Python Programming

For Beginner

Operator

For Python 3

Python의 기본 연산자

1. 연산(계산)을 수행하는 도구
2. 연산의 주요 용어
   - 피연산자
   - 연산자
   - 연산

3. 연산의 종류
   - 수학 연산
   - 논리 연산
4. 연산자의 종류
   - 단항 연산자: +/-, ++/--, !
   - 이항 연산자: 대부분의 연산자
   - 삼항 연산자 → [결과(조건 참)] if [조건] else [결과 (조건 거짓)]

주요 연산자

1. 산술 연산자
   - +, -, /, *, %
2. 논리 연산자
   - <, >, <=, >=
3. 식별 연산자
   - is, is not
4. 연산자의 우선 순위
   - 괄호(), 지수, 곱셈과 나눗셈, 덧셈과 뺄셈
5. 괄호의 목적
   - 연산자의 기본 우선 순위 변경
   - 코드의 가독성 및 명확성 향상
   - bool b = a % 3 > c / 4 → bool b = (a % 3) > (c / 4)

```
print("닭을 세어 봅시다")

print("암탉", 25 + 30 / 6)
print("스탉", 100 - 25 * 3 % 4)

print("이제 계란도 세어봅시다")

print(3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6)

print("3 + 2 < 5 - 7는 참인가요?")

print(3 + 2 < 5 - 7)

print("3 + 2는 얼마인가요?", 3 + 2)
print("5 - 7는 얼마인가요?", 5 - 7)

print("아하, 거짓인 이유를 알았네요.")

print("좀 더 해볼까요.")

print("더 큰가요?", 5 > -2)
print("크거나 같은가요?", 5 >= -2)
print("작거나 같은가요?", 5 <= -2)
```

논리 연산

1. 연산자와 피연산자의 결합이 어떤 지점에서 참/거짓인지 판별
2. 논리 기호/연산자
   - and, or, not
   - !=, ==
   - <,>,>=, <=
   - True, False
3. 논리 구문 푸는 단계
   1) 등가 검사(!=, ==)
   2) 괄호 내의 and/or
   3) not 적용
   4) 남은 and/or 결정

```
#비교 연산
x = 10
y = 12
print('x > y is',x>y)
print('x < y is',x<y)
print('x == y is',x==y)
print('x != y is',x!=y)
print('x >= y is',x>=y)
print('x <= y is',x<=y)

#논리 연산
xx = True
yy = False
print('xx and yy is',xx and yy)
print('xx or yy is',xx or yy)
print('not xx is',not xx)

#식별 연산
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]
print(x1 is not y1)
print(x2 is y2)
print(x3 is y3)
```

Standard Input / Output

For Python 3

Print 함수 기본 사용법

1. Python의 표준 출력 함수
2. Print(value1, value2, ...)
3. 출력 값 사이 구분자 지정, sep
   - 기본 값: 공배
4. 출력 마지막 처리 지정, end
   - 기본 값: 줄 바꿈

```
# print 함수 기본 출력
print("하나","둘","셋",1,2,3)
print("하나","둘","셋",1,2,3,sep='-')
print("첫번째 값")
print("두번째 값") # 다른 줄에 출력
print("첫번째 값", end=" ---> ")
print("두번째 값") # 같은 줄에 출력
```

문자열 출력 총 정리

1. 작은 따옴표나 큰 따옴표 사용
   - 작은 따옴표 내에 큰 따옴표
   - 큰 따옴표 내에 작은 따옴표
2. 확장 문자 ₩가 필요한 경우
   - 큰 따옴표 내에 큰 따옴표 출력
   - 작은 따옴표 내에 작은 따옴표 출력
3. 동일 문자열 여러 번 출력
   - “value * 횟수”

```
print('Hello World!!')
print("Hello World!!")
print("나의 이름은 '한사람'입니다.")
print('나의 이름은 "한사람"입니다.')
print("나의 이름은 \"한사람\"입니다.")
print('나의 이름은 \'한사람\'입니다.')
print('-' * 40)
```

확장 문자(escape sequence)

1. 입력하기 어려운 문자 입력
2. 지원하는 확장 문자 목록

코드	설명
\n	문자열 안에서 줄을 바꿀 때 사용
\t	문자열 사이에 탭 간격을 줄 때 사용
\\	문자 \를 그대로 표현할 때 사용
\’	작은따옴표(‘)를 그대로 표현할 때 사용
\”	큰따옴표(“)를 그대로 표현할 때 사용
\r	캐리지 리턴(줄바꿈 문자, 현재 커서를 가장 앞으로 이동)
\f	폼 피드(줄바꿈 문자, 현재 커서를 다음 줄로 이동)
\a	벨 소리(출력 시 PC 스피커에서 ‘삑’ 소리가 난다)
\b	백 스페이스
\000	널 문자

```
tabby_cat = "\t탭이 적용됨."
persian_cat = "이 줄이 아닌 \n다른 줄에 나옴."
backslash_cat = "나는 \\ 시베리안 \\ 고양이."

fat_cat = """
해야 할 일 목록:
\t* 고양이 밥 주기
\t* 물고기
\t* 개박하\n\t* 오리새
"""

print(tabby_cat)
print(persian_cat)
print(backslash_cat)
print(fat_cat)
```

