

변수와 판다스 API 활용



파이썬 변수개념 할머니에게 설명하듯이, 이 변수에서 판다스와 api연결, 파일관리에 꼭 필요한 개념 알려줘

좋아요! 😊

파이썬의 **변수 개념**을 "컴퓨터를 잘 모르는 할머니께" 설명드리듯이, 아주 쉽게 알려드릴게요. 그리고 이 변수가 **판다스 (Pandas), API 연결, 파일 관리**에 왜 중요한지도 같이 설명드릴게요.

🧠 변수란?

"변수는 이름표다!"

할머니, 예를 들어 부엌에 김치를 담은 통이 있잖아요?

그 통에 '김치' 라고 이름표를 붙이면, 나중에 딱 보면 "아, 이 통은 김치가 들어있구나!" 하고 알 수 있죠.

📦 컴퓨터도 마찬가지예요.

어떤 값(예: 숫자, 글자, 데이터...)을 저장해두고 **이름표(=변수)** 를 붙이면, 나중에 쉽게 꺼내 쓸 수 있어요.

python

```
kimchi = "배추김치"
print(kimchi)
```

이러면 **kimchi** 라는 변수에 "배추김치"라는 내용이 담겨 있고, 나중에 이걸 다시 꺼내 볼 수 있어요!

📌 변수는 왜 중요할까요?

우리가 이름을 붙여놔야 나중에 쉽게 불러다 쓸 수 있으니까요!

🛒 [1] 판다스(Pandas)에서 변수는 이렇게 중요해요!

판다스는 표(엑셀처럼 생긴 데이터) 를 다루는 도구예요.

python

```
import pandas as pd
my_data = pd.read_csv("가계부.csv")
```

📝 여기서 **my_data**가 바로 변수예요.

이 안에 표(데이터)가 들어 있고,

우리는 계속 이 이름(**my_data**)을 써서 데이터를 분석하거나 바꿔요!

예:

python

```
my_data.head() # 표의 위쪽 몇 줄 보여줘!
```


[2] API 연결에서도 변수는 필수예요!

API는 쉽게 말하면 인터넷으로 정보 가져오는 우체통 같은 거예요.

python

```
url = "https://example.com/data"
response = requests.get(url)
data = response.json()
```

- **url** : 주소를 담은 변수
- **response** : 가져온 편지(정보)
- **data** : 그 안에 담긴 실제 내용

 이렇게 이름 붙여야, 나중에 필요한 정보만 쓱쓱 꺼낼 수 있어요.

[3] 파일 저장/불러오기에서도 변수는 이렇게 쓰여요!

python

```
file_name = "가계부.csv"
data = pd.read_csv(file_name)
```

- **file_name** 에는 파일 이름이 들어가 있어요.
- 나중에 이 이름으로 파일을 불러오거나 저장할 수 있어요!

이걸 안 쓰면 매번 "가계부.csv" 라는 이름을 다시 적어야 해서 너무 번거로워요.

꼭 기억해야 할 핵심 정리!

개념	쉽게 말하면	실제 예시
변수	이름표를 붙여 저장하는 그릇	<code>kimchi = "배추김치"</code>
판다스에서	표를 담는 이름	<code>data = pd.read_csv(...)</code>
API 연결	가져온 정보를 담는 상자	<code>response = requests.get()</code>
파일 처리	파일 이름, 내용 등을 저장하는 그릇	<code>file_name = "test.csv"</code>

마무리

할머니, 변수는 "이름표를 붙여놓고 나중에 편하게 부르기 위한 것"이에요.
요리할 때 양념통에 이름 붙이는 것과 똑같죠!
그래서 파이썬에서 변수는 아주 기본이지만, 엄청 중요하답니다 😊

필요하면 각 상황에 맞는 예제도 하나하나 실습으로 도와드릴게요!
다음은 어떤 걸 더 배우고 싶으세요?



변수작성법및 예약어를 사용했을때 에러도 알려줘

좋아요! 😊

이번엔 파이썬에서 **변수 작성법**과, **예약어**를 변수로 쓰면 생기는 에러를 아주 쉽게, 할머니께 설명드리듯 차근차근 알려드릴게요.

👤 1. 변수 작성법 (이름 붙이는 규칙)

컴퓨터도 사람처럼 물건(값)을 저장할 때 **이름을 붙여줘야** 나중에 꺼낼 수 있어요.
근데 이 이름에도 **규칙**이 있어요!

