2024학년도 상암고등학교 2학년 프로그래밍 교과
파이썬 코드작성 수행평가 문제집 및 정답 코드 안내

1. 수행평가 실시일 : 24년 5월 16일(목) 수업시간 중
2. 수행평가 출제 방식
   - 해당 문제집에서 5문제 출제(기초 2, 응용 2, 심화1)
   - 문제당 20점(부분점수 있음), 총 100점
   - 코랩으로 코드 작성 후 스크린샷(동작결과까지 포함) pdf 파일로 제작 후 구글폼 제출
3. 담당교사 : 성원경
4. 코랩 정답지 URL :https://pythonkorea.com/128
5. 문의 : wonking710@naver.com

[기초] 1번 문제 : input()함수와 fstring 활용 문제

[문제 조건]

- 입력 : 이름, 나이, 성격, 취미, 기타소개하고싶은 내용 값을 input()함수로 받은 뒤 변수 저장
- 출력 : print의 f 스트링 문법을 써서 위 변수값 출력 하여 자기소개 하기

[코드 결과 예시]

```
성격 >> ENFP
취미 >> 자전거타기
기타 소개하고 싶은 내용은? >> 귀여운 1살 딸

<<< 출력 예시, 스타일은 달라도 됨 >>>
안녕! 내이름은 성원경이야. 나이는 35살이지.
성격은 ENFP이고, 취미는 자전거타기야.
아참, 그리고 귀여운 1살 딸도 있어
```

```python
### 1번 정답 코드

# 입력 받기
이름 = input("이름 모닝 >> ")
별명 = input("별명 모닝 >> ")

# f스트링으로 출력
print(f"안녕 ! 내 이름은 {이름}이야. 사람들은 나에게 {별명}이라 부르지 ")
```

```
이름 모닝 >> 성원경
별명 모닝 >> 상암동 실바형
안녕 ! 내 이름은 성원경이야. 사람들은 나에게 상암동 실바형이라 부르지
```

[기초] 2번 문제 : 리스트 함수 활용

[문제 조건]

- 1) 음식 리스트 생성 ['김밥','돈까스','제육덮밥','탕후루']
- 2) '탕후루' 원소 제거
- 3) '김치찌개' 원소 추가
- 4) 가나다라 순으로 정렬
- 위 단계별로 변화되는 리스트 출력할 것.
- 음식은 바뀌어도 상관없음.
- 코드 결과 예시

```
['김밥', '돈까스', '제육덮밥', '탕후루']
['김밥', '돈까스', '제육덮밥']
['김밥', '돈까스', '제육덮밥', '김치찌개']
['김밥', '김치찌개', '돈까스', '제육덮밥']
```

```python
### 2번 정답 코드
빅뱅 = ['승리', '지디', '탑', '태양']
print(빅뱅)

빅뱅.remove('승리')
print(빅뱅)

빅뱅.append('대성')
print(빅뱅)

빅뱅.sort()
print(빅뱅)
```

[기초] 3번 문제 : 평균 성적 계산하기 문제

[문제 조건]

- 입력 : 국어, 영어, 수학, 사회, 과학, 정보 등 성적 입력
- 출력 : 평균값 계산후 평균값에 맞는 등급 출력
- ex) 평균 > 90 : A, 평균 > 80 : B, ... 평균 < 70 : D 이런식.

[코드 결과 예시]

```
국어 성적을 입력하세요 : 100
영어 성적을 입력하세요 : 40
수학 성적을 입력하세요 : 100
사회 성적을 입력하세요 : 50
과학 성적을 입력하세요 : 60
정보 성적을 입력하세요 : 100
평균 점수는 75.00점이며, 부여된 학점은 'C'입니다.
```

```python
### 3번

#점수 입력받기

국어 = int(input("국어 몇점? >> "))
수학 = int(input("수학 몇점? >> "))
정보 = int(input("정보 몇점? >> "))

평균 = (국어 + 수학 + 정보) / 3

if 평균 > 90 :
    print("A")
if 평균 > 80 :
    print("B")
if 평균 < 80 :
    print("C")
```

