

Basic Knowledge Python

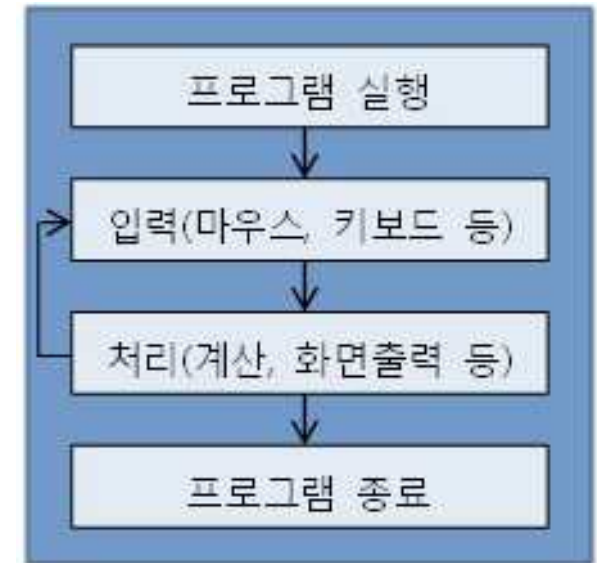
Computer Programming, 개발환경, IDE, Python이란

Computer Program

- 컴퓨터의 도입: 사람 vs. 컴퓨터
 - 컴퓨터가 복잡하고 어려운 문제를 빨리 풀 수 있음
 - 실수가 없음
 - 미사일 궤도 계산, 미/적분, 단순 반복 계산 등등
- 편의성, 효율성
 - 수기입력 -> DB 활용 -> 검색 등
 - 리포트 작성, Visualization
 - 스마트폰: App
 - 임베디드 시스템: 특정 목적만을 수행하는 프로그램이 들어가 있는 시스템
 - 스마트 냉장고, 세탁기, 자율주행 자동차 등
- 재미, 흥미
 - 게임


MATLAB, Octave, 수치
계산 Tool, 구구단
프로그램, etc.

Microsoft Office,
DBMS,
포토샵, 회원등록 프로그램, etc.



Coding vs. Programming

- Coding: 특정 알고리즘을 선택한 프로그래밍 언어 (Python, Java, etc.)로 변환하는 작업
 - 코더, **개발자**,.....

코딩 (coding) 

[명사]

1. 어떤 일의 자료나 대상에 대하여 기호를 부여하는 일.
2. [정보·통신] 작업의 흐름에 따라 프로그램 언어의 명령문을 써서 프로그램을 작성하는 일.
3. [정보·통신] 프로그램의 코드를 작성하는 일.

- Programming: 프로그램의 실행을 위하여 필요한 프로젝트 관리 등을 모두 포함한 행위
 - 프로그래머, 로직/알고리즘 설계, 아키텍터, Manager(관리자),..., **개발자**

명사

1. 정보·통신 **컴퓨터 프로그램을 작성**하는 일. 일반적으로는 **프로그램 작성 방법의 결정, 코딩(coding), 에러 수정** 따위의 작업을 이르지만 특수하게 코딩만을 이룰 때도 있다.

새로운것을 만드는 맛있는 사람들이 **개발자**

<https://tyson.tistory.com/99>

Computer Programming (wiki)

- 컴퓨터 프로그래밍 또는 간단히 프로그래밍(프로그램 작성) 혹은 코딩
 - 하나 이상의 관련된 추상 알고리즘을 특정한 프로그래밍 언어를 이용해 구체적인 컴퓨터 프로그램으로 구현하는 기술
- 컴퓨터 프로그램(computer program): 컴퓨터에서 실행될 때 특정 작업(specific task)을 수행하는 **일련의 명령어들의 모음(집합체)**
 - 시스템 프로그램: 컴퓨터 시스템과 하드웨어들을 제어 및 관리하는 프로그램
 - OS(Incl. Device Driver), 컴파일러, 링커 등
 - 응용 프로그램: 사용자가 원하는 기능을 제공하는 프로그램(지속적으로 컴퓨터의 리소스를 사용)
 - 워드, 엑셀, 포토샵, 게임, 등

Python, Java, C/C++ 등의 프로그래밍 언어라는 것을 이용하여 특정 프로그램을 작성

하고 Interpreter/Compiler를 통하여 컴퓨터가 처리할 수 있는 명령어들의 모음(기계어)으로 변경하여, 실행 파일을 만드는 것

Python (파이썬이란)

