

학습목표

1. 기본 프로그래밍 구조 이해: 파일 입출력, 기본 자료형, 사용자 입력 등 프로그래밍의 기본적인 구조를 이해하고 적용하는 능력을 기릅니다.
2. 문제 해결과 디버깅 능력 향상: 코드 작성 중 발생할 수 있는 문제를 해결하고, 오류를 디버깅하는 과정에서 지속적인 학습과 성장을 경험합니다.
3. 창의적 사고 개발: 주어진 스켈레톤 코드 내에서 창의적으로 사고하고 년센스 퀴즈봇을 구현함으로써 창의적 문제 해결 능력을 개발합니다.

년센스 Quiz봇과 함께하는 코딩 여행에 오신 것을 환영합니다!



허수아비의 아들 이름은?(두 글자)

이번 프로젝트에서는 코딩을 배우는 과정을 더욱 즐겁고 흥미롭게 만들어줄 도우미, 년센스 퀴즈봇을 만들어볼 거예요. 이 퀴즈봇은 영어 단어 테스트부터 한국 지리 퀴즈까지 다양한 형태로 변신할 수 있지만, 오늘 우리는 가장 재미있는 버전인 '년센스 퀴즈'로 도전해볼 겁니다. 여러분의 창의력과 프로그래밍 기술을 발휘하여, 흥미진진한 퀴즈봇을 만들어 보세요. 시작해볼까요?

Let the coding fun begin!

파일설명

quizBot.py

- 스켈레톤 코드(기본적인 뼈대가 구현되어 있고, 살을 덧붙여 완성시킬 수 있는 코드)
- 구현 단계:
 - 1) 파일에서 퀴즈와 답 읽기
 - 2) 퀴즈 풀기 및 정답 카운트
 - 3) 점수 계산 및 출력
 - 4) 결과레포트 제작
 - 5) 퀴즈 추가 기능 ※일찍 끝난 경우
- 입력 파일: 년센스퀴즈.txt
- 출력 파일: 결과레포트.txt

quizBot정답.py

- 전체가 구현된 완성 코드
- 완성하는 방법은 여러 가지이므로, 꼭 이 파일의 방식을 따를 필요는 없습니다.

진행방법

1. quizBot.py 확인

- 단계별 스켈레톤 코드와 주석 처리되어있는 설명을 확인한다.
- 구현하고자 하는 단계의 주석 표시("'''")를 지운다.

2. 코드 작성 및 실행

- 단계별 요구사항에 맞게 코드를 작성한다.
- 실행하고 결과를 확인한다. 오류가 뜨지 않는다면, 다음 단계로 넘어간다.

3. 오류 해결

- 오류가 났을 경우, 차분히 오류 메시지를 확인하고 원인을 파악한다.
- 학습 자료 확인, 인터넷 검색, 코드 수정 등을 시도해보자.
- 도움이 필요하다면 손 들기. 선생님이 찾아가서 도와드리겠습니다~

Step 1. 파일에서 퀴즈와 답 읽기


먼저, 년센스 퀴즈와 답이 적힌 파일을 열어서 내용을 읽어봅시다.

```
# '년센스퀴즈.txt' 파일을 읽기 모드(r)로 엽니다.  
f = open("년센스퀴즈.txt", "r")  
  
# 파일의 모든 줄을 읽어와서 'lines'라는 변수에 저장합니다.  
lines = f.readlines()  
  
# 파일을 열었으면 반드시 닫아주어야 합니다.  
f.close()
```

☆ 트러블 슈팅! encoding='utf-8'

```
user@DESKTOP-B2C2U53 MINGW64 ~/Desktop/수업시연  
$ python -u "c:\Users\user\Desktop\수업시연\quizBot정답.py"  
Traceback (most recent call last):  
  File "c:\Users\user\Desktop\수업시연\quizBot정답.py", line 15, in <module>  
    lines = f.readlines()  
UnicodeDecodeError: 'cp949' codec can't decode byte 0xb0 in position 11: illegal multibyte sequence
```

오류가 생기는 건 자연스러운 일이에요!
한번에 완벽한 코드를 짜는 것보다 중요한 것은
오류가 나타났을 때 좌절하지 않고,
원인을 분석해나가는 마음가짐입니다.