[기초] 4번 문제 : 구구단 출력 코드

[문제 조건]

- 입력 : 1~100까지 정수 입력, 범위에 벗어나면 100(클때) 혹은 1(작을때)로 고정
- 출력 : 해당 구구단 코드 출력

[코드 결과 예시]

```
몇 단? >> :500
100단을 출력합니다.
100*1=100
100*2=200
100*3=300
100*4=400
100*5=500
100*6=600
100*7=700
100*8=800
100*9=900
```

```python
### (기초) 4번 문제 정답 코드

# 입력
단수 = int(input("몇단 >> "))
# 범위 제한(클램프)
if 단수 > 100 :
    단수 = 100
if 단수 < 1 :
    단수 = 1

# 구구단 출력
print(단수, "단을 출력합니다. " )
for ii in range(1, 10) :
    print(f"{단수} * {ii} = {단수*ii}")


```

```
몇 단? >> :500
100단을 출력합니다.
100*1=100
100*2=200
100*3=300
100*4=400
100*5=500
100*6=600
100*7=700
100*8=800
100*9=900
```

[기초] 5번 문제 : 데이터 슬라이싱을 활용한 현재 시간 출력하기

[문제 조건]

- 2024-05-08 08:45:53.239165+09:00 형태의 날짜_문자열 데이터를 받아 온 뒤 이를 데이터 슬라이싱 하여
- 현재 시간은 2024년 05월 08일 08시 45분입니다. 형태로 출력할 것

[초기 코드 제시]

```
from datetime import datetime
import pytz

# 한국 시간대 설정
한국_시간대 = pytz.timezone('Asia/Seoul')

# 현재 시간을 한국 시간대로 가져오기
현재_시간_한국 = datetime.now(한국_시간대)

# 날짜 및 시간을 문자열로 변환
날짜_문자열 = str(현재_시간_한국)
print(날짜_문자열)
```

[결과 예시]

```
2024-05-08 08:45:53.239165+09:00
현재 시간은 2024년 05월 08일 08시 45분입니다.
```

```
## 5번문제

from datetime import datetime
import pytz

# 한국 시간대 설정
한국_시간대 = pytz.timezone('Asia/Seoul')

# 현재 시간을 한국 시간대로 가져오기
현재_시간_한국 = datetime.now(한국_시간대)

# 날짜 및 시간을 문자열로 변환
날짜_문자열 = str(현재_시간_한국)
print(날짜_문자열)

## 여기부터 코드작성 시작
년 = 날짜_문자열[0:4]
월 = 날짜_문자열[5:7]
일 = 날짜_문자열[8:10]
시간 = 날짜_문자열[11:13]
분 = 날짜_문자열[14:16]

print(f"현재시간 : {년}년 {월}월 {일}일 {시간}시 {분}분")
```

```
2024-05-08 08:45:53.239165+09:00
현재 시간은 2024년 05월 08일 08시 45분입니다.
```

[응용] 6번 문제 : 2차 함수 생성 후 x값 넣기

[문제 조건]

- y= ax^2+ bx +c 형식의 이차함수 생성
- a = 1, b = 3, c = 2 지정
- 2차함수 형태 출력 후
- x값이 1, 10까지 순차적으로 10번 들어갈때
- y값 출력하여라

[초기 코드 제시]

```
# a, b, c 값을 설정
a = 1
b = 3
c = 2

# 함수 형태 출력
print("함수 형태")
print(f"f(x) = {a}x²  + {b}x + {c}")
```

