

# 변수와 함수/연산자

## 프로그램이 하는 일

- 정보(데이터)를 처리한다.

- **정보(데이터):** 변수(Variable)와 값(Value) 로 프로그램에 표현한다.
- **처리한다:** 연산자(Operator)와 함수(Function) 으로 프로그램에 표현한다.

표현(Expression): 어떤 것을 코드에 작성하는 방법을 표현(Expression)이라고 한다.  
 예를 들어 프로그램에 숫자 십을 표현할 때 10 이라고 한다. 이런 것을 정수의 표현/표현식 이라고 한다.

## 변수 (Variable)

- 데이터를 저장하는 메모리 공간
- 이 메모리 공간은 이름(변수이름)으로 관리된다.
  - 그래서 변수이름은 우리가 사용하는 값의 이름으로 사용된다.

## 변수 선언및 초기화

- 변수명 = 값
  - 변수는 반드시 선언하면서 값을 대입 해야 한다.
  - 선언시 대입할 값이 없으면 None을 대입 ( name = None )

## 식별자 규칙과 변수이름 주는 관례

### 식별자 규칙

- 식별자란 파이썬에서 사용하는 것들을 구분하기 위해 주는 이름을 말한다.
- 사용할 수 있는 문자는 **일반 문자**(영어 알파벳 뿐 만 아니라 한글 한자 등 모든 일반 문자를 사용할 수 있다.), **숫자**, **특수 문자는 \_(underscore) 만 가능**.
- 숫자는 **두번째 글자** 부터 사용할 수 있다.
- **\*\*예약어(keyword, reserved word)\*\***는 사용할 수 없다.
- 대소문자를 구별한다.

## 파이썬 키워드

### 변수이름 관례

- 모든 글자를 소문자로 한다.
- 여러단어를 조합할 경우 `_` 로 구분한다.
- 예: `name`, `age`, `customer_name`, `product_id`, `sale_price`

In [ ]:

### print() 함수

- ( ) 안에 전달해주는 값(Argument)을 문자열로 변환 후 화면(Teminal)에 출력하는 함수
  - 값에 엔터를 붙인뒤 출력한다.
- 여러개의 값을 출력할 경우 나열해서 전달하면 된다.
- 여러개의 값을 출력할 경우 구분자는 공백을 기본값으로 한다. 다른 구분자를 사용할 경우 `sep=구분자문자열` 를 전달한다.
- 뒤에 엔터 대신 다른 값을 붙일 경우 `end=붙일문자열` 을 전달한다.

In [ ]:

## 변수 사용

- 변수를 사용하는 것은 변수에 값을 변경(대입)과 변수가 가진 값을 조회하는 두가지가 있다.

### 변수에 값 대입/할당 (assignment)

1. 변수명 = 값
  - `=` 을 기준으로 왼쪽은 변수, 오른쪽은 대입할 값
  - 대입할 값으로 다음이 올 수 있다.
    - Literal 값(값 표현식)
    - 다른 변수
    - 연산식
  - 변수 초기화와 변수값 변경의 파이썬 구문은 동일하다. 변수를 처음 만들고 값을 대입하면 초기화이고 그 이후 할당하는 것들은 변경이 된다.
2. 여러 변수에 동일 한 값 대입
  - `a = b = c = 30`
3. 여러 변수에 다른 값 대입을 한 구문으로 처리
  - `a, b, c = 10, 20, 30`

In [ ]:

#### 4. 대입(할당) 연산자(Assignment operator)

- 변수의 값을 그 변수와 다른 값을 연산한 결과로 변경한다.

연산자	예	동일연산
=	x = 1	
+=	x += 1	x = x + 1
-=	x -= 1	x = x - 1
*=	x *= 1	x = x * 1
/=	x /= 1	x = x / 1
%=	x %= 1	x = x % 1
//=	x //= 1	x = x // 1
**=	x **= 1	x = x ** 1

In [ ]:

#### 변수값 조회

- 값을 사용해야 하는 자리에 변수명을 표현하면된다.
  - 변수명을 쓰면 그 변수가 가진 값을 조회해 사용한다.

In [ ]:

#### 변수를 메모리에서 제거

메모리에서 변수(데이터 저장공간)을 제거할 때는 `del` 변수명 명령어를 사용한다.

