**REPORT**

**[ 과제 : 생성형AI를 활용한 문제 해결 보고서]**



|  |  |
| --- | --- |
| 과 목 명 | 파이썬과학프로그래밍기초 |
| 교 수 명 | 김 병 정 |
| 학 번 | 20237107 |
| 작 성 자 | 하 태 영 |
| 제 출 일 | 2025.05.30 |

**한림대학교**

|  |  |
| --- | --- |
| 작성항목 | 제출요령 |
| 활용한 AI 서비스 | Perplexity |
| AI 활용 주체 | 하태영 20237107 |
| AI 활용 문제유형 | 문제지 |
| AI 활용 과정 | 질문 : 파이썬 LC 관련 문제를 만들어줘 답변 : 파이썬 List Comprehension(LC) 관련 문제 아래는 파이썬의 리스트 컴프리헨션(List Comprehension, LC) 개념을 이해하고 활용할 수 있도록 구성한 다양한 난이도의 문제들입니다. 각 문제는 실제 코딩 테스트나 실무에서 자주 활용되는 패턴을 바탕으로 만들어졌습니다.  **1. 기본 문법 이해**   1. 0부터 9까지의 제곱수를 담은 리스트를 LC로 만드세요. 2. 1부터 20까지의 숫자 중 짝수만 담은 리스트를 LC로 만드세요.   **2. 조건문 활용**   1. 1부터 30까지의 숫자 중 3의 배수만 담은 리스트를 LC로 만드세요. 2. 다음 리스트에서 음수만 골라 새로운 리스트를 LC로 만드세요.   python  nums = [3, -7, 2, -1, 0, 9, -5]  **3. if-else(삼항 연산자) 활용**   1. 1부터 10까지의 숫자 중 홀수는 'odd', 짝수는 'even'으로 변환한 리스트를 LC로 만드세요.   **4. 중첩 for문 활용**   1. 2중 for문을 사용하여 (i, j) 형태의 튜플을 모두 담는 리스트를 LC로 만드세요. 단, i는 0~1, j는 0~2 범위입니다.   **5. 문자열 처리**   1. 문자열에서 모음(a, e, i, o, u)을 제거한 새로운 문자열을 LC로 만드세요.   python  word = 'mathematics'  **6. 2차원 리스트 평탄화**   1. 다음과 같은 2차원 리스트를 1차원 리스트로 만드는 LC를 작성하세요.   python  matrix = [  [1, 2, 3],  [4, 5, 6],  [7, 8, 9]  ]  **7. 응용 문제**   1. 1~100 사이의 소수가 아닌 수(합성수)만 담은 리스트를 LC로 만드세요. 2. a, b, c가 1~20 범위에서 a2+b2=c2a^2 + b^2 = c^2a2+b2=c2를 만족하는 피타고라스 삼각형 (a < b < c) 조합을 모두 찾는 LC를 작성하세요.   각 문제는 파이썬 LC의 다양한 활용법(조건, 삼항 연산자, 중첩, 문자열, 2차원 리스트 등)을 자연스럽게 익힐 수 있도록 설계되었습니다. |
| AI와의 협업의 장 단점 | 장점   * 문제를 유형별로 정리해서 출제해준다. * AI를 사용하지 않았다면, 문제를 찾는데 시간이 낭비되었을 것이다. 따라서 시간절감을 할 수 있다.   단점   * AI가 항상 정확한 것은 아니므로 문제에 오류가 발생할 가능성이 있다. |