[결과 예시]

```
함수 형태
f(x) = 1x²  + 3x + 2
결과 출력
x = 1, y = 6
x = 2, y = 12
x = 3, y = 20
x = 4, y = 30
x = 5, y = 42
x = 6, y = 56
x = 7, y = 72
x = 8, y = 90
x = 9, y = 110
x = 10, y = 132
```

```python
### (응용) 6번 문제 정답
def 이차함수(a, b, c, x):
    return a * x**2 + b * x + c

# a, b, c 값을 설정
a = 1
b = 3
c = 2

# 함수 형태 출력
print("함수 형태")
print(f"f(x) = {a}x²  + {b}x + {c}")

print("결과 출력")
# x 값이 1부터 10까지의 값을 가지도록 설정
for x in range(1, 11):
    y = 이차함수(a, b, c, x)
    print(f"x = {x}, y = {y}")
```

```
함수 형태
f(x) = 1x²  + 3x + 2
결과 출력
x = 1, y = 6
```

[응용] 7번 문제 : BMI 계산기

[문제 조건]

- 아래 공식에 맞는 BMI 계산기를 만들기 위해 키(cm), 몸무게(kg)을 입력 받고, 계산 한 뒤 범위에 맞게 등급을 출력하여라.

$$BMI = \frac{\text{몸무게(KG)}}{\text{키(M)} \times \text{키(M)}}$$

[코드 힌트]

```
def bmi_category(bmi):
    # BMI 값에 따른 카테고리 반환
    if bmi < 18.5:
        return "저체중"
    elif 18.5 <= bmi < 25:
        return "정상 체중"
    elif 25 <= bmi < 30:
        return "과체중"
    else:
        return "비만"
```

[결과 예시]

```
체중(kg)을 입력하세요: 75
키(cm)을 입력하세요: 176
당신의 BMI는 24.21이며, 이는 '정상 체중' 범주에 속합니다.
```

```python
#7번 문제 코드 정답
def calculate_bmi(weight, height):
    bmi = weight / (height / 100) ** 2  # 키를 미터 단위로 변환 후 BMI 계산
    return bmi

def bmi_category(bmi):
    # BMI 값에 따른 카테고리 반환
    if bmi < 18.5:
        return "저체중"
    elif 18.5 <= bmi < 25:
        return "정상 체중"
    elif 25 <= bmi < 30:
        return "과체중"
    else:
        return "비만"

# 사용자로부터 체중과 키 입력 받기
weight = float(input("체중(kg)을 입력하세요: "))
height = float(input("키(cm)을 입력하세요: "))

# BMI 계산
bmi = calculate_bmi(weight, height)

# BMI 카테고리 결정
category = bmi_category(bmi)

# 결과 출력
print(f"당신의 BMI는 {bmi}이며, 이는 '{category}' 범주에 속합니다.")
```

```
체중(kg)을 입력하세요: 75
키(cm)을 입력하세요: 176
당신의 BMI는 24.210743801652892이며, 이는 '정상 체중' 범주에 속합니다.
```

[응용] 8번 문제 : 사칙연산 계산기 프로그램 만들기

[문제 조건]

- 입력 : 숫자1, 숫자2, 연산기호
- 지원되는 연산 기호는 더하기 (+), 빼기 (-), 곱하기 (*), 나누기 (/) 입니다.
- try-execept 구문으로 발생할 수 있는 에러 처리
- 출력 : 연산기호에 맞는 계산값 출력

[코드 힌트]

```
try:
    # 사용자 입력 받기
    num1 = float(input("첫 번째 숫자를 입력하세요: "))
    num2 = float(input("두 번째 숫자를 입력하세요: "))
    operator = input("연산자를 입력하세요 (+, -, *, /): ")

    # 계산 결과 얻기
    result = 계산기(num1, num2, operator)

    # 결과 출력
    print(f"결과는: {result}")

except ValueError:
    # 숫자 변환 중 오류 발생 시
    print("오류: 유효한 숫자를 입력하세요.")
```

