PYTHON PORTFOLIO

20170684 장동익

CONTENTS

001 파이썬 언어의 개요와 첫 프로그래밍

002 파이썬 프로그래밍을 위한 기초 다지기

003 문자열과 논리연산

004 조건과 반복

005 항목의 나열인 리스트와 튜플

006 문자열과 논리연산

Part 1.

파이썬 언어의 개요와 첫 프로그래밍

파이썬이란?

- 1991년 네덜란드의 귀도 반 로섬(Guido van Rossum)이 개발
- C, C++, 자바 등 어떤 컴퓨터 프로그래밍 언어보다 배우기 쉬움
- 직관적이고 이해하기 쉬운 문법
- 객체 지향의 고수준 언어
- 앱(App)과 웹(WEB) 프로그램 개발 목적
- 웹 서버, 과학 연산, 사물 인터넷(IOT), 인공지능, 게임 등의 프로그램 개발하는 강력한 도구

파이썬의 특징

1. 직관적이고 쉽다.

   아주 간단한 영어 문장을 읽듯이 보고 쉽게 이해할 수 있도록 구성

2. 널리쓰인다.

   구글, 아마존, 핀터레스트, 인스타그램, IBM, 디즈니, 야후, 유튜브, 노키아, NASA 등과
   네이버, 카카오톡의 주력 언어 중 하나

3. 개발 환경이 좋다.