- 원인

이 오류 메시지는 파일을 읽을 때 인코딩 문제가 발생했음을 나타냅니다. Python에서 파일을 열 때 기본적으로 운영체제의 기본 인코딩을 사용하는데, Windows에서는 cp949가 자주 사용됩니다. 그러나 파일이 다른 인코딩, 예를 들어 UTF-8로 인코딩되어 있다면, 이 기본 인코딩으로는 파일을 올바르게 읽을 수 없고 오류가 발생합니다.

- 해결

이 문제를 해결하려면 파일을 열 때 올바른 인코딩을 명시적으로 지정해야 합니다. 대부분의 경우, 현대 텍스트 파일은 UTF-8 인코딩을 사용하므로, 파일을 열 때 `encoding='utf-8'` 옵션을 추가하여 문제를 해결할 수 있습니다.

- 수정된 코드

```
f = open("년센스퀴즈.txt", "r", encoding='utf-8')
```

Step 2. 퀴즈 풀기

읽어온 퀴즈를 사용자에게 보여주고 답을 입력받아 정답과 비교해봅시다.

```
# 정답을 맞힌 횟수를 저장할 변수를 초기화합니다.
correct_count = 0

# 모든 퀴즈를 차례로 보여주고 사용자의 답을 받습니다.
for i in range(0, len(lines), 2): # 퀴즈와 답이 한 쌍이므로 2씩 증가
    print("퀴즈:", lines[i].strip()) # strip() 함수로 불필요한 공백과 줄바꿈을 제거
    user_answer = input("답을 입력하세요: ")
    if user_answer == lines[i+1].strip():
        print("정답입니다!")
        correct_count += 1 # 정답을 맞혔으므로 카운트를 1 증가
    else:
        print("틀렸습니다. 정답은", lines[i+1].strip(), "입니다.")
    print("-----")
```

for 반복문을 사용하여 모든 퀴즈를 차례로 보여주고, input 함수를 사용하여 사용자로부터 답을 입력받습니다. 입력받은 답과 정답을 비교하여 결과를 출력합니다.

Step 3. 점수 계산 및 출력

모든 퀴즈를 푼 후에는 점수를 계산하여 사용자에게 알려줍니다.

```
# 정답률을 계산하고 출력합니다.
score = correct_count / len(lines) * 100 # 정답률 계산
print("당신의 점수는", score, "%입니다.")
```

len(lines) // 2는 퀴즈의 총 개수를 의미합니다. 정답률은 (정답을 맞힌 횟수 / 퀴즈의 총 개수) * 100으로 계산할 수 있습니다.

Step 4. 결과 레포트 만들기

사용자의 이름과 점수를 결과레포트.txt 파일에 기록해봅시다.

```
# 사용자의 이름을 입력받습니다.  
user_name =   
  
# 결과 레포트를 파일에 기록합니다.  
with  as file:  
    file.write("이름: " + user_name + "\n")  
    file.write("점수: " + str(score) + "%")  
print("결과가 output.txt 파일에 저장되었습니다!")
```

input 함수로 사용자의 이름을 입력받고, open 함수의 "w" 모드를 사용하여 결과 레포트를 파일에 기록합니다. "w" 모드는 파일을 쓰기 모드로 열고, 파일이 이미 존재하면 내용을 지우고 새로 씁니다

*이제 모든 기능을 구현했습니다! 이 코드를 따라하며 프로그래밍의 기초를 익혀보세요
어려운 부분이 있다면 주석을 참고하거나 질문해주세요. Happy coding!*

Step 5. 퀴즈 추가 기능 (step4까지 일찍 끝난 경우 진행합니다.)

사용자가 원한다면 새로운 퀴즈와 답을 추가할 수 있도록 해봅시다.

```
# 사용자에게 퀴즈를 추가할지 물어봅니다.  
add_quiz = input("새로운 퀴즈를 추가하시겠습니까? (예/아니오): ")  
if add_quiz == "예":  
    new_quiz =   
    new_answer =   
  
    # 입력받은 퀴즈와 답을 파일에 추가합니다.  
    with  as file:  
        file.write("\n" +  + "\n" + )  
    print("퀴즈가 성공적으로 추가되었습니다!")
```

input 함수를 사용하여 사용자로부터 새로운 퀴즈와 답을 입력받고, open 함수의 "a" 모드를 사용하여 파일에 추가합니다. "a" 모드는 파일이 존재할 경우 내용을 끝에 추가하는 모드입니다.