✅ 변수 이름 붙일 때 지켜야 할 규칙

규칙	설명	예시
1. 숫자로 시작 ❌	변수는 숫자로 시작하면 안 돼요	1apple = "사과" ❌ 에러 발생!
2. 영어, 숫자, 밑줄(_)만 가능	특수기호, 공백은 안돼요	my name = "할머니" ❌
3. 띄어쓰기 ❌	띄어쓰기 안돼요	my name = "철수" ❌
4. 예약어 사용 금지	예약어는 파이썬이 이미 쓰고 있는 단어예요	if = 10 ❌ 에러 발생

👉 2. 변수 작성법 예시 (올바른 것)

python

```
name = "김철수"
age = 80
kimchi_jar = "총각김치"
total3 = 100
_mydata = "비밀데이터"
```

✅ 이런 건 전부 괜찮아요!

❌ 3. 예약어를 변수로 쓰면 생기는 에러

예약어(reserved words)는 파이썬이 이미 중요한 용도로 쓰고 있어서
변수 이름으로 쓰면 안 돼요!

예를 들어:

```
python

if = 10
```

✗ 이런 코드는 에러가 나요:

```
javascript

SyntaxError: invalid syntax
```

🔒 대표적인 예약어 10개만 기억해봐요

예약어	뜻	쓸 수 없음
if	만약에	조건문
else	그렇지 않으면	조건문
for	반복	반복문
while	~동안	반복문
def	함수 정의	함수 만들기
class	클래스 정의	객체 만들기
return	결과 내보내기	함수 안에서 사용
import	라이브러리 불러오기	import pandas 처럼
from	가져오기	모듈 불러올 때
try	에러 처리 시작	에러 예방

전체 예약어는 **keyword** 라는 모듈을 쓰면 다 확인할 수 있어요.

```
python

import keyword
print(keyword.kwlist)
```

💡 변수 이름 실수 예시와 고치는 법

실수 코드	에러 메시지	올바른 코드
for = 5	SyntaxError	count = 5
3name = "할머니"	SyntaxError	name3 = "할머니"
my-name = "김"	SyntaxError	my_name = "김"
my name = "김"	SyntaxError	my_name = "김"

👉 요약

- 변수 = 값에 붙이는 이름표

- 영어, 숫자, _ 로만 만들어야 해요
- 숫자로 시작 ❌, 띄어쓰기 ❌, 예약어 ❌
- 예약어는 파이썬이 미리 쓰는 말이라 변수로 못 써요
- 예약어를 변수로 쓰면 **SyntaxError** 가 나요

필요하다면, 간단한 연습 퀴즈도 만들어드릴게요! 😊

"이 변수 이름은 맞을까요? 틀릴까요?" 같은 문제도 재미있어요.

다음으로는 변수랑 같이 자주 쓰이는 **자료형**도 설명해드릴까요? (예: 숫자, 글자, 리스트 등)



예약어를 사용하면 나오는 에러는

좋은 질문이에요! 😊

파이썬에서 **예약어(reserved word)** 를 변수 이름으로 쓰려고 하면, **SyntaxError** 라는 에러가 발생해요.

이 에러는 **문법적으로 잘못된 코드**라는 뜻이에요.



예제: 예약어를 변수로 사용했을 때

```
python
```

```
if = 10
```

🔥 결과:

```
arduino
```

```
File "<stdin>", line 1
  if = 10
    ^
```

```
SyntaxError: invalid syntax
```

- **SyntaxError**: 문법에 맞지 않는 코드를 썼다는 뜻이에요.
- **invalid syntax**: 문법이 잘못되었어요.

🚫 다른 예약어 예제들

1. for

```
python
```

```
for = 5
```

```
javascript
```

```
SyntaxError: invalid syntax
```

2. class

```
python
```

```
class = "hello"
```

```
javascript
```

```
SyntaxError: invalid syntax
```

3. import

```
python
```

```
import = "데이터"
```

```
javascript
```

```
SyntaxError: invalid syntax
```

에러를 쉽게 기억하는 방법

실수 코드	에러 메시지	왜 에러일까?
<code>if = 3</code>	<code>SyntaxError</code>	<code>if</code> 는 조건문
<code>for = 5</code>	<code>SyntaxError</code>	<code>for</code> 는 반복문
<code>class = "A"</code>	<code>SyntaxError</code>	<code>class</code> 는 객체 만들기
<code>return = 1</code>	<code>SyntaxError</code>	<code>return</code> 은 함수 결과 반환
<code>import = "데이터"</code>	<code>SyntaxError</code>	<code>import</code> 는 라이브러리 불러오기

예약어를 피해서 변수 만드는 팁

- 예약어가 헛갈릴 땐 이름 뒤에 `_` 를 붙여도 괜찮아요!

```
python
```

```
if_ = 3 # 👍 가능
```

```
class_ = "반장" # 👍 가능
```

💡 보너스: 예약어 전체 목록 확인 방법

```
python
```

```
import keyword
print(keyword.kwlist)
```

결과:

```
python
```

```
['False', 'None', 'True', 'and', 'as', 'assert', ..., 'yield']
```

필요하시면, 예약어 퀴즈도 만들어 드릴게요!

