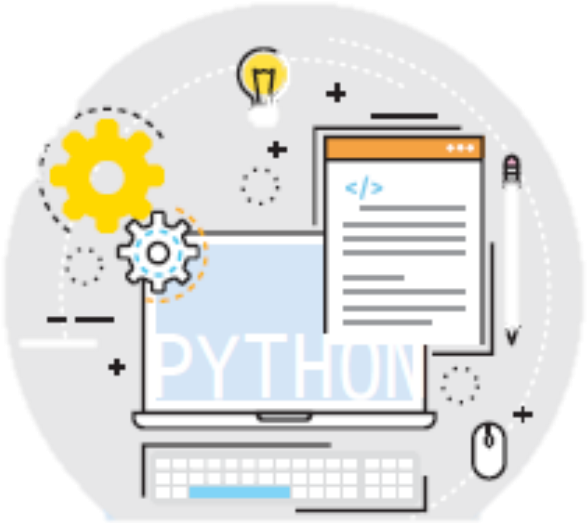


CHAPTER 13

패턴 퀴즈 게임기

학습 목표

기본 학습 목표

- 다양한 패턴을 만들 수 있다.
- 패턴을 이용하여 패턴을 묻고 맞히는 함수를 만들 수 있다.

심화 학습 목표

- 일차원 리스트와 이차원 리스트의 차이를 이해할 수 있다.
- 이차원 리스트를 이용하여 패턴을 묻고 맞히는 함수를 호출할 수 있다.

핵심 학습 요소

핵심 학습 요소

- 리스트, 사용자 정의 함수, 리턴 값 2개 이상

문제 상황

문제 상황 CT

다양한 숫자 패턴을 맞히는 놀이를 하고 싶다. 다양한 숫자 패턴을 생성한 다음 각 패턴에 대해 제일 앞 숫자부터 차례대로 보여준다. 각 패턴에서 제일 마지막 숫자를 플레이어가 추론한다.

문제 분석

입력

- 추론한 패턴 숫자를 입력

출력

- 패턴 추론 맞힘 여부
- 몇 개를 맞혔는지 알려주기

문제 분석

문제 분해

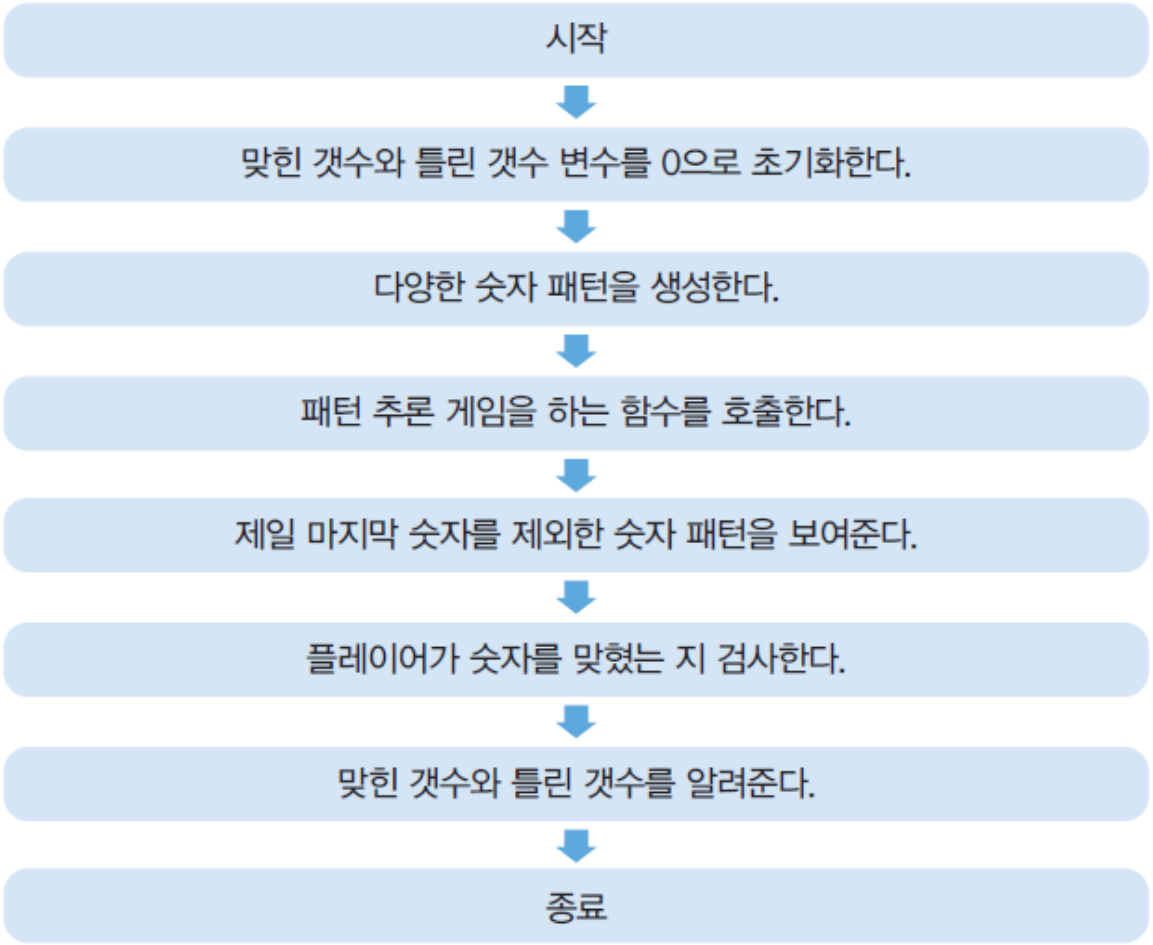
- 임의의 숫자 패턴 생성하기
- 패턴을 인자로 받아서 패턴 추론하는 함수 작성하기
- 함수 호출하여 패턴을 인자로 전달하여 처리하기

