

1. 파이썬 기본요소

1. 파이썬의 기본요소
2. 파이썬의 자료구조
3. 파이썬의 제어문
4. 함수와 모듈
5. 과학용 라이브러리
6. 시각화
7. 클래스



1. 파이썬 기본요소

1.1 파이썬의 기본 요소

식별자 / 키워드 / 코드작성규칙(명령 들여쓰기) / 주석

1.2 변수

1.3 파이썬의 자료형

1.4 문자열

연접 +	중복연접 *	문자참조
슬라이싱	join	split
Str		

1.5 연습문제



1.1 파이썬의 기본요소 - 식별자

- 기본요소

- 식별자, 키워드, 코드작성규칙(명령 들어쓰기), 주석

- 1.1 식별자

- Python 식별자는 변수, 함수, 클래스, 모듈, 혹은 기타 객체들을 식별하는데 사용하는 이름을 말한다.
- 식별자는 A부터 Z 또는 a부터 z까지의 문자, 또는 밑줄 문자(_) 다음에 0 개 이상의 문자나 밑줄문자 혹은 숫자를 결합하여 만들 수 있다.
- Python 식별자에는 @, \$, and % 등과 같은 구분 문자를 사용할 수 없다.
- Python은 대소문자를 구분한다.
- Python 식별자의 명명법은 다음과 같다.
 - 클래스 이름은 첫 글자는 대문자로 시작하고, 다른 식별자들은 소문자를 사용한다.
 - 첫 문자가 단일 밑줄 문자(_)인 식별자는 private을 의미한다.
 - 두 개의 밑줄 문자(__)로 시작하는 식별자는 매우 강한 private 식별자이다.
 - 밑줄 문자 두 개로 끝나는 식별자는 Python에서 사용하는 특수 이름 이다.

1.1 파이썬의 기본요소 - 키워드

- 1.2 키워드

- 다음 스크립트는 Python의 현재 버전의 키워드(keywords)를 알아 보기 위한 명령어와 그 결과이다.

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

- 이 키워드는 상수나, 변수 혹은 다른 식별자들의 이름으로 사용할 수 없다.
- 예약어는 모두 소문자로 구성되어 있다.
- None은 키워드는 아니지만 Python에서 상수로 사용하고 있는 내장 객체이므로, 사용자가 다른 용도로 사용하면 안 된다.

1.1 파이썬의 기본요소 - 프로그램 줄과 들여쓰기

• 1.3 프로그램 코드 작성 규칙

- 클래스, 함수, 흐름 제어용 코드 블록을 구분하기 위해 들여쓰기를 사용한다.
- 들여쓰기를 할 때 공백(space) 문자의 개수는 변할 수 있으나 해당 블록 내에서는 반드시 공백 문자의 개수가 같아야 한다.
- 들여쓰기는 공백문자 또는 탭으로 할 수 있다.

```
if True:  
    print('True')  
else:  
    print('False')
```

정상적으로 실행되는 예

```
if True:  
    print('Answer')  
    print('True')  
else:  
    print('Answer')  
    print('False')
```

정상적으로 실행되지 않는 예

1.1 파이썬의 기본요소 - 한 명령어 여러 줄에 작성하기

- 한 명령어 여러 줄에 작성하기

- Python의 명령줄은 개행 문자(new line)로 끝난다.
- Python은 줄 연속 문자(line continuation character) (\)를 사용하여 현재 문장이 다음 문장과 연결되어 있다는 것을 나타낸다.

```
total = item_one + \  
        item_two + \  
        item_three
```

- [], {}, 또는 () 사이에 있는 내용들은 줄 연속 문자가 필요 없다. 예를 들면, 다음과 같이 한 문장을 두 줄로 나누어 쓸 수 있다.

```
days = ['Monday', 'Tuesday', 'Wednesday',  
        'Thursday', 'Friday']
```

- 한 줄에 여러 명령어 작성

- A=3; b=5

1.1 파이썬의 기본요소 - 주석문

- 1.4 주석문

- 문자열 외부에 있는 해쉬 문자(#)는 주석의 시작을 의미한다.
- Python 해석기는 그 줄에서 # 문자 뒤의 모든 내용은 주석으로 간주하여 무시한다.

```
# First comment  
print "Hello, Python!" # second comment
```

