

파이썬 프로그래밍

파이썬 시작하기



한국기술교육대학교
온라인평생교육원

■ 스크립트 언어의 이해

1. 스크립트 언어란 무엇인가?

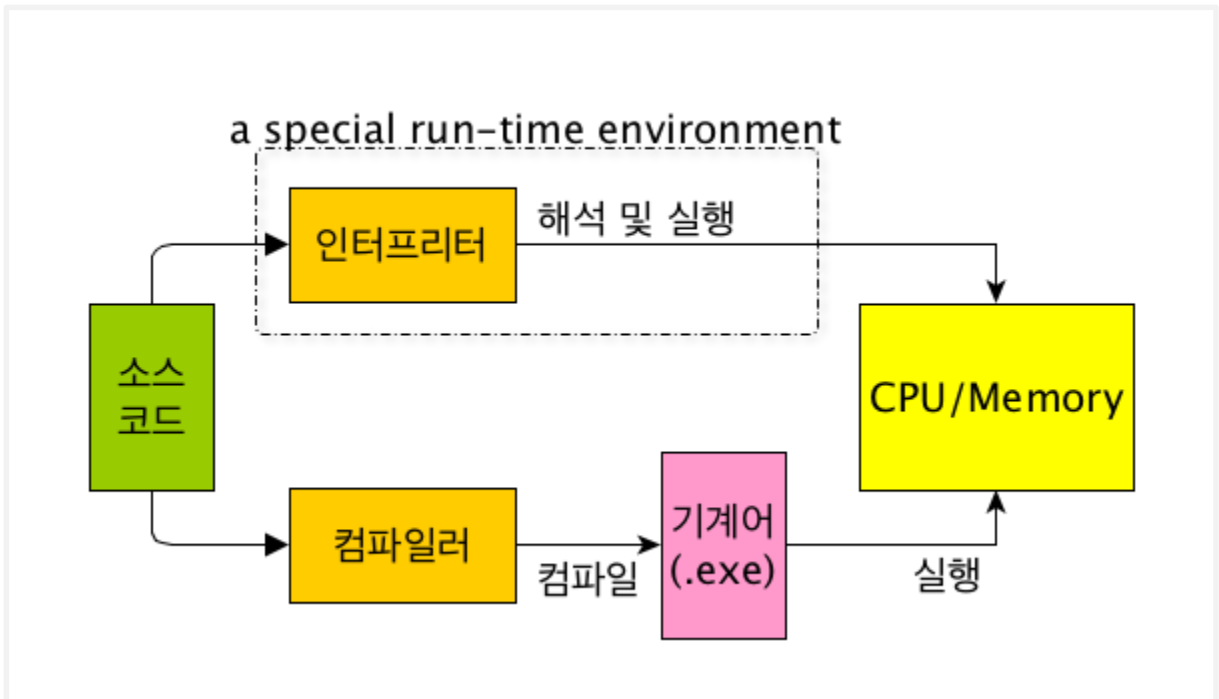
- 스크립트(scripts)의 사전적 의미 - 연극의 대사등이 적혀 있는 스크립트에서 유래
- Script Language의 wikipedia 정의
(http://en.wikipedia.org/wiki/Scripting_language)
: a programming language that supports scripts, programs written for a special run-time environment that can interpret (rather than compile) and automate the execution of tasks that could alternatively be executed one-by-one by a human operator

- 소스코드 = 스크립트
- 연기자가 스크립트를 보고 연기 = 컴퓨터가 스크립트를 읽어 수행
- a special run-time environment = 수행시간 환경
- 해석과 수행을 함
- 라인 단위로 해석하여 수행함

■ 스크립트 언어의 이해

2. 컴파일 언어와 스크립트 언어와의 비교

- 컴파일(Compile) 방식의 언어에 의해 작성되는 응용 프로그램은 컴파일러에 의해 기계어로 번역된 채로 실행되기 때문에, 수정이 빈번하게 발생할 경우에는 수정 후 다시 컴파일을 해야 함.
- 덩치가 큰 프로그램은 컴파일 시간이 꽤 길다. 즉, 간단한 수정에도 오랜 기간의 컴파일 시간이 요구된다.