- 1990년 암스테르담의 귀도 반 로섬(Guido Van Rossum)이 개발한 **인터프리터 언어**
- 파이썬의 사전적 의미
 - 고대 신화에 나오는 파르나소스 산의 동굴에 살던 뱀
- 구글에서 만든 소프트웨어의 50% 이상이 파이썬으로 작성됨
- 드롭박스(Dropbox), 인스타그램(Instagram)
- 공동 작업과 유지 보수가 매우 쉽고 편함

세상의 속도를
따라잡고 싶다면

**Do
it!**

점프 투 파이썬

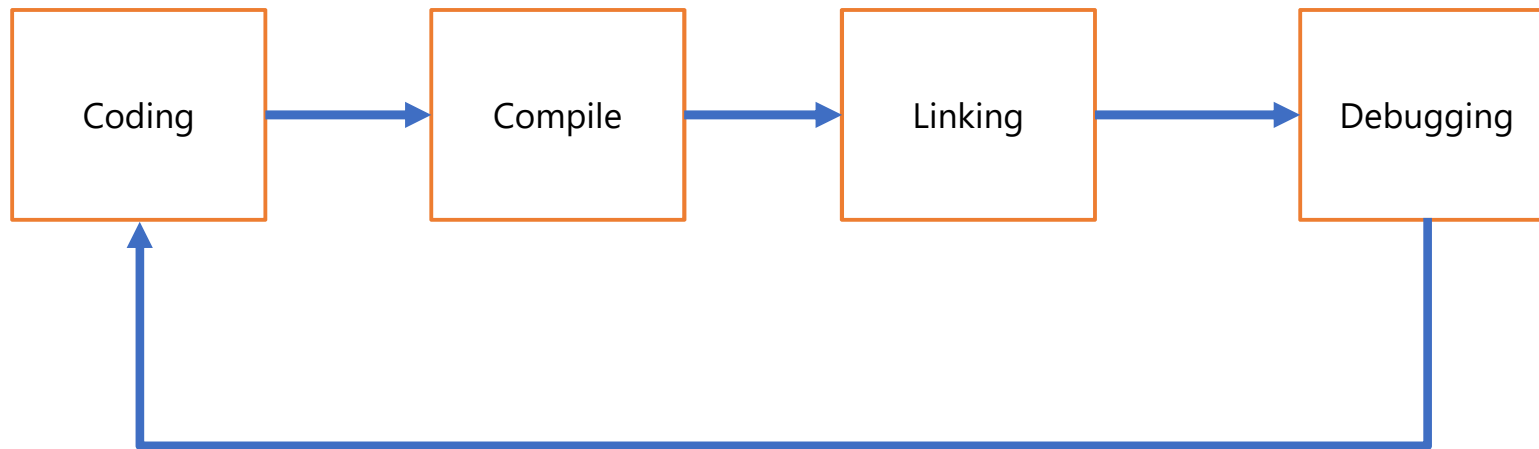
박응용 지음(위키독스 운영자)

교재



프로그램 작성 과정

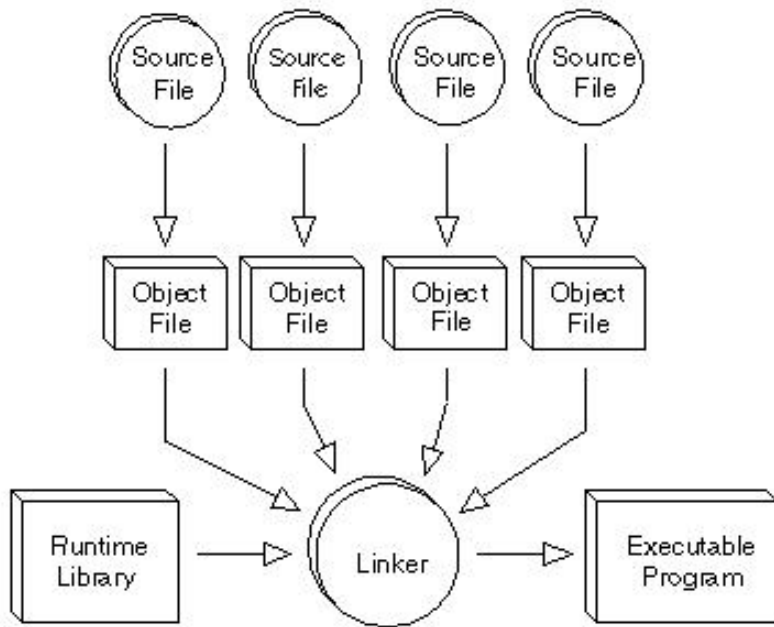
- Compiler: Compile + Linking
- Developer: Coding + Debugging (loop)