데이터

- 다양한 패턴의 숫자형 리스트
- 맞고 틀린 개수

알고리즘

전체 알고리즘 뼈대



프로그래밍

변수

- correctAns: 패턴 추론을 맞힌 개수
- wrongAns: 패턴 추론이 들린 개수
- pattern1, pattern2, pattern3, pattern4, pattern5: 숫자형 패턴 저장 리스트

프로그래밍

- 변수 초기화
- 숫자형 패턴 생성

```
correctAns =0
wrongAns = 0

pattern1 = [2, 4, 6, 8]
pattern2 = [13, 16, 19, 22]
pattern3 = [2, 3, 5, 7, 11]
pattern4 = [1, 1, 2, 3, 5, 8]
pattern5 = [31, 28, 31, 30]
```

프로그래밍

- 패턴 추론 함수 호출
- 맞힌 개수와 틀린 개수 리턴

```
correctAns, wrongAns = patternmatch(pattern1, correctAns, wrongAns)
correctAns, wrongAns = patternmatch(pattern2, correctAns, wrongAns)
correctAns, wrongAns = patternmatch(pattern3, correctAns, wrongAns)
correctAns, wrongAns = patternmatch(pattern4, correctAns, wrongAns)
correctAns, wrongAns = patternmatch(pattern5, correctAns, wrongAns)


print("%d개 패턴중  %d개 맞았어요" %(correctAns + wrongAns, correctAns))
```

실습해보기

실습해보기 13-1

위의 프로그램에 새로운 패턴을 하나 추가해 보자.

프로그래밍

- 패턴 비교 함수 정의
  - 페턴을 인자로 받아 플레이어에게 패턴을 보여주고 맞히도록 함
  - 리스트 크기 : len()
  - 줄바꿈 방지: end=" "

```
def patternmatch(pattern, correctAns, wrongAns):

    for i in range(len(pattern)-1):
        print(pattern[i], end=" ")

    guessAns = int(input("다음 수는 무엇일까요?"))

    if guessAns == pattern[len(pattern)-1]:
        correctAns = correctAns +1
        print("잘 했어요. 축하해요")
    else:
       wrongAns = wrongAns +1
       print("정답은%d 입니다" %(pattern[len(pattern)-1]))
    return correctAns, wrongAns
```

실습해보기

실습해보기 13-2

global 키워드를 이용하여 함수 밖의 correctAns, wrongAns 변수 값을 갱신해 보자.

프로그래밍 – 전체 프로그램

- 1번 째 줄
  - 패턴을 인자로 받아서 이를 패턴을 맞히도록 하는 patternmatch() 함수
- 1-17번째 줄
  - 맞은 개수와 틀린 개수 초기화
- 25번째 줄
  - 5개의 서로 다른 패턴을 인자로 전달하여 맞은 개수와 틀린 개수를 리턴 값으로 받음

```
01  def patternmatch(pattern, correctAns, wrongAns):  # 패턴 맞히기 함수
02
03      for i in range(len(pattern)-1):
04          print(pattern[i], end=" ")
05
06      guessAns = int(input("다음 수는 무엇일까요?"))
07
08      if guessAns == pattern[len(pattern)-1]:
09          correctAns = correctAns +1
```

프로그래밍 – 전체 프로그램

```
10          print("잘 했어요. 축하해요")
11      else:
12         wrongAns = wrongAns +1
13         print("정답은%d 입니다" %(pattern[len(pattern)-1]))
14      return correctAns, wrongAns
15
16  correctAns =0
17  wrongAns = 0
18
19  pattern1 = [2, 4, 6, 8]          # 다양한 숫자 패턴들
20  pattern2 = [13, 16, 19, 22]
21  pattern3 = [2, 3, 5, 7, 11]
22  pattern4 = [1, 1, 2, 3, 5, 8]
23  pattern5 = [31, 28, 31, 30]
24
25  correctAns, wrongAns = patternmatch(pattern1, correctAns, wrongAns)
26  correctAns, wrongAns = patternmatch(pattern2, correctAns, wrongAns)
27  correctAns, wrongAns = patternmatch(pattern3, correctAns, wrongAns)
28  correctAns, wrongAns = patternmatch(pattern4, correctAns, wrongAns)
29  correctAns, wrongAns = patternmatch(pattern5, correctAns, wrongAns)
30
31
32  print("%d개 패턴중  %d개 맞았어요" %(correctAns + wrongAns, correctAns))
```

테스트와 디버깅

테스트와 디버깅 CT

입력값/ 출력 결과	확인 및 유의사항
2 4 6 다음 수는 무엇일까요?8 잘 했어요. 축하해요 13 16 19 다음 수는 무엇일까요?22 잘 했어요. 축하해요 2 3 5 7 다음 수는 무엇일까요?11 잘 했어요. 축하해요 1 1 2 3 5 다음 수는 무엇일까요?8 잘 했어요. 축하해요 31 28 31 다음 수는 무엇일까요?30 잘 했어요. 축하해요 5개 패턴중 5개 맞았어요	▪ 예상된 테스트 결과인지 확인한다. ▪ 함수 호출시 인자 전달이 제대로 되었는지 확인한다.

심화활동

심화 활동 CT

패턴을 1차원 리스트로 생성하니 여러 리스트 변수를 생성해야 하고 함수 호출시 각 리스트별로 호출하여 반복성이 있음을 알 수 있다. 따라서 2차원 리스트를 사용하여 프로그램을 더 단순화하는 방안을 찾아보자. 또한, 패턴 제일 마지막 숫자 보다는 임의의 위치에 있는 숫자를 묻는 형태로 수정해 보자.