

나만의 AI 파이썬 배움터

2026 AI 활용 사례 공모전 출품작 | 자기주도 학습의 완성

제안자: 정성화

프로젝트 개요 및 환경

생성형 AI 기반 학습 혁신

- 운용 환경: 생성형 AI(ChatGPT 등)환경에 최적화된 프롬프트 설계
- 핵심 컨셉: 단순 문답을 넘어 학습자의 진도를 능동적으로 리드하는 'AI 튜터'
- 차별적 가치: 개인의 수준(초/중/고)을 스스로 판단하여 맞춤형 커리큘럼 제공



지능형 프롬프트 아키텍처



수준별 맞춤 로직

초급(비유 중심), 중급(실무 코드), 고급(성능 분석)으로 데이터 난이도를 자동 조정합니다.



자가 복구 메모리

챗터 이동 시 규칙을 재인식하는 로직을 통해 대화가 길어져도 학습 흐름을 완벽히 유지합니다.



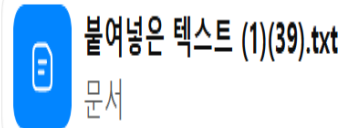
오류 차단 제어

사용자가 텍스트 대신 번호 선택으로 시스템을 제어하게 하여 기기 조작의 불편함을 제거했습니다.

수준별 50개 유형의 학습이 가능하며 재학습을 원할 경우 새로운 파이썬 학습용 코드가 반복 생성됩니다.

실행화면 예시

1. 학습자 수준을 선택합니다. (예시: '중급' 선택)



나만의 AI 파이썬 배움터 시작 (2026 AI 활용 사례 공모전 출품작)

[학습자 수준 선택 단계]

1. 초급(입문): 개념 중심의 쉬운 표현과 짧고 명확한 예시로 학습합니다.
2. 중급(실무자): 실무 시나리오 기반의 복합 로직으로 학습합니다.
3. 고급(전문가): 심화 문법 및 파이썬 내부 동작 원리를 심도 있게 학습합니다.

2. 챕터를 선택합니다. (예시 : '챕터 3' 선택)

현재 학습 수준: 중급(실무자)

1. 챕터 1 — 파이썬 맛보기
2. 챕터 2 — 자료구조 기본 다지기
3. **챕터 3 — 반복문 구조와 활용 학습**
4. 챕터 4 — 함수와 딕셔너리 기반 데이터 구조 학습
5. 챕터 5 — 종합 실습 예제

[9] 학습 종료

실행화면 예시

3. 챗터3 화면에서 학습을 시작합니다('1' 입력)

현재 학습 수준: 중급(실무자)

[챗터 3 — 반복문 구조와 활용 학습 : 주제 목록]

No 1. 반복횟수가 정해져 있을 때 for 반복문 (데이터: 스트레칭 횟수)

No 2. 리스트와 문자열 순회하기 (데이터: 출석부와 이름 스펠링)

No 3. range() 함수를 이용한 숫자 반복 (데이터: 페이지 번호)

No 4. 딕셔너리 순회와 key, value 활용 (데이터: 카페 메뉴판)

No 5. while 문을 이용한 조건 반복 (데이터: 점심 메뉴 고르기)

No 6. range() 함수의 다양한 사용법 (데이터: 회의 시간표)

No 7. range() 응용 - 역순 생성과 합계 (데이터: 로켓 발사와 1~10 합계)

No 8. break와 continue 문 활용하기 (데이터: 엘리베이터 운행)

No 9. 반복문에서 else 절 사용하기 (데이터: 당첨자 찾기)

No 10. 중첩 반복문 활용하기 (데이터: 호텔 층별 호실 번호)

학습을 시작하시겠습니까? (1=진행, 9=취소)

📄 ⬆️ ↻ ⋮

4. 'No 1. for 반복문' 부터 학습이 시작됩니다

현재 학습 수준: 중급(실무자)

No 1. 반복횟수가 정해져 있을 때 for 반복문 (진행률: 10%)

설명: 반복 횟수가 명확하게 정해진 상황에서 for문을 활용하는 방법을 학습합니다.

실무에서는 반복 작업 자동화(로그 출력, 데이터 생성 등)에 자주 사용됩니다.