- 사용자 입력 대기

- 다음 명령줄은 화면에 프롬프트를 표시하고, 사용자가 엔터를 누를때 까지 대기한다

```
#!/usr/bin/python  
input("\n\nPress the enter any key to exit.")
```

1.2 변수

- 변수
 - 데이터를 저장하는 저장장소(특정 메모리)이다
 - 예, $a = 3$
 - 변수선언(대입문)은 변수이름(a) 대입연산자(=) 상수데이터(3)로 구성된다.
 - 변수는 대입되는 데이터의 형에 따라 정수형 변수, 실수형 변수 등으로 불리어 진다.
- 변수 이름 명명 규칙 : 실별자 명명규칙
 - 대소문자를 구분한다.
 - 첫 글자는 숫자가 될 수 없다.
 - 문자(letters), 숫자, 그리고 밑줄문자(underscores)로 구성된다.
 - Python 문법에 사용되는 예약어와 라이브러리의 함수명은 변수 이름으로 사용할 수 없다.
 - 변수명은 한글도 허용된다.
- Python에서 변수를 사용할 때 **변수형을 사전에 선언하지 않는다**
- 대화형 모드에서 가장 마지막으로 화면에 나타난 값은 변수 `_`에 저장된다.

```
>>> a=3 #정수 상수 3을 정수형 변수 a에 대입(저장)
>>> b=4 #정수 상수 4를 정수형 변수 b에 대입(저장)
>>> a+b #변수 a와 변수 b에 저장된 내용을 더하여 출력
7
>>> d=2.5 #d는 실수가 저장되므로 실수형 변수이다
>>> a+d #정수와 실수가 연산되면 실수형이 된다
5.5
>>> 내나이 = 22 #한글도 변수명이 될 수 있다.
>>> 내나이 #변수 내나이에 저장된 내용 출력
22
>>> _+10 #_에는 바로 전에 내용이 저장되어 있다.
32
```


1.3 파이썬의 자료형

- 기본자료형

- 정수형 (integer), 긴정수형 (long integer: 메모리가 허락하는 한 무제한의 자리수로 정수를 계산할 수 있다),
- 부동소수점수형, 복소수형, 문자열형, 유니코드문자형, 함수형 등이 있다.

- 시퀀스(sequence) 자료형

- 문자열
- 리스트
- 튜플 (tuple)
- 사전형 (dictionary)

```
>>> a=3+4j
>>> b=2+5j
>>> a*b
(-14+23j)
>>> a+b
(5+9j)
>>> 1+j
Traceback (most recent call last)
  File "<stdin>", line 1, in <mo
NameError: name 'j' is not defin
>>>
```

```
>>> c=complex(5,4)
>>> c
(5+4j)
>>> a/b
(0.89655172413793105-0.241379310344482757j)
>>> c.real
5.0
>>> c.imag
4.0
>>> abs(3+4j)
5.0
>>> (3+5j).real
3.0
>>> (3+5j).imag
5.0
>>> z.conjugate()
(1-1j)
>>>
```

1.4 문자열

- 문자
 - 'a', 'b', 'c' 와 같은 알파벳을 문자(상수)라고 하고
- 문자열
 - 문자의 조합으로 만들어지는 것을 ‘문자열’ 이라고 한다.
 - 문자와 문자열은 모두 문자열로 취급하고 작은 따옴표 또는 큰 따옴표로도 표기한다.
- 문자열 변수
 - name = ‘Gilldong Hong’

```
>>> 'spam eggd'      #문자열 상수
'spam eggd'
>>> 'doesn\'t'       #문자 ‘삽입
"doesn' t"
>>> "yes", he said'  #문자열속의 문자열
"yes", he said'
>>> hello="This is \ #문자열(상수)을 저장하는 문자열 변수 선언의 예
... rather long string containing several lines. “
>>> print(hello)
'This is rather long string containing several lines.
>>> hello            #print(hello)와 같다
'This is rather long string containing several lines.'
```