In [ ]:

#### 데이터 타입 (Data Type)

- 값들의 형태(type)에 따라 종류를 분류한 것
- 데이터타입은 값을 표현하는 방식과 관련 연산자들을 익히도록 한다.

#### None 값

- 아무 값도 없음을 나타내는 값

In [ ]:

## 숫자형(Numeric) - 정수, 실수

- 정수(int)와 실수(float) 형이 있다.
  - 정수 표현식
    - 10, 20, -1, -20, 0
    - 16진수 표기
      - 0x 로 시작 (0xAF29)
    - 8진수 표기
      - 0O 으로 시작 (0o71)
  - 실수
    - 20.1, 0.123411, 15.2321598
    - 지수표기법
      - $5e7 \rightarrow 5.0 \times 10^7$
      - $3e-7 \rightarrow 3 \times 10^{-7}$

In [ ]:

## 연산자

### 산술연산자

Operator(연산자)	설명	예
+	덧셈	$x + y$
-	뺄셈	$x - y$
*	곱셈	$x * y$
/	나눗셈	$x / y$
%	Modulus(나머지 나누기 연산자)	$x \% y$
//	Floor division(몫 나누기 연산자)	$x // y$
**	제곱연산	$x ** y$

In [ ]:

## 논리형(bool)

- 참(True)과 거짓(False)을 표현하는 데이터 타입
  - 제어문에서 많이 사용된다.
- 값 표현식
  - True, False

- bool 값이 들어가야 하는 구문에 다른 타입의 값을 작성하면 자동적으로 bool 타입으로 변환된다. (묵시적 형변환이라 한다.)
  - 0글자의 문자열, 숫자 0, None, 원소가 하나도 없는 자료구조는 False로 변환되고 그 이외는 True로 변환된다.

In [ ]:

- input([label]) 함수
  - 사용자로부터 값을 입력받을 때 사용한다.
  - 사용자의 입력을 기다렸다 사용자가 값을 입력하고 엔터를 치면 그 입력한 값을 읽어 반환한다.
  - 어떤 값을 입력받을 것인지 Label을 문자열로 전달할 수 있다.

In [ ]:

## 논리형 관련 연산자

### 비교 연산자

- 결과가 bool 타입
- 기준은 왼쪽의 피연산자이다.

연산자	설명	예
==	같은가?	x == y
!=	같지 않은가?	x != y
>	x가 큰가?	x > y
<	x가 작은가?	x < y
>=	x가 크거나 같은가?	x >= y
<=	x가 작거나 같은가?	x <= y

In [ ]:

### 논리 연산자

- 피연산자가 bool 타입이고 결과도 bool 타입이다.
- &, |, ^ 기호를 연산자로 사용할 때는 피연산자를 ( )로 묶어 줘야 한다.

연산자	설명	예
and, &	피연산자 모두 True이면 True를 하나라도 False이면 False를 반환.	x > 5 and x < 10
or,	피연산자 둘중 하나만 True이면 True를 둘다 False이면 False를 반환.	x > 5 or x < -4
^	XOR 연산. 피연산자가 다를 경우 True, 같으면 False를 반환.	(x > 5) ^ (x < 4)
not	단항연산자. 결과를 뒤바꾸어 반환. True는 False로 False는 True로 반환.	not(x < 5 and x < 10)

In [ ]:

## 조건 연산자

True일때 결과값 **if** 조건식 **else** False일때 결과값

- 조건식이 True이면 앞의 값을 False이면 뒤의 값을 연산결과로 반환한다.

In [ ]:

## 문자열 (string)

- 0글자 이상의 문자열을 표현한다.
- 파이썬 3.0 부터는 유니코드 문자열을 지원하므로 영문자 뿐 아니라 모든 나라의 철자들을 기본으로 사용할 수 있다.
- 문자열 표현식
  - 작은 따옴표 또는 큰 따옴표로 감싼다.
  - 여러줄 문자열(Multiline string)은 3개 짜리 작은따옴표 또는 큰따옴표로 감싼다.
    - 문자열 데이터에 엔터가 들어갈 경우 편리하다.
  - Escape 문자
    - 키보드에는 있지만 글자로 표현할 수 없는 문자들을 표현할 때 사용한다. 예) 엔터, Tab, Back space 등
    - 사용하는 글자가 원래 의미에서 벗어나(escape) 다른 의미로 쓰인다고 해서 escape문자라고 한다.
    - 파이썬 뿐 아니라 모든 프로그래밍 언어에서 사용하는 표준이다.
    - 표현: \문자