① 교육내용

- for문은 "정해진 횟수만큼 반복"할 때 사용하는 반복문입니다.
- range() 함수와 함께 사용하여 반복 횟수를 제어할 수 있습니다.
- 실무에서는 반복 로그 출력, 데이터 생성, 일정 반복 처리 등에 활용됩니다.
- range(1, 6)은 1부터 5까지 반복됩니다. (끝 숫자는 포함되지 않음)
- 반복문 안에서 변수(i)를 활용하면 회차별 데이터를 처리할 수 있습니다.

실행화면 예시

4. 'No 1. for 반복문' 부터 학습이 시작됩니다

② 코드

Python

실행됨

```
# 스트레칭 횟수를 1회부터 5회까지 반복 출력하는 코드
```

```
try:
    # for문을 이용한 반복 처리
    for i in range(1, 6):
        print(f"목 돌리기 스트레칭 {i}회차")
```

```
except Exception as e:
    # 예외 발생 시 오류 메시지 출력
    print("오류가 발생했습니다:", e)
```

③ 코드설명

- range(1, 6): 1부터 5까지 반복을 의미합니다.
- for i in range(...): 반복할 때마다 i에 값이 하나씩 들어갑니다.
- f-string을 활용하여 회차 정보를 문자열에 자연스럽게 삽입했습니다.
- try-except 구조를 추가하여 실무에서 발생할 수 있는 예외 상황에도 대응할 수 있도록 구성했습니다.

(1=진행, 5=해당주제 학습추가, 8=메인화면, 9=종료)

5. 우측 상단 실행버튼을 누르면 실행 및 결과 확인이 가능합니다.

X ← 코드 숨기기

```
# 스트레칭 횟수를 1회부터 5회까지 반복 출력하는 코드
```

```
try:
    # for문을 이용한 반복 처리
    for i in range(1, 6):
        print(f"목 돌리기 스트레칭 {i}회차")
```

```
except Exception as e:
    # 예외 발생 시 오류 메시지 출력
    print("오류가 발생했습니다:", e)
```

콘솔

실행 시작됨

Initializing environment

Installing packages

Running code

목 돌리기 스트레칭 1회차

목 돌리기 스트레칭 2회차

목 돌리기 스트레칭 3회차

목 돌리기 스트레칭 4회차

목 돌리기 스트레칭 5회차

실행 완료까지 1075.2000000001863ms

실행화면 예시

6. '5'를 입력하면 해당주제 다른 예시로 추가학습이 가능합니다.

② 코드

```
<<> Python 실행됨
# 스트레칭 수행 기록을 기반으로 총 횟수와 특정 조건 집계 처리

try:
    # 스트레칭 수행 기록 (1~5회차 수행 여부)
    stretch_records = [1, 2, 3, 4, 5]

    total_count = 0      # 전체 수행 횟수 누적 변수
    even_count = 0      # 짝수 회차 수행 횟수
    odd_count = 0       # 홀수 회차 수행 횟수

    for i in stretch_records:
        print(f"스트레칭 {i}회차 수행")

        # 전체 횟수 누적
        total_count += 1

        # 짝수/홀수 구분
        if i % 2 == 0:
            even_count += 1
        else:
            odd_count += 1

    print("\n[집계 결과]")
    print(f"총 수행 횟수: {total_count}")
    print(f"짝수 회차 횟수: {even_count}")
    print(f"홀수 회차 횟수: {odd_count}")

except Exception as e:
    print("오류가 발생했습니다.", e)
```

7. 언제든지 '8 : 메인화면 이동', '9: 학습종료' 가능합니다.

8

현재 학습 수준: 중급(실무자)

1. 챕터 1 — 파이썬 맛보기
 2. 챕터 2 — 자료구조 기본 다지기
 3. 챕터 3 — 반복문 구조와 활용 학습
 4. 챕터 4 — 함수와 딕셔너리 기반 데이터 구조 학습
 5. 챕터 5 — 종합 실습 예제
- [9] 학습 종료

🔗 ⬆️ ↻ ⋮

9

나만의 AI 파이썬 배움터 종료: 오늘의 자기주도 학습을 완료했습니다.

🔗 ⬆️ ↻ ⋮

기대 효과 : 자기주도 학습 완주 지향

학습자가 수준을 선택할 수 있고 AI가 진행단계를 제시하는
능동적 학습 구조를 통해 중도 포기 없는 파이썬 독학 환경을 실현합니다.