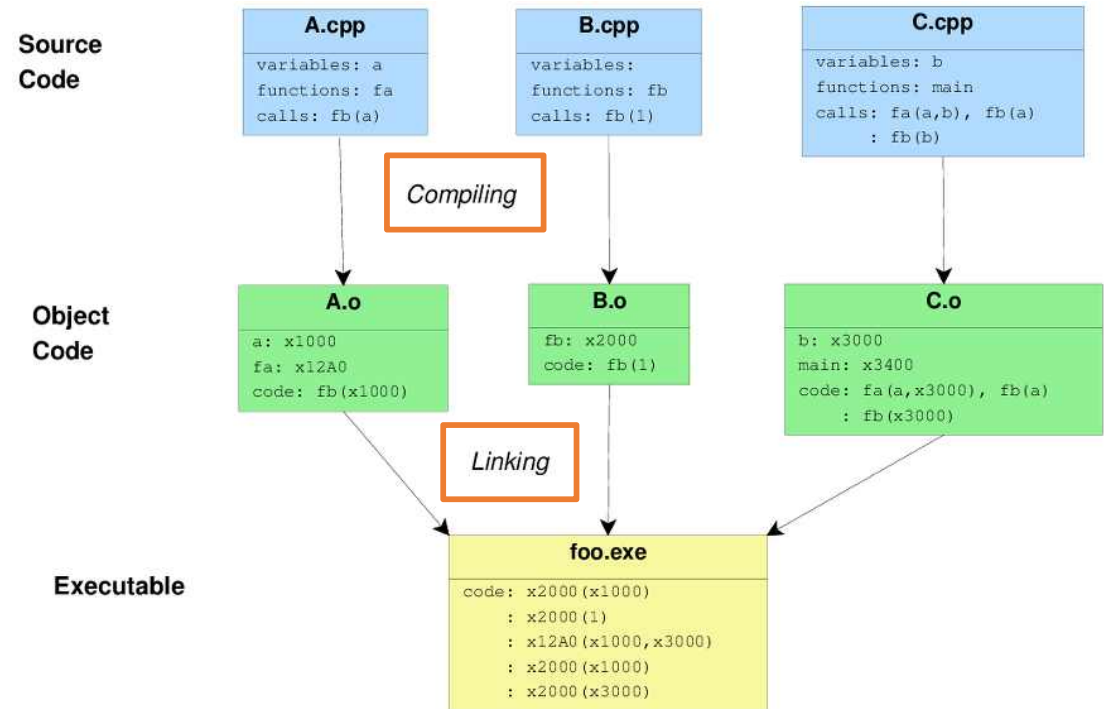
- Compile: 특정 프로그래밍 언어로 작성된 코드를 컴퓨터가 처리할 수 있는 기계어로 바꾸는 것
- Linking: 기계어로 된 부분을 실행파일로 바꾸는 것

Compiler (Overview)

- Compiler: Compile + Linking



출처: <https://www.webopedia.com/definitions/link/>



Interpreter(인터프리터: 번역기 등)

- 코드를 한 줄 씩 읽어 실행
 - 코드를 한 줄 씩 읽어 기계어로 번역(컴파일)하고 실행한 후, 결과 값을 출력
- R, Python, Ruby, perl

	Compiler	Interpreter
번역단위	프로그램 단위	명령 줄 단위
수행과정	전체 프로그램이 메모리에 올라간 후 수행	줄 단위로 번역 및 수행
목적 프로그램	필요	불필요
효율성	계속 사용 가능, 아키텍처가 변경되면 다시 컴파일 필요	실행 시 마다 번역, 인터프리터가 있을 경우, 별도로 컴파일과정 불필요
기억장소	많이 필요함	적게 필요함

Python 특징

- 파이썬은 인간다운 언어이다.
 - 파이썬은 사람이 생각하는 방식을 그대로 표현할 수 있는 언어

```
if 4 in [1, 2, 3, 4]: print("4가 있습니다")
```

=

만약 4가 1, 2, 3, 4 중에 있으면 '4가 있습니다'를 출력한다.

- 파이썬은 문법이 쉬워 빠르게 배울 수 있다
 - 문법 자체가 아주 쉽고 간결하며 사람의 사고 체계와 매우 닮아 있음
 - 유명한 프로그래머인 **에릭 레이먼드(Eric Raymond)**는 공부한 지 단 하루 만에 자신이 원하는 프로그램을 작성!