## 주요 Escape 문자

Escape문자	설명
\n	엔터
\t	Tab
\b	Backspace
\\	
\"	"
\'	'

- r-string (raw string)
- 문자열 앞에 r 접두어를 붙인다. (raw)
- escape 문자구분자인 \를 무시하여 escape 문자로 변환하지 않고 작성한 그대로 사용한다.
- ex) r"c:\test\example\a.txt"

In [ ]:

## 문자열 연산자

- 문자열 + 문자열
  - 두 문자열을 합친다.
  - 문자열은 문자열과만 + 연산을 할 수 있다. 다른 타입과 합칠 경우 다른 타입을 문자열로 변환한 뒤 연산해야 한다. (str() 함수이용)

In [ ]:

- 문자열 \* 정수
  - 문자열을 정수번 반복해서 합친다.

In [ ]:

- in, not in
  - 문자열A in 문자열B
    - 문자열A가 문자열B에 있으면 True, 없으면 False를 반환
  - 문자열A not in 문자열B
    - 문자열A가 문자열B에 없으면 True, 있으면 False를 반환

In [ ]:

- len(문자열)
  - 글자수를 반환한다.

In [ ]:

## Indexing과 Slicing

- Indexer 연산자
  - 집합형태[식별자]
    - 여러개의 값이 모여있는 집합 형태의 데이터에서 그 중 일부를 조회할 때 사용하는 연산자.
    - 문자열, 자료구조 등 다양한 데이터타입에 사용한다.
    - 조회하고자 하는 일부를 식별할 수 있는 식별자 를 [] 안에 넣어 조회한다.
- indexing과 slicing
  - indexing**: 집합내에서 하나의 값을 조회하는 방법
  - slicing**: 집합내에서 여러개의 값들을 범위로 지정해 조회하는 방법

## Indexing과 Slicing을 이용해 문자열내의 일부 문자들(sub string) 조회

- 문자열 내의 각 문자들은 식별자로 index를 가진다.
  - 양수 index
    - 문자열 앞에서 부터 붙여주는 index
    - 0부터 1씩 증가하는 값을 붙여준다.
    - 앞에서 몇번째 글자 식으로 조회할 때 사용한다.
  - 음수 index
    - 문자열 맨 뒤에서 부터 붙여주는 index
    - 1 에서 1씩 감소하는 값을 붙여준다.
    - 뒤에서 몇번째 글자 식으로 조회할 때 사용한다.
  - 모든 문자는 양수/음수 index 두개를 가진다.

안	녕	하	세	요	.		반	갑	습	니	다	.
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

### Indexing

- 문자열[index]
  - index의 문자를 조회
  - 변경은 안된다.
    - 문자열 값 안의 일부를 변경할 수 없다.
    - 문자열처럼 내부의 값을 변경할 수 없는 데이터 타입을 불변(Immutable)이라고 한다.

In [ ]:

### Slicing

- 기본구문: 문자열 [ 시작 index : 종료 index : 간격]
  - 시작 index ~ (종료 index - 1)
  - 간격을 지정하면 간격만큼 index를 증/감한다. (생략 시 1이 기본 간격)
- 0번 index 부터 조회 할 경우 시작 index는 생략가능
  - str\_value [ : 5] => 0 ~ 4 까지 조회
- 마지막 index까지 (끝까지) 조회 할 경우 종료 index는 생략 가능
  - str\_value[2 : ] => 2번 index 에서 끝까지
- 명시적으로 간격을 줄 경우
  - str\_value[ : : 3 ] => 0, 3, 6, 9.. index의 값 조회
  - str\_value[1 : 9 : 2] => 1, 3, 5, 7 index의 값 조회
- 시작 index > 종료 index, 간격을 음수로 하면 역으로 반환한다.(Reverse)
  - str\_value[5: 1: -1] => 5, 4, 3, 2 index의 값 조회
  - str\_value[: -1] => 마지막 index ~ 0번 index 까지 의미. Reverse 한다.



