

코딩을 시작하거나 AI로 코드를 생성하세요.

✓ 파이썬이란

- 1990년 암스테르담의 귀도 반 로섬(Guido Van Rossum)이 개발한 인터프리터 언어
- 컴퓨터 프로그래밍 교육을 위해 많이 사용하지만, 기업의 실무를 위해서도 많이 사용하는 언어. 구글에서 만든 소프트웨어의 50% 이상이 파이썬으로 작성

파이썬의 특징

- 파이썬은 인간다운 언어이다
- 파이썬은 문법이 쉬워 빠르게 배울 수 있다
- 파이썬은 무료이지만 강력하다
- 파이썬은 간결하다
- 파이썬은 프로그래밍을 즐기게 해준다
- 파이썬은 개발 속도가 빠르다

파이썬으로 할 수 있는 일

- 웹 개발: Django, Flask 등의 프레임워크를 사용하여 웹 애플리케이션을 개발할 수 있습니다.
- 데이터 분석: Pandas, NumPy, SciPy와 같은 라이브러리를 사용하여 데이터를 분석하고 처리할 수 있습니다.
- 머신 러닝과 인공지능: TensorFlow, PyTorch, Scikit-learn 등의 라이브러리를 활용하여 머신 러닝 모델을 구축하고 훈련시킬 수 있습니다.
- 자동화: 파이썬 스크립트를 작성하여 일상적인 작업을 자동화하고, 시스템 관리 작업을 수행할 수 있습니다.
- 게임 개발: Pygame과 같은 라이브러리를 사용하여 간단한 게임을 개발할 수 있습니다.
- 모바일 애플리케이션 개발: Kivy 또는 BeeWare와 같은 라이브러리를 사용하여 모바일 애플리케이션을 개발할 수 있습니다.
- 데스크탑 애플리케이션 개발: PyQt, Tkinter 등의 라이브러리를 활용하여 데스크탑 애플리케이션을 개발할 수 있습니다.
- 시스템 스크립팅과 네트워킹: 시스템 유틸리티를 개발하거나 네트워크 프로토콜을 구현할 수 있습니다.
- 임베디드 시스템과 하드웨어 제어: 라즈베리 파이와 같은 임베디드 시스템을 제어하고 하드웨어를 프로그래밍할 수 있습니다.
- 사이언티픽 컴퓨팅: 과학적 연산과 시뮬레이션을 위해 파이썬을 활용할 수 있습니다.
- 교육: 파이썬은 초보자에게 프로그래밍을 가르치는 데 이상적인 언어로 평가받고 있습니다.
- 파이썬의 다양한 라이브러리와 프레임워크 덕분에, 이러한 분야에서의 작업이 더욱 쉽고 효율적으로 수행될 수 있습니다.

```
# ! 기호는 colab 셀에서 Unix, Linux 셸 명령어 실행시 필요
!python --version
```

```
Python 3.10.12
```

```
#현재 작업 디렉토리 표시, %는 IPython 환경 (col.포함)에서 제공하는 매직 명령어 실행 시 필요
# # 하나는 단일 라인에 대해 실행, %%는 셀 전체에 적용하고자 할 때 사용
%pwd
```

```
'/content'
```

더블클릭 또는 Enter 키를 눌러 수정

```
%time
sum=0
for i in range(1000000):
    sum += i

print(sum)

499999500000
CPU times: user 140 ms, sys: 0 ns, total: 140 ms
Wall time: 142 ms
```

✓ 용어

- 식별자 : 프로그래밍 언어에서 이름을 붙일 때 사용하는 단어. 주로 변수 또는 함수 이름 등으로 사용
- 주석 : 프로그램을 설명하기 위해 사용. # 기호로 주석 처리
- 연산자 : 스스로 값이 되는 것이 아니고 값과 값 사이에 무언가 기능을 적용할 때 사용
- 자료 : 리터럴이라고 하는데 숫자이든 문자이든 어떠한 값 자체를 의미. 1, 10, "Hello"
- 키워드 : 파이썬이 만들어질 때 이미 사용하겠다고 예약해 놓는 것. False, None, True, ...
- 프로그래밍 언어에서 사용자가 이름을 정할 때 키워드는 사용할 수 없음

식별자

count, user_name, _is_valid, calculate_area, Car, model, year, math 및 m 모두 유효한 식별자. 각각의 식별자는 특정한 데이터 또는 기능에 이름을 부여하여 코드 내에서 해당 데이터나 기능을 참조할 수 있게하며 코드의 가독성과 유지 보수성을 높이는 데 중요한 역할

```
# 변수 식별자
count = 0
user_name = "Alice"
_is_valid = True

# 함수 식별자
def calculate_area(radius):
    return 3.14159*radius*radius

# 클래스 식별자
class Car:
    def __init__(self, model, year):
        self.model = model
        self.year = year

# 모듈 식별자
import math as m
```

✓ 식별자 기본 규칙

- 키워드를 사용하면 안된다
- 특수문자는 언더 바 (_)만 사용
- 숫자로 시작하면 안된다
- 공백은 포함할 수 없다

```
import keyword
print(keyword.kwlist)
print("WnWt", len(keyword.kwlist))
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally',
```

```
35
```

```
alpha
break # 키워드
alpha10
_alpha
273alpha # 숫자로 시작
Alpha
ALPHA
has space # 공백
```

```
File "<ipython-input-52-3cf020c5f4ff>", line 5
```

```
273alpha # 숫자로 시작
```

```
^
```

```
SyntaxError: invalid decimal literal
```

Q. 주어진 문자열 리스트에서 유효한 파이썬 변수 이름만을 추출하여 반환하는 함수를 작성하시오

```
identifies=["var1", "2things", "variable_name", "time!", "True"]
```

```
# 키워드 제거
```

```
import keyword
```

```
for iden in identifies:
```

```
    if iden in keyword.kwlist:
```

```
        identifies.remove(iden)
```

```
# 숫자 prefix 제거
```

```
for iden in identifies:
```

```
    if iden[0].isnumeric():
```

```
        identifies.remove(iden)
```

```
print(identifies)
```

```
# 특수 문자 제거
```

```
import string
```

```
for iden in identifies:
```

```
    if iden in string.punctuation:
```

```
        identifies.remove(iden)
```

```
    if "!" in iden:
```

```
        identifies.remove(iden)
```

```
print(identifies)
```

```
['var1', 'variable_name', 'time!']
```

```
['var1', 'variable_name']
```