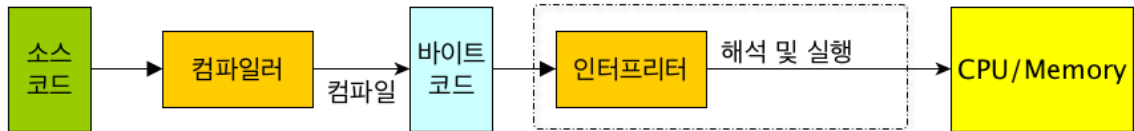


- 리눅스나 맥은 .exe. 파일이 아니어도 실행파일로 존재 가능
- 컴파일러 언어 단점 : 자원을 많이 요구하고 시간 소요됨
- special run-time environment 덕분에 단계가 많이 줄어들음
- special run-time environment = 인터프리터

■ 스크립트 언어의 이해

2. 컴파일 언어와 스크립트 언어와의 비교

- 반면에 수정이 빈번하게 발생하는 경우에는 소스 코드를 한 줄 한 줄 읽어 바로 바로 실행하는 인터프리터 방식이 상당히 유리하다.
- 스크립트 소스코드를 컴파일 방식에 의해 중간 코드(Bytecode)로 우선 만들고, 이를 다시 인터프리터 방식으로 해석하여 수행하는 방법도 종종 활용된다.



- 파이썬의 경우 두 방법 모두 활용 가능
- 바이트 코드(읽을 수 있는 언어) ≠ 기계어 (이진수, 0, 1)

■ 스크립트 언어의 이해

2. 컴파일 언어와 스크립트 언어와의 비교

- 스크립트 언어의 장점
 - 개발 시간이 단축된다.
 - 소스 코드 수정이 빠르고 간단하게 이루어진다.

- 소스코드의 수정을 많이 요구함 → 스크립트 언어가 더 용이

- 스크립트 언어의 단점
 - 중간 코드를 만드는 것은 간단하지만 그것을 실제로 실행시키는 것은 많은 작업을 필요로 한다.
 - 실행 시간이 오래 걸린다.

- special run-time environment = Java 가상머신
- Java → 컴파일 언어가 기본이지만 인터프리터 사용도 혼용
- 실행시간을 좀 더 단축하고자 수치분석이 많이 요구됨

■ 스크립트 언어의 이해

3. 대표적인 스크립트 언어

JavaScript
ActionScript
Perl
PHP
Python
Lua
Ruby

- 브라우저에서 웹페이지를 동적으로 만들어주는 Javascript
- Actionscript : flash 개발 시 사용
- Perl : 요즘에 많이 사용 X
- PHP : 동적인 웹 페이지 구현 시 많이 사용
- Lua, Ruby → 최근에 개발된 스크립트 언어

파이썬 프로그래밍

파이썬 시작하기



한국기술교육대학교
온라인평생교육원

■ 파이썬 언어의 유래 및 파이썬 언어의 특징

1. 파이썬의 사전적 의미

- Python의 사전적 의미는 독이 없는 뱀으로 먹이를 몸으로 감아서 압사시키는 큰 뱀, 또는 그리스 신화에 나오는 악마

2. 파이썬의 유래

- 프로그래밍 언어로서 Python 이름의 유래는 1970년대에 영국 BBC에서 방영한 "Monty Python's Flying Circus" 방영 프로그램에서 유래되었다. 이 프로그램은 매우 어색하고 이상한 행동을 보이는 사람들을 쇼형태로 방영하여 많은 인기를 누렸다.
- 이 언어를 만든 Guido van Rossum이 이 프로그램의 대단한 팬이었기 때문이었다.
- 1989년 12월 크리스마스 휴가를 보내고 있던 Guido는 휴가기간 동안 무엇인가 재미있게 할 수 있는 무언가를 찾고 있었다.
- 연휴라서 연구실이 닫힌 상황에서 집에서 컴퓨터를 가지고 예전부터 생각하고 있었던 인터프리터(interpreter)를 만들어 보기로 결심하였다.
- 즉, Guido가 무료한 한때를 보내기 위하여 만들기 시작했던 언어가 지금 많은 사람들이 사용하는 Python 언어가 되었다.

3. 파이썬의 출발

- 파이썬의 기반은 80년대 초반에 Guido가 동료들과 함께 개발을 하였던 ABC라는 언어이다.
- 80년대에 ABC라는 언어는 최고의 교육용 언어로 알려졌었고 전문 프로그래머가 아닌 사람이 배우기 편하게 만들어진 아주 훌륭한 언어였다.
- 그럼에도 불구하고 전문 프로그래머들에게는 거의 받아 들여지지 않고 잘 사용되지도 않았다.
- 이는 프로그래밍이 대중화가 많이 되지 않은 상황에서 전문 프로그래머들에게 외면 받았을 때에 나올 수 있는 결과라고 볼 수 있다.
- Guido는 파이썬으로 실제 문제를 더욱더 쉽게 해결할 수 있도록 설계하였으며 C언어나 유닉스 쉘을 대신해서 좀 더 편하게 사용할 수 있는 언어를 개발하는 데 초점을 맞추었다.

