 불러서 사용할 수 있다
- 사용자가 직접 만들 수도 있고, 내장된 함수도 있다
- 함수는 메서드(method)라고도 한다

```
mytext = "이것은 샘플 문장입니다!"
mynumbers = (5, 6, 7, 8, 9)

print("length:", len(mytext))              # 입력된 문자열의 길이를 구한다
print("max number:", max(mynumbers))       # 최대값
print("min number:", min(mynumbers))       # 최소값
print("sum:", sum(mynumbers))              # 총합
```

사용자 입력

- 사용자로부터 내용을 입력받을 수 있다

```
print("이름을 입력하세요:")
name = input()                                  # input() 함수로 입력을 받는다

age = input("나이를 입력하세요:")    # input() 함수에서 바로 입력을 받을 수 있다

print("안녕하세요,", age, "살", name, "님")
```

```
이름을 입력하세요:
최길동
나이를 입력하세요:22
안녕하세요, 22 살 최길동 님
```

※ 공백없이 문자열을 연결하는 방법은 추후에 다룬다

내장 모듈 맛보기

- 모듈이란 코드의 묶음이다
- 서로 관련이 있는 변수, 함수 등을 묶어서 특정 작업 처리에 사용한다
- 사용자가 직접 만들 수도 있고, 내장된 모듈도 있다

```
import math                 # 모듈을 사용하려면 import 명령어로 메모리에 올려야 한다

print(math.pi)              # 모듈의 변수 사용
print(math.sqrt(16))        # 모듈의 함수 사용

r = 10                      # 반지름
cir = 2 * math.pi * r       # 원의 둘레 = 2 * pi * r
```

오류 메시지

- 프로그래밍 중 종종 잘못된 코드를 입력하여 오류 메시지가 출력된다
- 당황하지 말고, 찬찬히 보면 어디에서 어떤 문제가 생겼는지 알 수 있다

```
frint(2020)
# print("안녕하세요')
```

※ 하나씩 진행해 본다

```
C:₩Anaconda3₩python.exe "C:/Users/lingu/OneDrive/Python Projects/교육_Python_1_Basic/TEST_11_오류메시지.py"
Traceback (most recent call last):
  File "C:/Users/lingu/OneDrive/Python Projects/교육_Python_1_Basic/TEST_11_오류메시지.py", line 3, in <module>
    frint(2020)
NameError: name 'frint' is not defined
```

※ frint 라는 것은 정의되지 않았다

```
C:₩Anaconda3₩python.exe "C:/Users/lingu/OneDrive/Python Projects/교육_Python_1_Basic/TEST_11_오류메시지.py"
  File "C:/Users/lingu/OneDrive/Python Projects/교육_Python_1_Basic/TEST_11_오류메시지.py", line 4
    print("안녕하세요')
                      ^
SyntaxError: EOL while scanning string literal
```

※ End Of Line에 이르기까지 필요한 문자열을 찾지 못했다

복합 대입 연산자

- 두 개의 연산자를 결합하여 사용하는 경우가 있다

```
x = 5

x = x + 1
print(x)

x += 1
print(x)

x = x * 2
print(x)

x *= 2
print(x)
```

연산자	같은 의미
x += y	x = x + y
x -= y	x = x - y
x *= y	x = x * y
x /= y	x = x / y
x //= y	x = x // y
x %= y	x = x % y
x **= y	x = x ** y

연산자 우선순위

- 연산자 사이에는 우선순위가 있다
- 이를 명확하게 하려면 괄호를 사용한다

```
x = 5 + 1 * 10
print(x)                    # 15

x = (5 + 1) * 10
print(x)                    # 60
```

우선순위	연산자	연산 의미
1	( )	괄호
2	**	지수승
3	*, /, //, %	곱셈, 나눗셈, 나머지 연산
4	+, -	덧셈과 뺄셈
5	=, 복합대입	대입연산자와 복합대입연산자