[결과 예시]

```
첫 번째 숫자를 입력하세요: 50
두 번째 숫자를 입력하세요: 100
연산자를 입력하세요 (+, -, *, /): -
결과는: -50.0
```

```
[ ]
### 8번 코드 정답

def 계산기(num1, num2, operator):
    # 연산자에 따른 계산 수행
    if operator == '+':
        return num1 + num2
    elif operator == '-':
        return num1 - num2
    elif operator == '*':
        return num1 * num2
    elif operator == '/':
        if num2 == 0:
            return "오류: 0으로 나눌 수 없습니다."
        return num1 / num2
    else:
        return "오류: 잘못된 연산자입니다."

try:
    # 사용자 입력 받기
    num1 = float(input("첫 번째 숫자를 입력하세요: "))
    num2 = float(input("두 번째 숫자를 입력하세요: "))
    operator = input("연산자를 입력하세요 (+, -, *, /): ")

    # 계산 결과 얻기
    result = 계산기(num1, num2, operator)

    # 결과 출력
    print(f"결과는: {result}")

except ValueError:
    # 숫자 변환 중 오류 발생 시
    print("오류: 유효한 숫자를 입력하세요.")

```

```
첫 번째 숫자를 입력하세요: 50
두 번째 숫자를 입력하세요: 100
연산자를 입력하세요 (+, -, *, /): -
결과는: -50.0
```

[응용] 9번 문제 : 업 다운 숫자 맞추기 게임 만들기

[문제 조건]

- 1~31까지 정수 입력, 범위에 벗어나면 범위 벗어남 경고
- 문자 입력시 다시 입력하라고 경고 발생
- 경고 3번 누적시 프로그램 종료
- 맞추면 정답 알려주고 프로그램 종료

[코드 힌트]

```
import random

target_number = random.randint(1, 31)  # 1에서 31 사이에서 숫자 하나를 랜덤하게 선택
warning_count = 0  # 경고 횟수를 세는 변수
```

[결과 예시]

```
1에서 31 사이의 숫자를 입력하세요: 15
업! 더 높은 숫자입니다.
1에서 31 사이의 숫자를 입력하세요: 23
다운! 더 낮은 숫자입니다.
1에서 31 사이의 숫자를 입력하세요: 19
정답입니다! 숫자는 19 이었습니다. 게임을 종료합니다.
```

```python
### (응용) 9번 문제 정답 코드
import random

target_number = random.randint(1, 31)  # 1에서 31 사이에서 숫자 하나를 랜덤하게 선택
warning_count = 0  # 경고 횟수를 세는 변수

while True:
    if warning_count >= 3:
        print("경고 횟수 초과! 프로그램을 종료합니다.")
        break

    try:
        guess = input("1에서 31 사이의 숫자를 입력하세요: ")
        guess = int(guess)  # 입력 값을 정수로 변환

        if guess < 1 or guess > 31:
            warning_count += 1
            print("범위를 벗어났습니다! 1에서 31 사이의 숫자를 입력해야 합니다. 경고 횟수:", warning_count)
        elif guess < target_number:
            print("업! 더 높은 숫자입니다.")
        elif guess > target_number:
            print("다운! 더 낮은 숫자입니다.")
        else:
            print("정답입니다! 숫자는", target_number, "이었습니다. 게임을 종료합니다.")
            break

    except ValueError:
        warning_count += 1
        print("잘못된 입력입니다! 숫자만 입력해야 합니다. 경고 횟수:", warning_count)
```

```
1에서 31 사이의 숫자를 입력하세요: 15
업! 더 높은 숫자입니다.
1에서 31 사이의 숫자를 입력하세요: 23
다운! 더 낮은 숫자입니다.
1에서 31 사이의 숫자를 입력하세요: 19
정답입니다! 숫자는 19 이었습니다. 게임을 종료합니다.
```

[응용] 10번 문제 : 주사위 프로그램 만들기

[문제 조건]

- 이 프로그램은 사용자가 가상의 주사위를 던지는 게임을 시뮬레이션합니다. 사용자는 주사위를 던지고, 그 결과를 확인할 수 있습니다.
- 사용자가 주사위를 던지려고 할 때마다 1부터 6 사이의 무작위 수를 생성하여 결과를 출력합니다.
- 사용자가 주사위를 계속 던지고 싶은지 묻는 코드를 추가하세요.
- 사용자가 게임을 종료하고 싶을 때까지 게임을 반복 실행하게 만드세요.
- 주사위를 던진 횟수와 각 눈금의 출현 빈도를 집계하여 게임이 끝날 때 보여줍니다.