▣ 파이썬 언어의 유래 및 파이썬 언어의 특징

4. 파이썬의 필요성

- 가장 중요한 대답: "생산성이 높기 때문"
- 먼저 개발하라! 그리고 나서 성능을 개선하라.

- 파이썬을 활용하면 빠른 시간 내에 수준 높은 코딩을 할 수 있음
- 개발을 먼저 할 시 장점 : 협업 시 비교하여 성능을 높이기 용이함

■ 파이썬 언어의 유래 및 파이썬 언어의 특징

5. 파이썬의 특징

- 대화 기능의 인터프리터 언어
- 동적인 데이터 타입 결정 지원

- 대화 기능 → 마치 컴퓨터와 개발자가 대화하는 듯한 느낌

```
def add(a,b):  
    return a+b  
print add(1,2)  
print add('abc', 'def')  
print add([1, 2, 3], [4, 5, 6])
```

```
3  
abcdef  
[1, 2, 3, 4, 5, 6]
```

- def → add 함수를 만드는 키워드
- print → add 함수 수행 값을 반환한 것을 출력하는 것
- 문자열을 더하면 문자들이 합쳐짐
- 리스트 + 리스트 = 하나의 리스트
- 함수의 a와 b는 타입이 정해져 있지 X
- 타입은 실제로 값이 변수에 할당되는 순간에 정해짐

■ 파이썬 언어의 유래 및 파이썬 언어의 특징

5. 파이썬의 특징

- 플랫폼 독립적 언어
- 개발 기간 단축에 초점을 둔 언어
- 간단하고 쉬운 문법
- 고수준의 내장 객체 자료형 제공

```
a=[12, 'abcde', 4+2j, [3,4,5]]  
a.append('add')  
print a;
```

```
[12, 'abcde', (4+2j), [3, 4, 5], 'add']
```

- 윈도우에서 개발한 파이썬 소스코드를 맥과 리눅스에서 수행 가능
- C언어의 경우 소스코드를 컴파일을 다시 진행해야 함
- 이미 OS에서 인터프리터가 설치가 되어 있기 때문
- a 변수 → 리스트
- 리스트 : 서로 다른 객체들을 원소로 갖고, 순서가 부여된 자료구조형
- 리스트 : 첫 번째, 두 번째 인덱싱을 할 수 있음

▣ 파이썬 언어의 유래 및 파이썬 언어의 특징

5. 파이썬의 특징

- 메모리 자동 관리
 - 쉬운 유지 보수
 - 많은 수의 라이브러리 제공
 - 짧아지는 코드
 - 높은 확장성
-
- C언어 또는 Java → 100라인, 파이썬 → 5~10라인
 - 코딩 시 필요한 모듈 끌어와 사용 가능
 - 직접 만든 모듈도 다른 사람에게 제공 가능

▣ 파이썬 언어의 유래 및 파이썬 언어의 특징

6. 파이썬 활용처

- 시스템 유틸리티
- GUI
 - wxpython, tkinter
- 게임 프로그래밍
 - 파이썬 게임엔진: PyOpenGL PySDL PyGame Kivy PyOgre Panda3D Cocos2D PySoy
- 웹 프로그래밍
 - django 프레임워크
- 수치 프로그래밍
 - nextworkx 모듈
- 데이터베이스 프로그래밍
- 기타
- 참고: <http://mulgu.kr/73>

- 시스템 유틸리티: OS가 지원해야 하는 명령어
- 리눅스 → ls, Docs → dir, copy, move

▣ 파이썬 언어의 유래 및 파이썬 언어의 특징

7. 파이썬의 장점

- Guido가 생각했던 Python 문법적 특징은 들여쓰기를 철저하게 지키도록 언어를 설계했다는 점이다.
- 이는 코드의 가독성을 현격히 높여준다.
- C 언어에서처럼 {} 등의 괄호를 넣지 않기 때문에 프로그램을 좀더 깔끔하게 만들어준다.
- 파이썬 코드는 재사용하기가 쉽다.
- 코드의 분석이 쉽기 때문에 다른 사람이 작성한 코드를 받아서 작업하는 사람들이 훨씬 더 작업을 편하게 해준다.