1.4.1 문자열 연산

• 4.1 연산

- 문자열 연접 +, 중복연접 *, 문자참조, 마지막 문자참조 -1
- slicing : 문자열의 일부를 조각내기
- 참조는 가능하지만 변경은 불가하다.

```
>>> word = 'Happy' + 'Day' #두문자열을 연접하여 문자형변수 word에 저장
>>> word #문자열변수의 저장내용 출력
'HappyDay'
>>> '<'+word+'>' #문자열 연접
'<HappyDay>'
>>> '<'+word*2+'>' #문자열 복제 와 연접
' <HappyDayHappyDay> '
>>> word[0] #문자열변수 word의 인덱스 0 번째 문자참조
'H'
>>> word[1] #문자열변수 word의 1번째 문자참조
'e'
>>> word[0:3] #word의 0부터 3전까지의 부분문자열(slicing)
'Hap'
```

index	0	1	2	3	4	5	6	7
word	H	a	p	p	y	D	a	y
reverse index	-8	-7	-6	-5	-4	-3	-2	-1

```
>>> word
'HappyDay'
>>> word[0:3] #문자열 처음부터 3 문자열 slicing
'Hap'
>>> word[:3] #word[0:3] 과 같음
'Hap'
>>> word[-1] #마지막 문자 참조
'y'
>>> word[-3:-1] #뒤에서 3번째부터 1번째 전까지 slicing
'Da'
>>> word[-3:] #뒤에서 3번째부터 마지막까지 slicing
'Day'
>>> word[0]='X' #문자열변수에 문자원소 대입은 불허
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
>>>
```

1.4.1 문자열 연산

- join : 리스트의 문자열을 연결한다.
- split : 문자열을 분리하여 리스트를 만든다.
- str : 객체를 문자열로 만든다.

```
>>> ';'.join(['abc','123','xyz','789' ]) #리스트 요소를 ';' 로 연결한다.
' abc;123;xyz;789 '
>>> ' abc;123;xyz;789 ' .split( ';' ) #문자열(str)을 문자 ;를 경계로 분리하고 리스트를 만들어 반환한다.
['abc', '123', 'xyz', '789']
>>> str(99) #정수를 문자열로 변환
' 99 '
>>> str([1,2,3, ' a' , ' b' , 'c' ]) #리스트 원소를 문자열로 변환
"[1, 2, 3, 'a', 'b', 'c']"
>>> a=5 # 정수 5 대입
>>> str(a==5) # 논리연산결과를 문자열 'True' 로 반환
'True'
>>> str(a==1) # 논리연산결과를 문자열 'False' 로 반환
'False'
>>> 'Today is '+'Stepember '+str(3)+' , '+str(2019)
'Today is Stepember 3, 2019'
>>> ';'.join([1,'abc',2,'edf' ]) #join은 문자열 원소에 대해서만 적용된다.
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: sequence item 0: expected str instance, int found
>>>
```

1.5 연습문제

1. 삼성전자를 42,000원에 100주 sk하이닉스를 72,000원에 30주를 매입하였다면 총 매입가를 계산하는 프로그램을 작성하세요.
2. 주가가 각각 5%, 10% 하락한 경우에 손실액을 구하는 프로그램을 작성하세요.
3. 2014년도 여름 여행중 Death Valley National Park의 온도가 115도 화씨 이었다. [\[link\]](#) 화씨 온도(F)를 섭씨 온도(C)로 변하는 프로그램을 작성하세요.
변환공식 $C = (F - 32) / 1.8$
4. 화면에 "pizza"를 10번 출력하는 프로그램을 작성하세요.
5. 다음 형식과 같이 본인의 이름, 생년월일, 주민등록번호를 출력하는 프로그램을 작성해 보세요.
이름: 파이썬 생년월일: 2014년 12월 12일 주민등록번호: 20141212-1623210
6. s라는 변수에 'Daum KaKao'라는 문자열이 바인딩(저장)되어 있다고 했을 때 문자열의 슬라이싱 기능과 연결하기를 이용해 s의 값을 'KaKao Daum'으로 변경해 보세요.
7. 변수에 'hello world'라는 문자열이 바인딩되어 있다고 했을 때 a의 값을 'hi world'로 변경해 보세요.