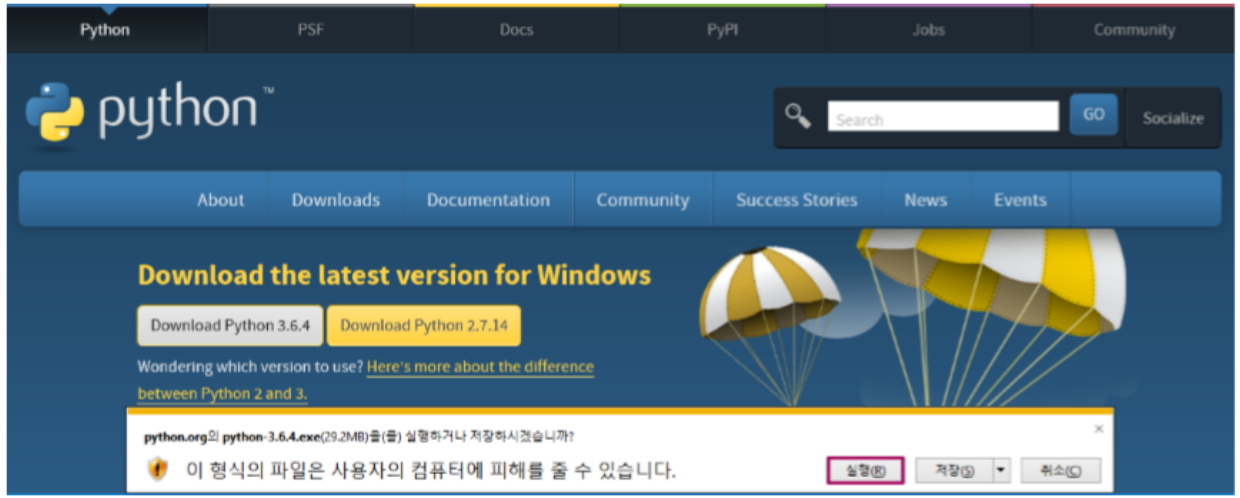
   게임, 인공지능, 수치해석 등 다양한 라이브러리와 커뮤니티 활성화

파이썬 프로그램 설치 1

- 프로그램 다운로드 받기

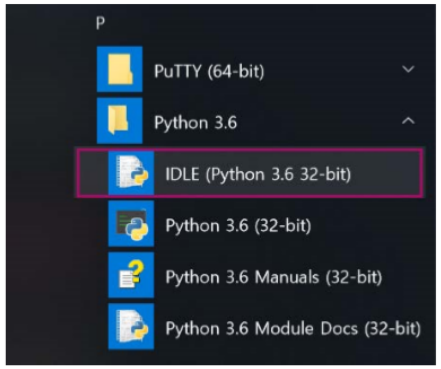
  http://python.org

- Downloads > Download Python 3.X.X > 실행 클릭



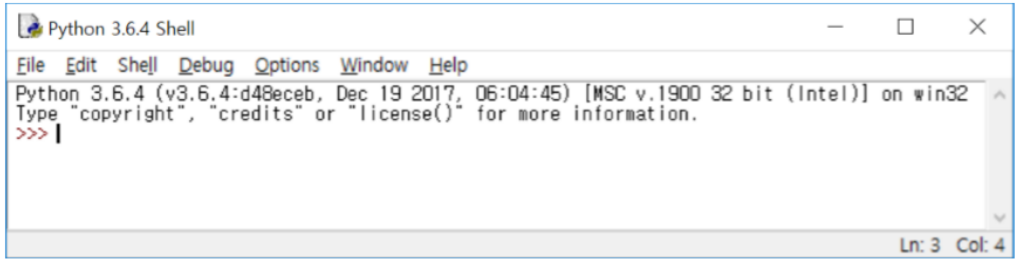
IDLE 프로그램 시작

- **IDLE** : 'Integrated Development and Learning Environment'의 약어로 파이썬의 '**통합 개발과 학습 환경**'



IDLE 파이썬 쉘 (Python shell)

- **IDLE** : 'Integrated Development and Learning Environment'의 약어로 파이썬의 '**통합 개발과 학습 환경**'



IDLE 파이썬 쉘 (Python shell)

- **파이썬 쉘** : 직접 파이썬 명령을 입력하고 엔터 키를 누르면 바로 그 결과가 쉘 화면에 출력
- 파이썬 쉘을 계산기처럼 사용

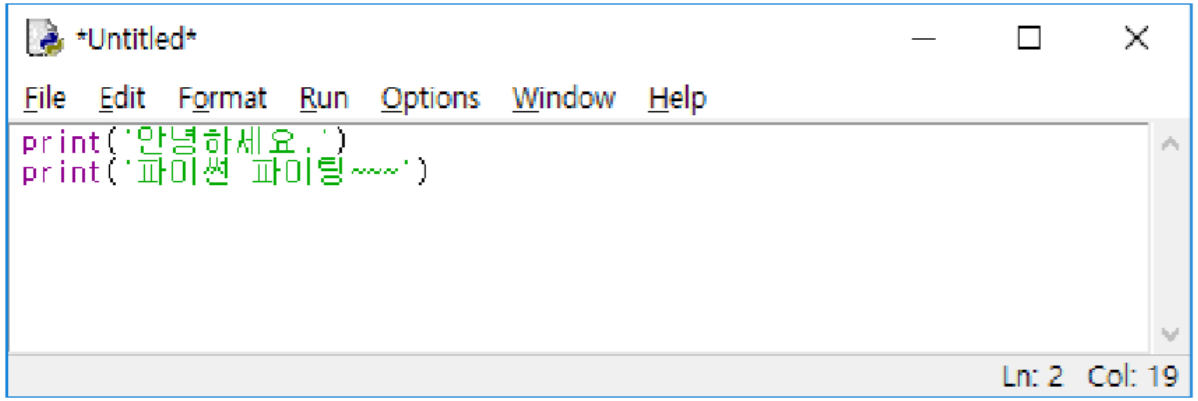
```
>>> 10 + 20

>>> 10 + 20 * 30

>>> 10 * 20 + 30 * 50
```

파이썬 프로그램 작성

- 텍스트 에디터



- 파일 저장 : hello.py

Part 2.

파이썬 프로그래밍을 위한 기초 다지기

변수란?

- 변수(Variable)는 값을 저장하는 박스
- 변수를 만든다는 것은 숫자나 문자열과 같은 데이터를 저장할 수 있는 공간을 마련하는 것
- 수학의 방정식에서의 x + y = 3 에서 x와 y는 어떤 변하는 값을 가지게 되는 변수
- 컴퓨터에서 변수도 수학 변수의 개념과 유사

변수 값의 저장과 출력

```
① >>> a = 5
② >>> print(a)
  5
```

```
① >>> a = 3
  >>> b = 7
② >>> c = a + b
③ >>> print(c)
  10
```

변수명의 규칙

- 변수명은 영문자 소문자로 시작
- 유효한 변수명 : a, b, x, y, l, j, str, animal, computer, age, sum, type1, type2, num2 ...
- 잘못된 면수명 : 12month, 10rule, 3numer

①
```
>>> animal = '사자'
>>> print(animal)
사자
```
②
```
>>> num1 = 7.8
>>> num2 = 3.57
>>> print(num1 + num2)
11.37
```
③
```
>>> 12month = '봄'
SyntaxError: invalid syntax
```

숫자 연산자 정리

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기
%	나머지 연산
//	나눈 후 소수점 이하 절삭
**	제곱 구하기

Part 3.