예: "이건 변수로 쓸 수 있을까요? 0/X"

또는 다음 단계로 자료형이나 리스트 공부로 넘어가도 좋아요! 😊



내장함수 리스트는 어떻게 확인. 이 내장함수도 변수로 사용하면 안되지

정확해요! 😊

내장함수(built-in function) 도 변수 이름으로 사용하면 안 되는 이유는, 파이썬이 이미 그 이름을 **중요한 기능으로 사용 중**이기 때문이에요.

🔍 1. 내장함수(built-in function)란?

파이썬이 기본적으로 제공하는 도구함이에요.
따로 설치하지 않아도 바로 쓸 수 있는 함수들이죠!

예:

- **print()**: 출력
- **len()**: 길이
- **sum()**: 합계
- **type()**: 타입 확인
- **input()**: 입력 받기

📖 2. 내장함수 목록 확인 방법

```
python
```

```
import builtins
print(dir(builtins))
```

이렇게 하면 쪽~~ 내장함수와 상수, 예외 클래스까지 다 나와요.

📌 대표적인 내장함수 몇 가지

함수명	설명
<code>print()</code>	화면에 출력
<code>input()</code>	사용자 입력
<code>len()</code>	길이 구하기
<code>sum()</code>	합계 구하기
<code>max() / min()</code>	최대값 / 최소값
<code>type()</code>	데이터 타입 확인
<code>int() / str() / float()</code>	형 변환

🔪 3. 내장함수를 변수로 쓰면 생기는 문제

에러는 안 나지만, 내장함수 기능이 덮어쓰기(재정의) 돼서 나중에 문제가 생겨요! ⚠️

예시:

```
python

sum = 10
print(sum([1, 2, 3]))
```

결과:

```
php

TypeError: 'int' object is not callable
```

이유:

- `sum`이라는 이름에 숫자 `10`을 넣었기 때문에
- `sum()` 함수가 더 이상 함수가 아니고 숫자가 되어버린 거예요.
- 그래서 `sum([...])` 을 하면 "10은 함수가 아닌데 왜 ()를 써?" 라는 오류가 나요.

✅ 해결 방법: 내장함수 이름은 변수로 쓰지 말기

아래 이름은 변수로 피해야 해요!

```
python

sum, len, max, min, input, print, str, int, float, list, dict, set, tuple, type, open, range
```


💡 실수한 경우 해결법

python

```
sum = 10
del sum # sum이라는 이름을 지워서 원래 함수 복원
```

👉 정리

구분	변수로 쓰면?	결과
예약어	❌ 문법 에러	SyntaxError 발생
내장함수	❌ 덮어쓰기(재정의)	TypeError , 버그 위험

필요하시면,

- 내장함수 퀴즈,
- 또는 헛갈리는 함수 리스트 정리도 도와드릴게요!

다음으로는 "자료형", "리스트", "조건문" 중 어디로 가볼까요? 😊



파이썬의 변수와 다른 프로그램의 변수 차이는

아주 좋은 질문이에요! 😊

"파이썬의 변수와 다른 언어의 변수는 무엇이 다를까?"

프로그래밍을 본격적으로 배우기 시작할 때 꼭 짚고 넘어가야 할 핵심이에요.

🧠 요약 먼저!

항목	파이썬 변수	C, Java 등 다른 언어 변수
자료형 선언	자동(동적)	직접 지정(정적)
변수 생성	x = 10	int x = 10;
유형 변경	가능 (x = 10 → x = "문자")	불가능 (자료형 고정)
메모리 관리	자동 (가비지 컬렉션)	명시적 할당/해제 필요 (C계열)
변수는 실제로?	"이름표" (객체에 대한 참조)	대부분은 "값을 담는 상자" 개념

🧩 1. 자료형 선언 방식의 차이

🐍 파이썬: 동적 타입(dynamic typing)

변수를 만들 때 **자료형을 따로 지정하지 않아요.**
파이썬이 자동으로 판단해요!

```
python

x = 10          # 정수
x = "hello"     # 문자로 바뀌어도 OK
```

C, Java: 정적 타입(static typing)

변수를 만들 때 **자료형을 꼭 적어야 해요.**

```
c

int x = 10;      // 정수형 변수
x = "hello";     // ❌ 에러! (정수형에 문자는 안돼요)
```

2. 변수의 본질: “값” vs “객체 참조”

파이썬: 변수는 값이 아닌 객체를 가리키는 이름표

```
python

a = [1, 2, 3]
b = a          # b는 a와 같은 리스트를 가리킴

b[0] = 99
print(a)       # [99, 2, 3] ← a도 바뀜!
```

