UNISTUDY

파이썬

3강. 변수와 자료형(연산자, 표준입출력 장치, 문자열)

1. 표현식과 문장

- 표현식 (expression)
 · 파이썬에서 어떠한 값을 만들어내는 간단한 코드
 · 값이란 숫자 수식, 문자열 등이 될 수 있음

```
273
10 + 20 + 30 * 10
"Python Programming"
```

```
print("Python Programming")
```

1. 표현식과 문장

- 문장 (statement)
  · 표현식이 하나 이상 모일 경우
  · 그 자체로 어떠한 값을 만들 수 없으면 문장이 아님
- 프로그램 (program)
  · 문장이 모여서 형성

```
+

-
```

2. 키워드

- 키워드 (keyword)
  · 특별한 의미가 부여된 단어
  · 파이썬에서 이미 특정 의미로 사용하기로 예약해 놓은 것
  · 프로그래밍 언어에서 이름 정할 때 똑같이 사용할 수 없음
  · 대소문자 구별

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

2. 키워드

- 아래 코드로 특정 단어가 파이썬 키워드인지 확인 가능

```
>>> import keyword
>>> print(keyword.kwlist)
```

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

3. 식별자 (identifier)

- 프로그래밍 언어에서 이름 붙일 때 사용하는 단어
- 변수 또는 함수 이름 등으로 사용
- 키워드 사용 불가
- 특수문자는 언더바(_)만 허용
- 숫자로 시작 불가
- 공백 포함 불가
- 알파벳 사용이 관례
- 의미 있는 단어로 할 것

사용 가능한 단어	사용 불가능한 단어
alpha	break
alpha10	
_alpha	273alpha
AlpHa	
ALPHA	has space

3. 식별자 (identifier)

- 스네이크 케이스와 캐멀 케이스
- 공백이 없어 이해하기 어려움

· 스네이크 케이스 (snake case) : 언더바(_)를 기호 중간에 붙이기
· 캐멀 케이스 (camel case) : 단어들의 첫 글자를 대문자로 만들기

- 파이썬에서는 스네이크 및 캐멀 케이스 둘 모두 사용

식별자에 공백이 없는 경우	단어 사이에 _ 기호를 붙인 경우 (스네이크 케이스)	단어 첫 글자를 대문자로 만든 경우 (캐멀 케이스)
itemlist	item_list	ItemList
loginstatus	login_status	LoginStatus
characterhp	character_hp	CharacterHp
rotateangle	rotate_angle	RotateAngle

3. 식별자 (identifier)

- 식별자 구분하기
- 캐멀 케이스에서는 첫 번째 글자를 소문자로 적지 않음

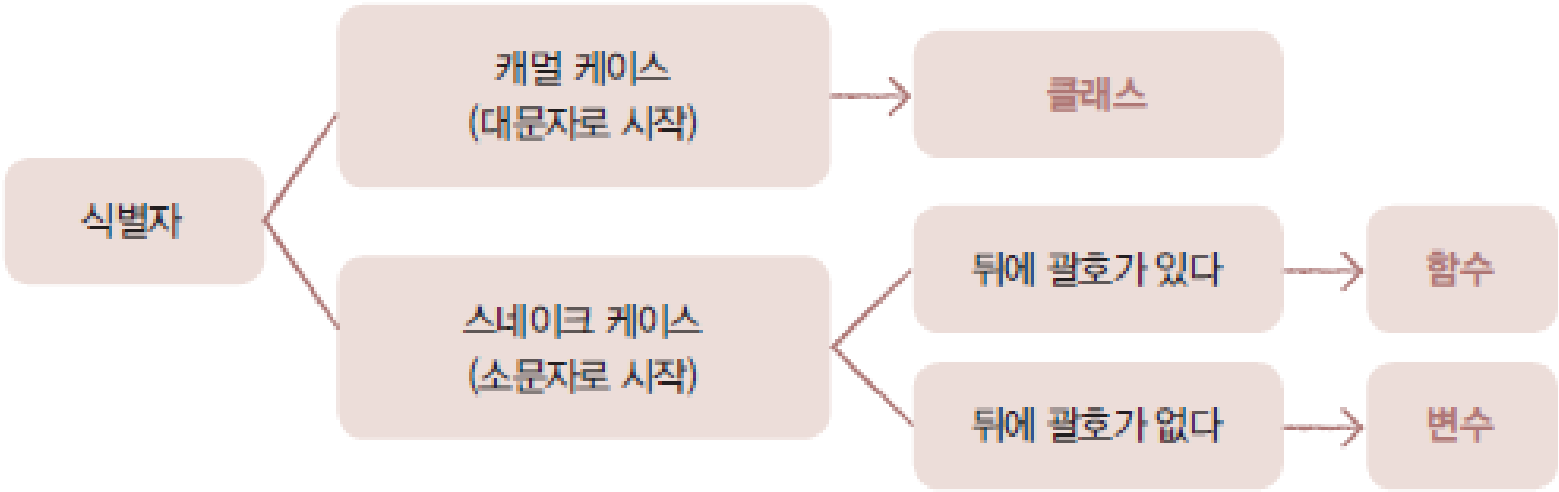
```
캐멀 케이스 유형 1 : PrintHello ⟶ 파이썬에서 사용합니다.
캐멀 케이스 유형 2 : printHello ⟶ 파이썬에서 사용하지 않습니다.
```

```
print          input          list          str          map          filter
```

```
Animal          Customer
```

3. 식별자 (identifier)

- 캐멀 케이스로 작성되었으면 클래스
- 스네이크 케이스로 작성되어 있으면 함수 또는 변수
- 뒤에 괄호 붙으면 함수
- 뒤에 괄호 없으면 변수



4. 주석 (comment)

- 프로그램 진행에 영향 주지 않는 코드
- 프로그램 설명 위해 사용
- # 기호를 주석으로 처리하고자 하는 부분 앞에 붙임

5. 연산자와 자료

- 연산자
· 스스로 값이 되는 것이 아닌 값과 값 사이에 무언가 기능 적용할 때 사용

```
>>> 1 + 1
2
>>> 10 - 10
0
```

5. 연산자와 자료

- 리터럴 (literal)
 · 자료 = 어떠한 값 자체 · 스스로 값이 되는 것이 아닌 값과 값 사이에 무언가 기능 적용할 때 사용

```
1
10
"Hello"
```

6. 출력 : print()

- print() 함수
  · 출력 기능
  · 출력하고 싶은 것들을 괄호 안에 나열
  · 하나만 출력하기

```
print(출력1, 출력2, ...)
```

```
>>> print("Hello! Python Programming...")
Hello! Python Programming...
>>> print(52)
52
>>> print(273)
273
```

6. 출력 : print()

- 여러 개 출력하기
- 줄바꿈하기

```
>>> print(52, 273, "Hello")
52 273 Hello
```

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

7. Summary

- 표현식 : 값을 만들어내는 간단한 코드
- 키워드 : 의미가 부여된 특별한 단어로, 사용자가 지정하는 이름에 사용할 수 없음
- 식별자 : 프로그래밍 언어에서 이름 붙일 때 사용
- 주석 : 프로그램 설명 시 사용하며, 프로그램 자체에 영향 주지 않음
- print() : 파이썬의 가장 기본적인 출력방법으로, 괄호 안에 출력하고 싶은 것을 입력

THANK YOU