문자열과 논리연산

문자열

- 문자열(String) : 하나 또는 여러 개의 문자로 구성된 데이터형
- 문자들의 앞과 뒤에 쌍 따옴표(") 또는 단 따옴표(')를 붙임
- "안녕하세요."와
  '안녕하세요.'는 동일

문자열 길이 구하기 : len()

```
>>> message = '안녕하세요!'
① >>> str_len = len(message)
② >>> print('문자열의 길이 : ' + str(str_len))
문자열의 길이 : 6
```

데이터 형 변환에 사용되는 함수

- **int()**

함수 int()는 실수Floating point나 문자열String을 정수형 숫자로 변환

- **float()**

함수 float()는 정수나 문자열을 실수로 변환

- **str()**

함수 str()은 정수형이나 실수형 숫자를 문자열로 변환

화면에 출력하기 : print()

- 컴퓨터 화면에 결과를 출력할 때 print() 함수 사용
- print() 함수를 사용법 4가지

1. print() 함수의 기본 사용
2. 파라미터 sep을 사용
3. 문자열 연결 연산자 +를 사용
4. 문자열 포맷 코드 %를 사용

문자열 포맷팅 코드의 예

포맷팅 코드	설명
%d	정수형 숫자
%s	문자열
%.2f	실수형 숫자, .2는 소수점 둘 째 자리까지 나타냄.
%%	% 기호 자체를 나타내는 데 사용함.
%6s	6자리의 문자열
%5d	5자리의 정수형 숫자

Part 4.

조건과 반복

if문의 세 가지 유형

if문의 유형	사용 형태
(1) if ~ 구문	만약 이 조건을 만족하면 ~ 해라!
(2) if ~ else ~ 구문	만약 이 조건을 만족하면 ~ 하고, 그렇지 않으면 ~ 해라!
(3) if ~ elif ~ else ~ 구문	만약 조건1을 만족하면 ~ 하고, 조건2를 만족하면 ~하고, ..., 그렇지 않으면 ~ 해라!

if문

- 어떤 수가 양수인지 아닌지를 판단하여 결과를 출력

```
if x > 0 :
    print('양수이다!')
else :
    print('음수 또는 0이다!')
```

if~ else~ 구문

```
if 조건식 :
    <문장1, 2, …>
else
    <문장A, B, …>
```

- 조건식이 참이면 <문장 1, 2, ….>를 수행
- 조건식이 거짓이면 else 다음의 <문장A, B, …>를 수행

if~ elif~ else~ 구문

```
if 조건식1 :
    <문장1, 2, ...>
elif 조건식2
    <문장A, B, ...>
...
else
    <문장i, ii, ...>
```

- 조건식1이 참이면 <문장 1, 2, ...>를 수행
- 조건식2가 참이면 <문장 A, B, ...>를 수행
- 조건식들이 모두 거짓이면 else 다음의 <문장i, ii, ...>를 수행

if문의 중첩

```
if 조건식 :
    <문장들>
elif 조건식 :
    if 조건식 :
        <문장들>
    else :
        <문장들>
else :
    <문장들>
```

for문의 기본 형식

```
① for 변수 in range(반복 횟수의 범위) :
②     문장1
      문장2
      ……
```

while문의 기본 형식

```
①  변수 값 초기화
②  while 조건식 :
③      문장1
       문장2
       ……
④      변수 값의 증가(또는 감소)
```

break문 : 반복 루프 빠져 나가기

```
① for i in range(1, 1001) :
②     print(i)

③     if i == 10 :
④         break
```

: : 실행 결과

```
1
2

...

10
```

Part 5.