C, Java 등: 변수는 값 자체를 저장하거나 복사

```
c

int a = 5;
int b = a;     // b는 a의 "값"만 복사

b = 10;
printf("%d", a); // 5 ← a는 그대로!
```

3. 메모리 관리 차이

- 파이썬: 변수는 쓰지 않으면 자동으로 버려요 (가비지 컬렉션)
- C: `malloc()` 으로 메모리 할당, `free()` 로 해제해야 해요

쉽게 비유하면

개념	파이썬	C/Java
변수란	"택배 송장" (주소를 가리킴)	"물건이 직접 담긴 상자"
변수 변경	송장이 다른 상자에 붙는 느낌	상자 안 내용물을 바꾸는 느낌

✅ 마무리 요약

항목	파이썬	C/Java
자료형 지정	❌ 필요 없음 (자동)	✅ 반드시 지정 (int , string)
타입 변경	자유롭게 가능 ($x = 1 \rightarrow x = "a"$)	에러 발생
변수 역할	객체를 참조하는 이름표	값을 직접 담는 상자
메모리 관리	자동 관리	수동 관리 또는 제한적 자동 관리

📁 보너스: 타입 힌트를 쓰면 파이썬도 약간 정적처럼 사용 가능해요

```
python
```

```
name: str = "철수"
age: int = 10
```

하지만 검사만 하고 강제하지 않아요. (mypy 같은 도구를 써야 강제됨)

필요하다면 "파이썬 변수 참조와 복사"나 "리스트 변수와 깊은 복사(deepcopy)"도 더 알려드릴게요!
다음으로는 자료형, 리스트, 함수 쪽으로 넘어가볼까요? 😊



장단점은

좋습니다! 😊

이번에는 파이썬 변수의 특징과 다른 언어(C, Java 등)의 변수와 비교한 장단점을 정리하고 설명해드릴게요.

🧠 핵심 요약

구분	파이썬 변수 (동적, 참조형)	다른 언어 변수 (정적, 값 저장형)
장점	유연하고 빠르게 코딩 가능	자료형이 명확해서 안정성 높음
단점	실수로 타입 섞이면 오류 나기 쉬움	자료형 일일이 지정, 코드가 길어짐

● 파이썬 변수의 장점

장점	설명	예시
✅ 코드 간결함	자료형 선언 안 해도 돼서 빨리 작성 가능	<code>x = 10, x = "hi"</code>
✅ 유연성	변수에 어떤 값이든 할당 가능	<code>x = [1,2,3] → x = True</code>
✅ 개발 속도 빠름	초보자나 스크립트 작성에 매우 유리	빠르게 테스트 가능
✅ 객체 참조로 메모리 효율적	큰 객체는 복사 안 하고 이름만 공유	<code>a = [1,2]; b = a</code>

● 파이썬 변수의 단점

단점	설명	예시
⚠️ 자료형 오류에 취약	실수로 문자열에 숫자 더함 → 예러	<code>"5" + 3 → TypeError</code>
⚠️ 오타도 조용히 통과	<code>value = 10</code> (오타)도 문제 없이 실행됨 → 추적 어려움	
⚠️ 큰 프로젝트에서 유지보수 어려움	변수에 어떤 타입이 들어있는지 모르면 디버깅이 힘들	

🔧 C, Java 변수의 장점

장점	설명
✅ 자료형이 명확	변수의 타입이 고정되어 오류 가능성 낮음
✅ 컴파일러가 오류 미리 잡아줌	<code>"hello" + 3</code> 같은 코드는 컴파일 단계에서 막음
✅ 대규모 프로젝트에 유리	협업할 때 변수 구조가 명확하니 유지보수 쉬움

✗ C, Java 변수의 단점

단점	설명
⚠️ 코드가 길고 딱딱함	<code>int x = 5;</code> 처럼 번거로움
⚠️ 타입 변환이 불편	정수 → 문자열, 리스트 등 변환이 복잡함
⚠️ 변수 복사 시 메모리 낭비	기본형은 값이 복사돼서 메모리 더 사용됨

🎯 상황별 추천

상황	추천 언어
빠르게 프로토타입 만들기, 데이터 분석	🐍 파이썬
게임, 시스템 프로그래밍, 하드웨어 제어	⚙️ C/C++
웹, 앱 대규모 서비스 구축	💻 Java, Kotlin
과학 계산, AI 개발	🧠 파이썬

📌 한 줄 요약

- ◆ **파이썬 변수**는 자유롭고 빠르지만, 실수도 쉽게 생김
 - ◆ **C/Java 변수**는 딱딱하지만, 명확하고 안전함
-

다음 단계로 넘어가자면,

- 자료형별 차이
- 변수 참조 vs 값 복사
- 함수에서 변수 전달 방식 (call by reference vs value)

도 이어서 도와드릴 수 있어요! 무엇이 궁금하신가요? 😊