In [ ]:

## Format string (형식문자열) 생성

- 문자열에 문장 형식/구성/Layout을 미리 만들어 놓고 값은 나중에 대입하는 방식으로 문자열을 만드는 것
  - 여러 문자열이 같은 Layer에 특정 값들만 바뀌는 경우 사용한다.
    - 이름 : XXX 나이 : XXX 성별 : XXX
    - 기본 형식은 같은데 XXX에 들어갈 값들은 그때 그때 다를 경우 사용

### 1. format() 함수 이용

- 문자열을 만들 때 값을 나중에 넣을 위치 {} 로 표시하고 format() 메소드에서 {}에 들어갈 값을 순서대로 넣는다.

값을 넣어줄 자리를 지정하는 것을 **placeholder** 라고 한다.

In [ ]:

### 2. % value 사용

- 타입을 지정하는 % value를 이용해 값을 넣어줄 자리를 지정한다.
- format 문자열과 넣어줄 값은 % 로 구분한다.

% value	설명
%s	string
%d	int-정수
%f	float-실수
%%	%

실수는 소수점자릿수를 6자리로 맞추기 때문에 정밀도를 표현하는 표현식을 지정하는 것이 좋다. ex) %.2f

In [ ]:

### 3. f-string (format string)

- 파이썬 3.6에서 추가된 형식
- 문자열 앞에 접두어 f 를 사용한다.
- 값을 넣을 자리에 {변수명} 을 이용해 변수가 가진 값을 문자열에 추가한다.

In [ ]:

## string 주요 메소드

메소드	설명
<code>split(구분문자열)</code>	구분 문자열을 기준으로 나눈다.
<code>strip()</code> , <code>lstrip()</code> , <code>rstrip()</code>	앞뒤(strip) 앞(lstrip) 뒤(rstrip) 공백 제거
<code>replace("바꿀 문자열", "새문자열")</code>	바꿀 문자열을 새문자열로 변경
<code>count('세려는 문자열')</code>	string안에 세려는 문자열이 몇개 있는지 반환
<code>index(문자열)</code> , <code>find(문자열)</code>	문자열이 몇번째 index에 있는지 반환
<code>upper()</code> , <code>lower()</code>	모든 글자를 대문자(upper), 소문자(lower)로 변환
<code>capitalize()</code>	문자열의 첫글자를 대문자로 변환
<code>islower()</code> , <code>isupper()</code>	문자열이 모두 소문자(islower), 대문자(isupper)이면 True 아니면 False를 반환
<code>startswith("문자열")</code> , <code>endswith("문자열")</code>	문자열로 시작하는지/끝나는지 여부 반환

In [ ]:

## 데이터 타입 변환 (형변환) 함수

- 값의 데이터타입을 확인하는 함수
  - `type(값)`
- 정수로 변환
  - `int(value)`
- 실수로 변환
  - `float(value)`
- 문자열로 변환
  - `str(value)`
- 논리형으로 변환
  - `bool(value)`

In [ ]:

## 동적타입 언어

- 변수가 가질 수 있는 값의 타입을 고정하지 않는다. 그래서 같은 변수에 서로 다른 타입의 값을 저장할 수 있다.
- 자유도는 높지만 프로그램의 규모가 커지면 오류의 원인이 될 수 있다.
- 파이썬은 동적 타입 언어이다.

- 정적 타입 언어**
  - 변수를 선언할 때 그 변수에 저장할 수 있는 값의 타입을 고정시킨다.

In [ ]:

## TODO

In [ ]:

#1. 주민번호 "901211-1027213"의 앞 6자리만 조회해서 출력하시오.

#2. "안녕하세요" 를 10번 출력하시오.

#3. 다음 문자열의 글자수를 출력하시오.  
str\_value = "akd|clkdldle|ql39du7마구0≡"

#4.  
name="TV"  
price=300000  
maker = "LG"  
# 위 변수의 값을 다음과 같은 형태로 출력하시오.  
#"제품명 : TV, 가격 : 300000, 제조사 : LG"

#5.  
fruits = "사과 복숭아 귤 배"  
# 위 fruits에 "수박"이 있는지 확인하는 코드를 작성하시오.

#6.  
str\_value="aldkjaldjfalflksajfladlkaalalkdjfa"  
# str\_value 문자열안에 a가 몇개 있는지 출력하시오.

#7. 두개의 정수를 입력받아서 곱한 결과를 출력하는 코드를 작성하세요.