항목의 나열인 리스트와 튜플

리스트

- 리스트 : 여러 개의 데이터 값을 하나의 변수에 담을 수 있는 데이터 구조

리스트명 = [데이터, 데이터, 데이터, ...]

```
score = [90, 89, 77, 95, 67]
fruit = ['apple', 'banana', 'orange']
```

append() 함수로 리스트 요소 추가

```
flower = ['무궁화', '장미', '개나리']
print(flower)

① flower.append('벚꽃')
② print(flower)
```

: : 실행 결과

```
['무궁화', '장미', '개나리']
['무궁화', '장미', '개나리', '벚꽃']
```

두 리스트를 하나로 합치기

```
    person1 = ['kim', 24, 'kim@naver.com']
    person2 = ['lee', 35, 'lee@hanmail.net']

①   person = person1 + person2
②   print(person)
```

: : 실행 결과

['kim', 24, 'kim@naver.com', 'lee', 35, 'lee@hanmail.net']

리스트 요소 삭제

```
member = ['황지웅', 20, '경기도 김포시', 'jiwoang@codingschool.info', '123-1234-5678']
print(member)

① member.remove(20)
② print(member)
```

: : 실행 결과

['황지웅', 20, '경기도 김포시', 'jiwoang@codingschool.info', '123-1234-5678']
['황지웅', '경기도 김포시', 'jiwoang@codingschool.info', '123-1234-5678']

튜플이란?

- **튜플 : 리스트의 형태와 사용법이 거의 유사**
- **튜플이 리스트와 다른 점**

1. 튜플에서는 리스트의 대괄호([ ]) 대신에 소괄호(())를 사용
2. 튜플에서는 리스트와는 달리 요소들의 수정과 추가가 불가

튜플을 이용한 음식점 메뉴 관리

```
① menu = ('짜장면', '우동', '짬뽕', '볶음밥')

② print(menu)
③ print(menu[0])
④ print(menu[2])
⑤ print(menu[0:3])

⑥ menu[1] = '사천탕면'
```

: : 실행 결과

```
('짜장면', '우동', '짬뽕', '볶음밥')
짜장면
짬뽕
('짜장면', '우동', '짬뽕')
Traceback (most recent call last):
  File "H:/첫_파이썬
/source/06/tuple_menu.py", line 8, in
<module>
    menu[1] = '사천탕면'
TypeError: 'tuple' object does not support
item assignment
```

튜플 합치고 길이 구하기

```
① tup1 = (10,20,30)
② tup2 = (40,50,60)

③ tup3 = tup1 + tup2

④ print(tup3)
⑤ print(len(tup3))
```

: : 실행 결과

```
(10, 20, 30, 40, 50, 60)
6
```

Part 6.

문자열과 논리연산

딕셔너리란?

- **딕셔너리 : 인덱스를 의미하는 '키'와 자료의 내용인 '값'을 이용하여 자료를 관리**

```
score = {'kor':90, 'eng':89, 'math':95}
member = {'name':'홍길동', 'age':18, 'phone':'01037873146'}
```

딕셔너리 기본 구조

```
① members = {'name': '황예린', 'age': 22, 'email': 'yerin@codingschool.info'}

② print(members)
③ print(members['name'])
④ print(members['age'])

⑤ print('길이 : %d' % len(members))
```

: : 실행 결과

```
{'name': '황예린', 'age': 22, 'email': 'yerin@codingschool.info'}
황예린
22
길이 : 3
```

딕셔너리 요소 추가/수정/삭제

```
① name = '홍지수'
② scores = {'kor': 90, 'eng': 89, 'math': 95, 'science': 88}
③ print(scores)

④ scores['kor'] = 70
   print(scores['kor'])

⑤ scores['music'] = 100
   print(scores)

⑥ del scores['science']
   print(scores)

⑦ print('이름 : %s' % name)
   print('국어 : %s' % scores['kor'])
   print('영어 : %s' % scores['eng'])
   print('수학 : %s' % scores['math'])
```

: : 실행 결과

```
{'kor': 90, 'eng': 89, 'math': 95, 'science': 88}
70
{'kor': 70, 'eng': 89, 'math': 95, 'science': 88, 'music': 100}
{'kor': 70, 'eng': 89, 'math': 95, 'music': 100}
이름 : 홍지수
국어 : 70
영어 : 89
수학 : 95
```

감사합니다

20170684 장동익