[코드 힌트]

```
while True:
    input("주사위를 던지려면 Enter를 누르세요...")
    result = roll_dice()
    roll_count += 1
    print(f"주사위 결과: {result}")
```

[결과 예시]

```
주사위를 던지려면 Enter를 누르세요...
주사위 결과: 1
그만 주사위를 던지실래요? (y/n):
주사위를 던지려면 Enter를 누르세요...
주사위 결과: 6
그만 주사위를 던지실래요? (y/n):
주사위를 던지려면 Enter를 누르세요...
주사위 결과: 4
그만 주사위를 던지실래요? (y/n): y

게임 종료.
총 주사위를 던진 횟수: 3
```

```python
### (응용) 10번 문제 정답 코드

import random

def roll_dice():
    return random.randint(1, 6)

roll_count = 0

while True:
    input("주사위를 던지려면 Enter를 누르세요...")
    result = roll_dice()
    roll_count += 1
    print(f"주사위 결과: {result}")

    continue_game = input("그만 주사위를 던지실래요? (y/n): ")
    if continue_game.lower() == 'y':
        break

print("₩n게임 종료.")
print(f"총 주사위를 던진 횟수: {roll_count}")
```

```
주사위를 던지려면 Enter를 누르세요...
주사위 결과: 1
그만 주사위를 던지실래요? (y/n):
주사위를 던지려면 Enter를 누르세요...
주사위 결과: 6
그만 주사위를 던지실래요? (y/n):
주사위를 던지려면 Enter를 누르세요...
주사위 결과: 4
그만 주사위를 던지실래요? (y/n): y

게임 종료.
총 주사위를 던진 횟수: 3
```

[응용] 11번 문제 : 로또 숫자 생성기

[문제 조건]

- 로또숫자는 1부터 45개 숫자중 6개를 뽑습니다. 그리고 보너스 숫자가 있습니다.
- 입력 : 생성을 원한는 로또 숫자 갯수를 물어봅니다.
- 출력 : 랜덤으로 원하는 갯수 만큼 로또 숫자를 출력해 줍니다

[코드 힌트]

```
# 사용자로부터 생성할 로또 세트의 개수를 입력받음
number_of_sets = int(input("생성할 로또 숫자 세트의 개수를 입력하세요: "))
```

[결과 예시]

```
생성할 로또 숫자 세트의 개수를 입력하세요: 5
생성된 로또 숫자:
[6, 8, 20, 22, 30, 41]
[3, 5, 20, 31, 36, 42]
[1, 8, 26, 28, 29, 43]
[6, 9, 17, 25, 42, 43]
[3, 7, 11, 16, 31, 40]
```

```python
### (응용) 11번 문제 정답
import random

def generate_lotto_numbers():
    # 1부터 45까지의 숫자 중 무작위로 6개 선택
    return sorted(random.sample(range(1, 46), 6))

# 사용자로부터 생성할 로또 세트의 개수를 입력받음
number_of_sets = int(input("생성할 로또 숫자 세트의 개수를 입력하세요: "))

# 원하는 개수만큼 로또 숫자 세트 생성
print("생성된 로또 숫자:")

# 사용자가 요청한 세트 수만큼 로또 번호를 생성하고 출력
for i in range(number_of_sets):
    lotto_numbers = generate_lotto_numbers()  # 로또 번호 생성
    print(lotto_numbers)  # 생성된 로또 번호 출력
```

```
생성할 로또 숫자 세트의 개수를 입력하세요: 5
생성된 로또 숫자:
[6, 8, 20, 22, 30, 41]
[3, 5, 20, 31, 36, 42]
[1, 8, 26, 28, 29, 43]
[6, 9, 17, 25, 42, 43]
[3, 7, 11, 16, 31, 40]
```

[심화] 12번 문제 : 랜덤 생성 숫자중 가장 큰 숫자 찾기

[문제 조건]