- 파이썬은 들여쓰기를 안 할 경우 error 발생 → 들여쓰기 강제

파이썬 프로그래밍

파이썬 시작하기



한국기술교육대학교
온라인평생교육원

■ 파이썬 코딩 환경 구성

1. 파이썬 코딩 환경 구축

Python 전용 통합개발환경(IDE) 사용

- Enthought Canopy
 - 별도의 파이썬 설치 없이 아래 통합개발환경을 사용하면 자동으로 파이썬이 설치됨
 - CANOPY EXPRESS (Free) 다운로드 사이트:
<https://www.anaconda.com/distribution/>
- Python 실행
 - Enthought Canopy 직접 활용
 - 커맨드 창 (터미널)
 - ipython 명령어 이용

- ipython이 좀 더 대화환경을 깔끔하게 제공

▣ 파이썬 코딩 환경 구성

1. 파이썬 코딩 환경 구축

이클립스 사용

- 파이썬 설치
 - 파이썬 다운로드 사이트: <https://www.anaconda.com/distribution/>
- - 본 강좌에서는 파이썬 2.7.x 버전 사용
- 이클립스에서 파이썬 설치 방법
 - PyDev 플러그인 설치
 - 참고 사이트: <http://arisu1000.tistory.com/27707>

- 이번 과정은 2.7.8 버전 사용
- PyDev가 있어야 제대로 설치 된 것
- Quick Auto-Config 클릭 → 시스템 내 파이썬 인터프리터 검색

▣ 파이썬 코딩 환경 구성

1. 파이썬 코딩 환경 구축

추천하는 코딩 환경 구성 방법

- 1) Enthought Canopy 설치
 - 2) 이클립스 설치 (설치가 안되어 있는 경우)
 - 3) PyDev 플러그인 설치
-
- 두 가지 파이썬 코딩 환경을 합쳐 사용하는 것을 추천
 - 기본적인 파이썬 설치보다 Canopy 설치가 더 유리
 - 환경설정 관련된 것도 한꺼번에 설정 가능하고 도구 같이 설치 가능

■ 파이썬 코딩 환경 구성

2. 대화식 모드로 간단한 파이썬 예제 실행하기

```
print 4+5  
print "Hello World!"
```

```
9  
Hello World!
```

- workspace → 이클립스가 가지고 있는 파일들이 저장되는 기본 폴더
- 저장 → 윈도우: Ctrl + s, 맥: command + s

■ 파이썬 코딩 환경 구성

3. 파이썬 버전 알아보기

```
# -*- coding: utf-8 -*-  
print 4 + 5  
print 12 - 32  
print (4 + 5) * 6  
print 4 + 5 * 6  
print 9 / 5      # 정수/정수 연산 결과는 정수  
print 9.0 / 5.0  
print 9 / 5.0
```

```
9  
-20  
54  
34  
1  
1.8  
1.8
```

- # 마크 → 주석을 의미
- 한글을 코드 내 사용 시 맨 상단에 현재 소스코드의 인코딩 방식을 표기
- 현재 대부분 utf-8 방식 사용

■ 파이썬 코딩 환경 구성

4. 파일로 저장하여 실행하기

```
print 'Hello'  
print "Hi there!"
```

```
Hello  
Hi there!
```

```
a = 'My name is '  
b = 'Amenda'  
print a + b
```

```
My name is Amenda
```

▣ 파이썬 코딩 환경 구성

5. 모듈파일 실행하기

```
import sys
print sys.version
print
print sys.version_info
```

```
2.7.6 | 64-bit | (default, Jun  4 2014, 16:42:26)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)]
```

```
sys.version_info(major=2, minor=7, micro=6, releaselevel='final', serial=0)
```

▣ 파이썬 코딩 환경 구성

6. 파일로 저장하여 실행하기

- 다음 내용의 cal.py 만들기

```
# file: cal.py
import calendar
calendar.prmonth(2014, 11)
```

- 커맨트 창 (터미널 창)에서 해당 폴더로 이동 후 다음 명령어 수행 python.py
python cal.py

- prmonth = print month의 약자

▣ 파이썬 코딩 환경 구성

7. 대화식 모드에서 혹은 다른 모듈에서 모듈 파일 실행하기

```
execfile('/Users/yhhan/Documents/workspace/python/01/cal.py')
```

```
November 2014
Mo Tu We Th Fr Sa Su
      1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

- 파일 하나 = 모듈