
업무자동화를 위한
파이썬 기초

양선주 강사

강의 소개

- 프로그래밍 언어 파이썬의 기초 문법과 함수의 사용법을 이해하고, 다양한 문제 해결을 위한 알고리즘을 코드로 구현하는 수업입니다.
- 코딩을 처음 해 보는 분도 쉽게 따라할 수 있습니다.

강사 소개

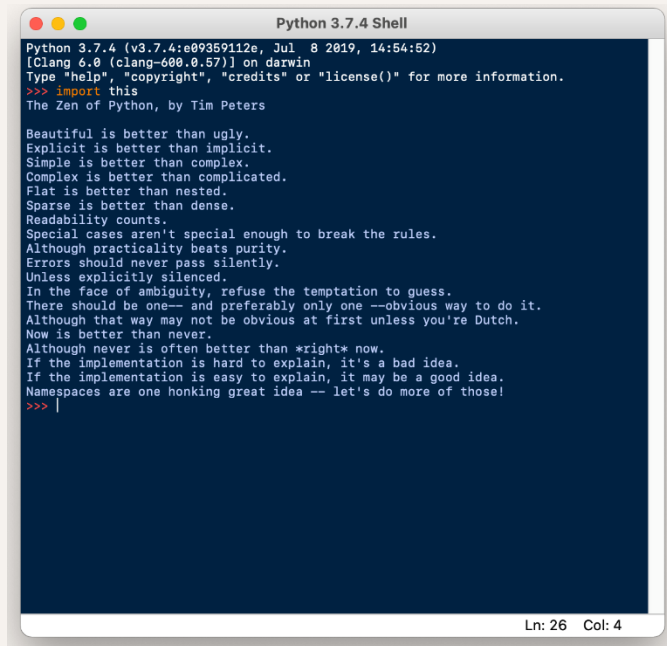
- (주) 유미시스 개발팀
- (주) 키즈플랜 SW 코딩강사
- SW 미래채움 인천관내 SW 교육
- 방과후 SW 교육강사 (인천 예일고등학교 / 인천 도림고등학교 등)
- 충주북여중, 인천 연송고, AI 특강 교육안 개발
- Cos Pro 파이썬 1급 / SW 코딩지도사 1급 / 정보처리기사

강의 계획

차시	강의내용
1차시	파이썬 소개와 개발환경 준비하기
2차시	파이썬 기본문법과 입출력 함수 활용하기
3차시	자료형과 변수 이해하기
4차시	반복과 조건문을 활용하여 프로그램 개발하기
5차시	객체지향의 이해와 함수의 활용
6차시	모듈의 개념을 이해하고 활용하기

사용 프로그램 및 플랫폼

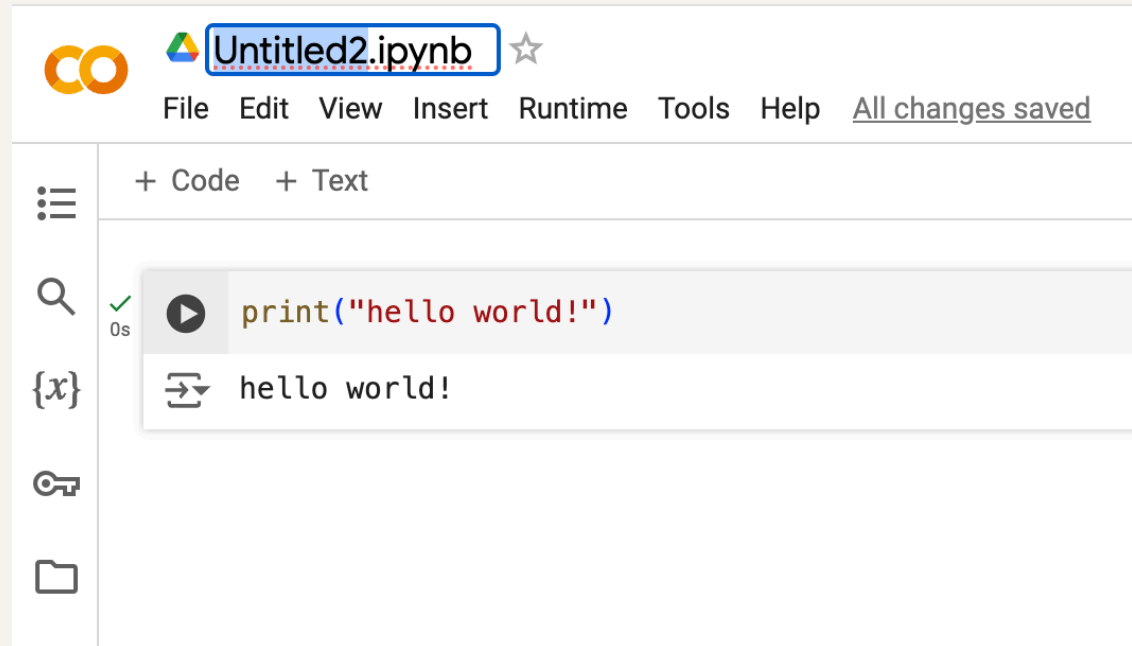
- 파이썬 Idle, google colab. 등




```
Python 3.7.4 Shell
Python 3.7.4 (v3.7.4:00359112e, Jul 8 2019, 14:54:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>> |
```


Ln: 26 Col: 4




CO  **Untitled2.ipynb** ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

✓ 0s  `print("hello world!")`

 hello world!

기대효과

- 프로그래밍을 통해 반복적 업무를 자동화 하여 빠르고 쉽게 해결할 수 있는 능력을 키웁니다.
- 인공지능과 데이터 활용을 위한 기초 프로그래밍 능력을 얻을 수 있습니다.