에릭 레이먼드

미국 소프트웨어 개발자



SW개발자/해커

Python 특징

- 파이썬은 무료이지만 강력하다
 - 오픈 소스 → 무료로 언제 어디서든 파이썬을 다운로드하여 사용 가능
 - 파이썬과 C는 찰떡 궁합
 - 프로그램의 전반적인 뼈대는 파이썬으로 만들고, 빠른 실행 속도가 필요한 부분은 C로 만들어서 파이썬 프로그램 안에 포함

```
from ctypes import cdll
libc = cdll.LoadLibrary('msvcrt.dll')
libc.printf(b'hello world!\n')

listTmp = [1, 2, 3, 4, 5, 6, 7]
print(f'{len(listTmp)}')

for idx in listTmp:
    libc.printf(b"%d ", idx)
```

```
C:\ProgramData\Anaconda3\python.exe D:/AI_programming_python/CPython.py
7
hello world!
1 2 3 4 5 6 7
Process finished with exit code 0
```

Python 특징

- 파이썬은 간결하다

펄(Perl)	파이썬
100가지 방법으로 하나의 일 처리	가장 좋은 방법 1가지만 사용

- 프로그램이 실행 되게 하려면
줄(들여쓰기)을 반드시 맞추어야 함
 - 가독성 ↑

```
# simple.py
languages = ['python', 'perl', 'c', 'java']

for lang in languages:
    if lang in ['python', 'perl']:
        print("%6s need interpreter" % lang)
    elif lang in ['c', 'java']:
        print("%6s need compiler" % lang)
    else:
        print("should not reach here")
```

Python 특징

- 파이썬은 간결하다 (C vs. python) <https://ideone.com/>

```
1  #include <stdio.h>
2
3  int main(void) {
4      char lang_0[10] = "python";
5      char lang_1[10] = "perl";
6      char lang_2[10] = "C";
7      char lang_3[10] = "java";
8
9      int idx = 0;
10     for(idx=0; idx<4; idx++)
11     {
12         if( (idx==0) && (lang_0[0]=='p') )
13         {
14             printf("%s need interpreter\n", &lang_0[0]);
15         }
16         else if( (idx==1) && (lang_1[0] == 'p') )
17         {
18             printf("%s need interpreter\n", &lang_1[0]);
19         }
20         else if( (idx==2) && (lang_2[0] == 'C') )
21         {
22             printf("%s need Compiler\n", &lang_2[0]);
23         }
24         else if ( (idx==3) && (lang_3[0] == 'j') )
25         {
26             printf("%s need Compiler\n", &lang_3[0]);
27         }
28         else
29         {
30             printf("maybe should not reach here\n");
31         }
32     }
33     return 0;
```

simple.py

```
languages = ['python', 'perl', 'c', 'java']
```

```
for lang in languages:
```

```
    if lang in ['python', 'perl']:
```

```
        print("%6s need interpreter" % lang)
```

```
    elif lang in ['c', 'java']:
```

```
        print("%6s need compiler" % lang)
```

```
    else:
```

```
        print("should not reach here")
```

Python 특징

- 파이썬은 간결하다
 - 프로그램이 실행 되게 하려면 줄(들여쓰기)을 반드시 맞추어야 함
 - 가독성 ↑

```
#include <stdio.h>
int main(void) {char lang_0[10] = "python";char lang_1[10] = "perl";char lang_2[10] = "C";
char lang_3[10] = "java";
int idx = 0;for(idx=0; idx<4; idx++)
{if( (idx==0) && (lang_0[0]=='p') )
{
printf("%6s need interpreter\n", &lang_0[0]);
}
else if( (idx==1) && (lang_1[0] == 'p') )
{printf("%6s need interpreter\n", &lang_1[0]);
}
else if( (idx==2) && (lang_2[0] == 'C') )
{
printf("%6s need Compiler\n", &lang_2[0]);
}else if ( (idx==3) && (lang_3[0] == 'j') )
{
printf("%6s need Compiler\n", &lang_3[0]);}else
{
printf("maybe should not reach here\n");}
}
return 0;}
```

```
▶ for lang in language:
if lang in ['python', 'perl']:
    print("%6s need interpreter" % lang)
elif lang in ['C', 'java']:
    print("%6s need compiler" % lang)
else:
    print("maybe should not reach hear?")
```

```
❏ File "<ipython-input-12-e585de529c3d>", line 2
    if lang in ['python', 'perl']:
        ^
```

IndentationError: expected an indented block

Python 특징

- 파이썬은 개발 속도가 빠르다

“Life is too short, You need Python.”
인생은 너무 짧으니 파이썬이 필요해.

