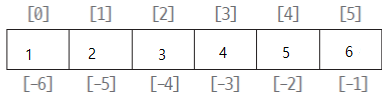
1문제)

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)



value를 그림으로 나타내보면 이렇게 된다.

1-(1) 그러므로 values[1]은 2가 된다.

1-(2) 맨 끝에 값부터 [-1]이 시작되기 때문에 6이 된다.

1-(2’) value[1]부터 value[3-1]까지 출력되므로 [2, 3]이 출력된다.

1-(3) len은 리스트의 길이가 출력되므로 6이 출력된다.

1-(4) 5의 값이 들어있는 곳이 values[4]이므로 4가 출력된다.

1-(5) 가장 큰 수가 출력되므로 6이 출력된다

1-(6) 모든 수의 합이 출력되므로 21이 출력된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

텍스트이(가) 표시된 사진

자동 생성된 설명

위와 같은 방식을 이용해 한 줄로 리스트를 정의한 수 있다.

2-(1) values = [0] \* 10 혹은 values = [0 for x in range(10)] 로 쓸 수 있지만 같은 수가 값으로 계속 들어가기 때문에 전자의 방법을 쓰는 게 더 간편해 보인다.

2-(2) values = [x for x in range(1, 11)] 1부터 10까지 이므로 range(1,11)을 쓴다.

2-(3) values = [x\*2 for x in range(1, 6)] 구구단 2x5까지 출력하므로 range는 (1, 6)까지 쓴 뒤 2를 곱해준다.

2-(4) values = [x\*\*2 for x in range(1, 11)] 1부터 10까지 각 수의 제곱 수 이므로 range(1,11)을 제곱해준다.

텍스트, 편지이(가) 표시된 사진

자동 생성된 설명

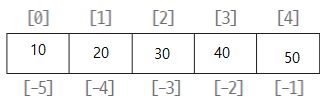
풀이 및 답)

[0]의 위치인 10부터 [(3-1)]의 위치인 30까지 삭제되므로 [40, 50, 60]이 출력된다.

텍스트, 편지이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

이므로

40과 [30, 40]이 출력된다.

음수여도 똑같이 [-3]의 위치인 30부터 [(-1-1)]의 위치인 40까지이므로 헷갈리지 않다.

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

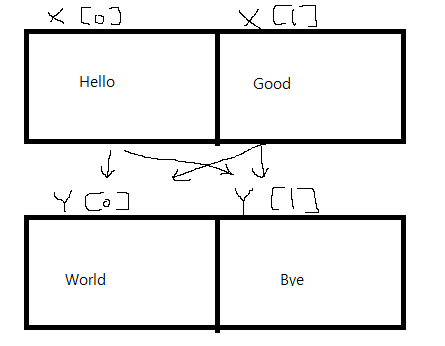
리스트[ start : stop : step] 이므로 -2이만큼 step하므로 역으로 한 칸을 뛰어넘게 된다.

그래서 [50, 30, 10]이 출력되게 된다.

차트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)



문자열에서 +는 두 문자열이 합쳐지게 된다.

그러므로 ['HelloWorld', 'HelloBye', 'GoodWorld', 'GoodBye']가 출력된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

문자열의 대소를 비교할 때는 맨 앞자리부터 a-z순으로 누가 더 z에 가까운지를 비교하게 된다. 더 심화해서 말하면 맨 앞자리부터 ASCII코드로 바꿔서 뭐가 더 큰지 비교하게 된다. 그래서 zz가 출력된다

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

aList[1]인 2부터 aList[4-1]인 4까지 값이 치환되어 [1, 5, 6, 7]가 들어가게 된다.

여기서 궁금한 한 점이 생긴 나는 aList[1:4] = [5, 6, 7, 8]를 써보았다.

그랬더니 [1, 5, 6, 7, 8]이 나왔다. 이로써 알게 된 점은 [3]까지 값이 치환되는 것이 아니라 그냥 2부터 값이 맞춰 들어가는 것 같다. 파이썬 쉽지 않다… 그래도 이렇게 이것저것 해보려고 하는 것 같다. 메커니즘을 이해하는데 도움이 된다.

텍스트, 편지이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

aList 리스트의 값들에 2를 곱한 값들이 들어가므로

[2, 4, 6, 8, 10, 12, 14]가 출력된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

맨 끝에 60이 추가되므로 [10, 20, 30, 40, 50, 60]이 출력된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

(1), (4)이 복사가 된다. list를 이용하여 깊은 복사하는 것은 수업시간에 배웠고 copy() 메서드를 이용해서도 복사를 할 수 있다. list 자체 코드에도 copy()메서드가 사용되었기 때문에 결국 같은 개념이다. 링크를 복사하는 얕은 복사가 아닌 값 자체를 복사하는 깊은 복사이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

12-(1) 맨 끝에 10이 추가되므로 [1, 2, 3, 4, 5, 6, 10]이 된다.

12-(2) 맨 뒤의 값이 빠지게 되므로 [1, 2, 3, 4, 5]가 된다.

12-(3) values[1]의 자리에 20이 들어가고 값들이 밀리기 때문에 [1, 20, 2, 3, 4, 5, 6]이 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이 및 답)

0인 처음부터 길이인 끝까지 2이므로 한 칸씩 건너뛰어서 그 값을 더한 result가 출력되기 때문에 1 + 3 + 5 + 7 + 9 인 25가 출력된다.

텍스트, 편지이(가) 표시된 사진

자동 생성된 설명

values[i] = i \* i

풀이 및 답)

문제가 짤려서 책을 참고했다.

맨 끝 값이 values[9]인데 이 for문은 (11-1)인 values[10]에 값을 넣으려고 하니까 잘못되었다.

다음 페이지로….

2문제)

텍스트이(가) 표시된 사진

자동 생성된 설명

소스코드)

텍스트이(가) 표시된 사진

자동 생성된 설명

실행화면)

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이)

입력 받을 값의 개수를 count 변수에 입력 받는다. 리스트 num을 선언해준다. 카운트 수만큼 반복문 for문을 돌려준다. 값을 입력 받고 append 함수를 이용해 리스트 num에 추가해준다. 마지막으로 내장함수 sum()을 이용해 합을 구하고 print로 출력해준다.

3문제)

텍스트이(가) 표시된 사진

자동 생성된 설명

소스코드)

텍스트이(가) 표시된 사진

자동 생성된 설명

실행화면)

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이)

처음 문제를 봤을 때 이해가 되지 않았다. 문제가 디테일한 조건들이 많이 빠진 것 같다. 책에 출제자가 낸 의도는 지정된 값들이 저장된 num 리스트를 for문을 이용해 별을 찍는 것 같은데 이게 맞는지 엄청 고민한 것 같다. 책이 오타도 많고 퀄리티가 조금 아쉬운 것 같다… 먼저 num 리스트의 실행결과에 나온 값들을 넣어주고 for문을 사용하여 반복한다. 여기서 결과보면 스페이스 약 4칸 같이 보여서 tab이구나 싶어 \t를 이용하여 띄어쓰기를 해주었고 문자열 곱하기를 사용하여 출력해주었다.