Input 함수 기본 사용법

1. Python의 표준 입력 함수
2. Input()
3. Input('문자열')
4. 입력 값의 데이터 형식은 문자열
5. 사용자 EOF → EOFError
   - 리눅스 – Ctrl + D
   - Windows – Ctrl + Z + Return

```
# 표준 입력 기본 사용법
print('이름을 입력하세요', end="")
name = input();
print("이름 : {0}, type : {1}".format(name,type(name)))
name = input('이름을 입력하세요 ');
print("이름 : {0}, type : {1}".format(name,type(name)))
name = input('아무것도 입력하지 말고 EOF(Ctrl+D 또는 Ctrl
+Z+Enter)를 입력해보세요');
```

```
아무것도 입력하지 말고 EOF(Ctrl+D 또는 Ctrl+Z+Enter)를 입력해보세요^Z
Traceback (most recent call last):
  File "d:/PythonForBeginner/OperatorAndIO/input1.py", line 7, in <module>
    name = input('아무것도 입력하지 말고 EOF(Ctrl+D 또는 Ctrl+Z+Enter)를 입력해보세요');
EOFError
```

정수 입력

1. 문자열 입력을 정수형으로 변환
2. eval() 함수
   - Python 표현식으로 반환
3. Int() 클래스
   - 숫자나 문자열에서 정수 객체 반환
   - 인자 없으면 0
   - 진법 (기본 10진법) → int(str, base)

```
# 표준 정수 입력
data = input("정수를 입력하시오 : ")
print(data, type(data))
# print(data, type(data), data + 1) 에러
# 문자열과 정수를 +(더하기)할 수 없습니다.

data = eval(input("정수를 입력하시오 : "))
print(data, type(data), data + 1)

data = int(input("정수를 입력하시오 : "))
print(data, type(data), data + 1)

data = int(input("2진수를 입력하시오 : "), 2)
print(data, type(data), data + 1)

data = int(input("8진수를 입력하시오 : "), 8)
print(data, type(data), data + 1)

data = int(input("10진수를 입력하시오 : "), 10)
print(data, type(data), data + 1)

data = int(input("16진수를 입력하시오 : "), 16)
print(data, type(data), data + 1)
```

실수 입력

1. 문자열 입력을 실수로 변환
2. eval(str) 함수
   - Python 표현식으로 반환
3. float(str) 클래스
   - 실수 객체 반환
   - 인자 없으면 0 반환

```
# 표준 실수 입력
data = input("실수를 입력하시오 : ")
print(data, type(data))
# 에러 문자열과 실수를 +(더하기)할 수 없습니다.
# print(data, type(data), data + 1.2)

data = eval(input("실수를 입력하시오 : "))
print(data, type(data), data + 1.2)

data = float(input("정수를 입력하시오 : "))
print(data, type(data), data + 1.2)
```

튜플과 리스트로 입력

1. 문자열 입력을 튜플과 리스트로 변환
2. eval(str) 함수
   - Python 표현식으로 반환

```
# 튜플 및 리스트 입력
string = input("(1,2) 처럼입력하시오 ")
print(string, type(string))

string = eval( input("(1,2) 처럼입력하시오 "))
print(string, type(string))

string = input("[1,2,3,4,5,6] 처럼입력하시오 ")
print(string, type(string))

string = eval( input("[1,2,3,4,5,6] 처럼입력하시오 "))
print(string, type(string))
```

argv와 input 응용

1. 프롬프트 변수로 반복 줄이기
2. 게임 실행 방식
3. 2종류 입력 함께 쓰기
   - 스크립트 실행 시 입력 - argv
   - 스크립트 실행 중 입력 - input

```
from sys import argv

script, user_name = argv
prompt = '> '

print(f"안녕 {user_name}, 나는 {script} 스크립트야.")
print("질문 몇 가지 해도 되지.")
print(f"{user_name}, 나 좋아?")
likes = input(prompt)

print(f"{user_name}, 어디 사니?")
lives = input(prompt)

print("어떤 컴퓨터 가지고 있어?")
computer = input(prompt)

print(f"""
좋아, 날 좋아하냐는 물음에 {likes}라고 대답했지.
넌 {lives}에 사는 구나.
어딘지 잘 모르겠다.
{computer} 컴퓨터를 가졌다니!  멋진걸.
""")
```