- 파이썬의 빠른 개발 속도를 두고 유행처럼 퍼진 말

에릭 레이먼드

미국 소프트웨어 개발자



Python으로 할 수 있는 일

- 시스템 유틸리티 제작
 - 운영체제(윈도우, 리눅스 등)의 시스템 명령어를 사용하는 도구를 통한 시스템 유틸리티 제작
- GUI(Graphic User Interface) 프로그래밍
 - 화면에 윈도우 창을 만들고 프로그램을 동작시킬 수 있는 메뉴나 버튼, 그림 등을 추가하는 것
 - GUI 프로그래밍을 위한 도구들을 갖추고 있어, GUI 프로그램을 만들기 쉬움 (예 - Tkinter(티케이인터))
- C/C++와의 결합
 - C나 C++로 만든 프로그램을 파이썬에서, 파이썬으로 만든 프로그램을 C나 C++에서 사용 가능
- 웹 프로그래밍
- 수치 연산 프로그래밍
 - C로 작성된 수치 연산 모듈 NumPy를 통해 빠른 수치 연산 가능
- 데이터베이스 프로그래밍
 - Sybase, Infomix, Oracle, MySQL, PostgreSQL 등의 데이터에 접근하기 위한 도구 제공
 - 자료를 변형 없이 그대로 파일에 저장하고 불러오는 파이썬 모듈 피클(pickle)
- 데이터분석, 사물 인터넷
 - 판다스(Pandas) 모듈을 통한 데이터 분석, 라즈베리파이를 제어하는 도구를 통한 사물 인터넷 구현

Python으로 할 수 없는 일

- 시스템과 밀접한 프로그래밍 영역
 - 운영체제, 엄청난 횟수의 반복과 연산이 필요한 프로그램, 데이터 압축 알고리즘 개발 프로그램 등 대단히 빠른 속도를 요구하거나 하드웨어를 직접 건드려야 하는 프로그램에는 어울리지 않음
- 모바일 프로그래밍
 - 안드로이드 앱(App)을 개발하는 것은 아직 어려움

개발 환경

- 개발 환경 구축

- 소스코드를 작성하고, 기계어로 번역하여 작성된 프로그램을 바로 실행시켜 볼 수 있는 환경을 구축하는 것.

➔ IDE (Integrated Development Environment, 통합 개발 환경)

통합 개발 환경(Integrated Development Environment, IDE)이란 공통된 개발자 툴을 하나의 그래픽 사용자 인터페이스(Graphical User Interface, GUI)로 결합하는 애플리케이션을 구축하기 위한 소프트웨어입니다. 일반적으로 IDE는 다음과 같은 요소로 구성되어 있습니다.

- **소스 코드 편집기**: 시각적 신호를 활용한 구문(Syntax) 강조와 같은 기능을 포함하여 소프트웨어 코드를 작성하도록 돕는 텍스트 편집기로서, 언어별 자동 완성 기능과 코드 작성 중 버그 검사를 제공합니다.
- **로컬 빌드 자동화**: 컴퓨터 소스 코드를 바이너리 코드로 컴파일링하고, 바이너리 코드를 패키징하고 자동화 테스트를 실행하는 등 간편하고 반복 가능한 태스크를 개발자가 사용하는 소프트웨어의 로컬 빌드를 생성할 때 자동화해주는 유틸리티입니다.
- **디버거**: 원본 코드에 있는 버그 위치를 그래픽으로 표시할 수 있는 다른 프로그램을 테스트하는 프로그램입니다.

개발 환경

- 고려사항
 - 내가 개발 하고자 하는 시스템이 무엇인가?
 - 웹 어플리케이션 (DBMS, 웹 개발 IDE, etc.)
 - 범용 CPU(Intel/AMD CPU, etc.) 에서만 운영되는 시스템인가?
 - 범용 CPU가 아닌 ARM 계열의 CPU일 경우는 어떻게 해야 되는가?
 - 크로스 컴파일 환경 구축 필요
 - 버전에 민감한 시스템인가? (딥러닝 학습 프레임워크에 사용되는 python 버전 등)
 - Anaconda 등을 이용하여 Virtual Environment(가상 환경) 생성 가능

“ 개발환경을 잘 만드는 것만으로도 개발의 반은 진행한 것이다.”

Thank you

Q&A

www.kopo.ac.kr
jsshin7@kopo.ac.kr