- 입력 : 숫자생성 갯수입력 (범위 2 ~ 100), 범위 제한, 입력 오류시 다시 받기
- 출력 :
- 범위에 맞게 랜덤 정수 2개~ 100개 까지 생성
- 랜덤 생성된 숫자중 가장 큰 숫자 찾아서 출력
- 단 코드는 반복문과 판별문으로 구성할 것
- 리스트 함수의 sort() 등 햐당 기능 함수 사용 금지

[코드 힌트]

```
# 숫자 생성 개수 입력 받기
while True:
    try:
        count = int(input("생성할 숫자의 개수를 입력하세요 (2~100): "))
        if 2 <= count <= 100:
            break
        else:
            print("2에서 100 사이의 값을 입력해야 합니다.")
    except ValueError:
        print("유효한 정수를 입력해 주세요.")

# 랜덤 정수 생성
random_numbers = []
for _ in range(count):
    random_numbers.append(random.randint(1, 1000))

print("생성된 랜덤 숫자들:")
print(random_numbers)
```

[결과 예시]

```
생성할 숫자의 개수를 입력하세요 (2~100): 20
생성된 랜덤 숫자들:
[643, 455, 965, 797, 302, 146, 743, 662, 698, 863, 484, 520, 229, 304, 899, 926, 889, 128, 695, 331]
가장 큰 숫자는: 965
```

```python
### (심화) 12번 문제 정답 코드

import random

def find_max(numbers):
    max_number = numbers[0]  # 첫 번째 숫자를 최대값으로 초기 설정
    for number in numbers:
        if number > max_number:
            max_number = number  # 더 큰 숫자를 찾으면 최대값 갱신
    return max_number

# 숫자 생성 개수 입력 받기
while True:
    try:
        count = int(input("생성할 숫자의 개수를 입력하세요 (2~100): "))
        if 2 <= count <= 100:
            break
        else:
            print("2에서 100 사이의 값을 입력해야 합니다.")
    except ValueError:
        print("유효한 정수를 입력해 주세요.")

# 랜덤 정수 생성
random_numbers = []
for _ in range(count):
    random_numbers.append(random.randint(1, 1000))

print("생성된 랜덤 숫자들:")
print(random_numbers)

# 가장 큰 숫자 찾기
largest_number = find_max(random_numbers)

print("가장 큰 숫자는:", largest_number)
```

```
생성할 숫자의 개수를 입력하세요 (2~100): 20
생성된 랜덤 숫자들:
[643, 455, 965, 797, 302, 146, 743, 662, 698, 863, 484, 520, 229, 304, 899, 926, 889, 128, 695, 331]
가장 큰 숫자는: 965
```

[심화] 13번 문제 : 가위바위보 만들기

[문제 조건]

- 입력 : 가위, 바위, 보 중에 하나 외치기 !
- 출력 : 컴퓨터 랜덤으로 가위, 바위, 보 출력,
- 입력값과 비교해서 승, 패, 무 계산
- 3번 컴퓨터에세 승리시 프로그램 종료
- 몇번의 시도만에 승리했는지 결과값 출력

[코드 힌트]

```
def rock_paper_scissors():
    wins = 0
    attempts = 0

    while wins < 3:
        user_choice = input("가위, 바위, 보 중에 하나를 선택하세요: ")
        if user_choice not in ['가위', '바위', '보']:
            print("잘못된 입력입니다. 다시 입력하세요.")
            continue

        computer_choice = get_computer_choice()
        print(f"컴퓨터의 선택: {computer_choice}")

        result = get_winner(user_choice, computer_choice)
        print("결과:", result)
```