|  |  |
| --- | --- |
| 작성항목 | 제출요령 |
| 활용한 AI 서비스 | Perplexity |
| AI 활용 주체 | 하태영 20237107 |
| AI 활용 문제유형 | 강의영상 |
| AI 활용 과정 | 질문 : 파이썬 리스트 컴프리헨션 강의 영상 추천해줘  답변 파이썬 리스트 컴프리헨션 강의 영상 추천 아래는 파이썬 리스트 컴프리헨션(List Comprehension)을 쉽고 명확하게 설명하는 강의 영상들입니다. 초보자부터 실전 활용까지 단계별로 참고할 수 있습니다.  **1. 1분 파이썬 - (36) 리스트 컴프리헨션**   * 짧고 핵심적인 설명으로 리스트 컴프리헨션의 기본 구조와 활용법을 빠르게 익힐 수 있습니다. * 반복문, 조건문을 한 줄로 표현하는 방법을 간단한 예제와 함께 소개합니다1[7](https://hoood.tistory.com/entry/%ED%8C%8C%EC%9D%B4%EC%8D%AC%EA%B4%80%EB%A0%A8-%EC%9D%B8%EA%B8%B0-%EC%9C%A0%ED%8A%9C%EB%B8%8C-%EC%98%81%EC%83%81-%EC%A1%B0%ED%9A%8C%EC%88%98-%EC%88%9C%EC%9C%84)6.   **2. 프로그래머 김플 스튜디오 - 파이썬 리스트 컴프리헨션 쉽게 이해하기**   * 리스트 컴프리헨션의 기초부터 반복문, 조건문 결합, 다양한 실전 예제까지 단계별로 자세히 설명합니다. * 기존 for문과 append를 사용하는 방식과 리스트 컴프리헨션을 비교해 보여주어, 차이점과 장점을 명확히 이해할 수 있습니다3.   **3. 혼자 공부하는 파이썬 30강 - 리스트 내포(List Comprehension)**   * 파이썬의 리스트 내포 구문을 예제와 함께 차근차근 설명합니다. * 반복문을 사용한 리스트 생성과 리스트 내포의 차이, 실전 문제 풀이까지 다룹니다4.   **4. 제로부터 시작하는 파이썬 몸풀기 22. 리스트 컴프리헨션**   * 파이썬 기초와 데이터 분석 실기까지 연계해 설명하는 무료 강의입니다. * 리스트 컴프리헨션의 기본 문법과 실전 예제를 함께 실습할 수 있습니다2.   **5. 리스트 컴프리헨션으로 짧고 효율적인 파이썬 코드 작성하기!**   * 복잡한 문제를 리스트 컴프리헨션으로 간결하게 해결하는 방법을 소개합니다. * 실무에서 자주 쓰는 패턴과 코드 가독성 개선 사례를 보여줍니다8.  추천 학습 순서  1. **1분 파이썬** → 기초 개념 빠르게 익히기 2. **프로그래머 김플 스튜디오** → 구조 및 실전 예제 학습 3. **혼자 공부하는 파이썬** → 반복문과의 차이, 실전 문제 풀이 4. **제로부터 시작하는 파이썬** → 데이터 분석 관점에서 활용 연습 5. **짧고 효율적인 코드 작성하기** → 실무 응용 및 코드 최적화   각 강의는 유튜브에서 제목으로 검색하면 쉽게 시청할 수 있습니다. 초보자도 따라하기 쉽고, 실전 예제도 풍부해 리스트 컴프리헨션을 완벽하게 익힐 수 있습니다. |
| AI와의 협업의 장 단점 | 장점   * AI를 활용하면 자료 탐색 속도가 빠르다. * 만약 AI를 활용하지 않는다면 자료 탐색 속도가 오래 걸렸을 것이다.   단점   * 부적합한 자료가 포함될 가능성이 있으므로 직접 검증이 필요하다. |

|  |  |
| --- | --- |
| 작성항목 | 제출요령 |
| 활용한 AI 서비스 | Perplexity |
| AI 활용 주체 | 하태영 20237107 |
| AI 활용 문제유형 | 오프라인수업 |
| AI 활용 과정 | 질문 : 파이썬 LC에서 if문이 1개 일 때와 여러 개일 때의 차이는 무엇인지 이해가 잘 안된다.  답변 : 파이썬 리스트 컴프리헨션에서 if문이 1개일 때와 여러 개일 때의 차이 파이썬 리스트 컴프리헨션(List Comprehension, LC)에서 if문을 한 개만 쓰는 것과 여러 개 쓰는 것의 차이를 아래와 같이 정리할 수 있습니다.  **1. if문이 1개일 때**   * 기본 구조: 표현식for변수in반복가능객체if조건식표현식 for 변수 in 반복가능객체 if 조건식표현식for변수in반복가능객체if조건식 * 동작: 각 요소에 대해 조건식이 True인 경우만 리스트에 포함시킵니다.   python  a = [i **for** i **in** range(10) **if** i % 2 == 0]  *# 결과: [0, 2, 4, 6, 8]*  이 예시에서는 0~9 중 짝수만 리스트에 포함됩니다[1](https://seolwonkoo.tistory.com/28)[5](https://hobbylife.tistory.com/entry/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A6%AC%EC%8A%A4%ED%8A%B8-%EC%BB%B4%ED%94%84%EB%A6%AC%ED%97%A8%EC%85%98List-Comprehension%EC%82%AC%EC%9A%A9%EB%B0%A9%EB%B2%95-4%EA%B0%80%EC%A7%80-%EC%BD%94%EB%93%9C%EC%B6%95%EC%95%BD%EC%A4%84%EC%9D%B4%EA%B8%B0)[7](https://hgk5722.tistory.com/413).  **2. if문이 여러 개일 때**   * 기본 구조: 표현식for변수in반복가능객체if조건1if조건2...표현식 for 변수 in 반복가능객체 if 조건1 if 조건2 ...표현식for변수in반복가능객체if조건1if조건2... * 동작: 여러 if문은 **and** 조건처럼 동작합니다. 즉, 모든 조건이 True일 때만 리스트에 포함됩니다.   python  a = [i **for** i **in** range(10) **if** i % 2 == 0 **if** i > 5]  *# 결과: [6, 8]*  이 코드는 0~9 중 "짝수이면서 5보다 큰" 값만 리스트에 포함시킵니다. 즉, if i % 2 == 0 그리고 if i > 5를 모두 만족해야 합니다[3](https://github.com/shoark7/Effective-Python/blob/master/files/BetterWay08_ListComprehension.md)[4](https://pythonexamples.org/python-list-comprehension-multiple-if-conditions/)[5](https://hobbylife.tistory.com/entry/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A6%AC%EC%8A%A4%ED%8A%B8-%EC%BB%B4%ED%94%84%EB%A6%AC%ED%97%A8%EC%85%98List-Comprehension%EC%82%AC%EC%9A%A9%EB%B0%A9%EB%B2%95-4%EA%B0%80%EC%A7%80-%EC%BD%94%EB%93%9C%EC%B6%95%EC%95%BD%EC%A4%84%EC%9D%B4%EA%B8%B0)[7](https://hgk5722.tistory.com/413).   * 아래와 같이 **and** 연산자로 한 줄에 쓸 수도 있습니다.   python  a = [i **for** i **in** range(10) **if** i % 2 == 0 **and** i > 5]  *# 결과: [6, 8]*  두 방식은 결과가 동일합니다. 여러 if문을 나열하면 내부적으로 and로 연결한 것과 같습니다[3](https://github.com/shoark7/Effective-Python/blob/master/files/BetterWay08_ListComprehension.md)[7](https://hgk5722.tistory.com/413).  **3. if-else와의 차이**   * if문만 쓰면 "포함 여부"를 결정합니다. * if-else문은 "포함은 무조건 하되, 값 자체를 조건에 따라 다르게" 만듭니다.   python  a = [i **if** i % 2 == 0 **else** '홀수' **for** i **in** range(1, 6)]  *# 결과: ['홀수', 2, '홀수', 4, '홀수']*  여기서는 모든 값이 리스트에 들어가지만, 짝수면 숫자, 홀수면 '홀수'로 변환됩니다[1](https://seolwonkoo.tistory.com/28)[2](https://sparkbyexamples.com/python/if-else-in-a-list-comprehension/)[7](https://hgk5722.tistory.com/413).  **정리**   * if문 여러 개 → 각각의 조건이 모두 True일 때만 포함(암시적 and)[3](https://github.com/shoark7/Effective-Python/blob/master/files/BetterWay08_ListComprehension.md)[4](https://pythonexamples.org/python-list-comprehension-multiple-if-conditions/)[5](https://hobbylife.tistory.com/entry/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A6%AC%EC%8A%A4%ED%8A%B8-%EC%BB%B4%ED%94%84%EB%A6%AC%ED%97%A8%EC%85%98List-Comprehension%EC%82%AC%EC%9A%A9%EB%B0%A9%EB%B2%95-4%EA%B0%80%EC%A7%80-%EC%BD%94%EB%93%9C%EC%B6%95%EC%95%BD%EC%A4%84%EC%9D%B4%EA%B8%B0). * if문 1개 → 해당 조건만 True면 포함. * 여러 조건을 and/or로 한 줄에 써도 되고, if문을 여러 번 나열해도 됨. * if-else는 포함 여부가 아니라 "값 자체를 조건에 따라 다르게" 만듦[2](https://sparkbyexamples.com/python/if-else-in-a-list-comprehension/)[7](https://hgk5722.tistory.com/413).  예시 비교  | **형태** | **예시** | **결과** | **의미** | | --- | --- | --- | --- | | if 1개 | [x for x in a if x > 5] | 6, 7, 8, 9, 10 | 5보다 큰 값만 포함 | | if 2개 | [x for x in a if x > 5 if x % 2 == 0] | 6, 8, 10 | 5보다 크고 짝수만 포함 | | if-else | [x if x % 2 == 0 else '홀수' for x in a] | '홀수', 2, '홀수', 4, ... | 짝수는 숫자, 홀수는 '홀수'로 변환 |   **결론:**   * if문을 여러 개 쓰면 모두 만족해야 포함되고, and로 연결한 것과 같다. * if-else는 포함 여부가 아니라, 값을 조건에 따라 다르게 만든다[2](https://sparkbyexamples.com/python/if-else-in-a-list-comprehension/)[3](https://github.com/shoark7/Effective-Python/blob/master/files/BetterWay08_ListComprehension.md)[4](https://pythonexamples.org/python-list-comprehension-multiple-if-conditions/)[5](https://hobbylife.tistory.com/entry/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A6%AC%EC%8A%A4%ED%8A%B8-%EC%BB%B4%ED%94%84%EB%A6%AC%ED%97%A8%EC%85%98List-Comprehension%EC%82%AC%EC%9A%A9%EB%B0%A9%EB%B2%95-4%EA%B0%80%EC%A7%80-%EC%BD%94%EB%93%9C%EC%B6%95%EC%95%BD%EC%A4%84%EC%9D%B4%EA%B8%B0)[7](https://hgk5722.tistory.com/413). |
| AI와의 협업의 장 단점 | 장점 :   * AI를 활용하면 이해가 안되거나 모르는 정보를 빠르게 탐색 할 수 있다. * 만약 AI를 활용하지 않았다면, 탐색 시간이 많이 소요되었을 것이다.   단점 :   * AI의 설명은 논리적이고 빠르지만, 학습자의 정서적 반응, 동기 부여, 개별 상황에 따른 공감적 피드백은 부족할 수 있다. * 어느 부분에서 왜 어려워하는지에 대한 섬세한 파악과 격려는 교육자의 역할이 여전히 중요하다. |