[결과 예시]

```
가위, 바위, 보 중에 하나를 선택하세요: 보
컴퓨터의 선택: 바위
결과: 승리
가위, 바위, 보 중에 하나를 선택하세요: 보
컴퓨터의 선택: 바위
결과: 승리
가위, 바위, 보 중에 하나를 선택하세요: 보
컴퓨터의 선택: 보
결과: 무승부
가위, 바위, 보 중에 하나를 선택하세요: 보
컴퓨터의 선택: 바위
결과: 승리
축하합니다! 4번의 시도 만에 컴퓨터를 누적 3번 이겼습니다.
```

```python
### (심화) 13번 문제 정답 코드

import random

def get_computer_choice():
    return random.choice(['가위', '바위', '보'])

def get_winner(user_choice, computer_choice):
    if user_choice == computer_choice:
        return '무승부'
    elif (user_choice == '가위' and computer_choice == '보') or ₩
     (user_choice == '바위' and computer_choice == '가위') or ₩
      (user_choice == '보' and computer_choice == '바위'):
        return '승리'
    else:
        return '패배'

def rock_paper_scissors():
    wins = 0
    attempts = 0

    while wins < 3:
        user_choice = input("가위, 바위, 보 중에 하나를 선택하세요: ")
        if user_choice not in ['가위', '바위', '보']:
            print("잘못된 입력입니다. 다시 입력하세요.")
            continue

        computer_choice = get_computer_choice()
        print(f"컴퓨터의 선택: {computer_choice}")

        result = get_winner(user_choice, computer_choice)
        print("결과:", result)

        if result == '승리':
            wins += 1

        attempts += 1

        if wins == 3:
            print(f"축하합니다! {attempts}번의 시도 만에 컴퓨터를 누적 3번 이겼습니다.")
            break

rock_paper_scissors()
```

```
가위, 바위, 보 중에 하나를 선택하세요: 보
컴퓨터의 선택: 바위
결과: 승리
가위, 바위, 보 중에 하나를 선택하세요: 보
컴퓨터의 선택: 바위
결과: 승리
가위, 바위, 보 중에 하나를 선택하세요: 보
컴퓨터의 선택: 보
결과: 무승부
가위, 바위, 보 중에 하나를 선택하세요: 보
컴퓨터의 선택: 바위
결과: 승리
축하합니다! 4번의 시도 만에 컴퓨터를 누적 3번 이겼습니다.
```

[심화] 14번 문제 : 크리스마스 트리 만들기

[문제 조건]

- 입력 : 1~100까지 정수 입력 -> 트리의 높이, 범위에 벗어나면 1000(클때) 혹은 1(작을때)로 고정

```
# 문자 입력시 다시 입력하라고 경고 발생
```

- 출력 :

```
ex) 1 입력
 *

ex) 2 입력
 *
***

ex) 3입력
  *
 ***
*****
```

[코드 힌트]

```
# 크리스마스 트리의 높이 입력 받기
while True:
    input_value = input("크리스마스 트리의 높이를 입력하세요 (1-60): ")
    try:
        height = int(input_value)
        if 1 <= height <= 60:
            break
        print("입력한 값이 범위를 벗어났습니다. 1에서 60 사이의 값을 입력해주세요.")
    except ValueError:
        print('정수를 입력하세요!')

draw_christmas_tree(height)
```

[결과 예시]

```
크리스마스 트리의 높이를 입력하세요 (1-60): 100
입력한 값이 범위를 벗어났습니다. 1에서 60 사이의 값을 입력해주세요.
크리스마스 트리의 높이를 입력하세요 (1-60): 15
              *
             ***
            *****
           *******
          *********
         ***********
        *************
       ***************
      *****************
     *******************
    *********************
   ***********************
  *************************
 ***************************
*****************************
```

```python
### (심화) 14번 문제 정답

def draw_christmas_tree(height):
    for i in range(height):
        # 각 라인별로 필요한 별의 개수 계산: 1, 3, 5, ...
        stars = 1 + 2 * i
        # 가운데 정렬을 위한 공백 계산
        spaces = height - i - 1
        print(' ' * spaces + '*' * stars)

# 크리스마스 트리의 높이 입력 받기
while True:
    input_value = input("크리스마스 트리의 높이를 입력하세요 (1-30): ")
    try:
        height = int(input_value)
        if 1 <= height <= 30:
            break
        print("입력한 값이 범위를 벗어났습니다. 1에서 30 사이의 값을 입력해주세요.")
    except ValueError:
        print("정수를 입력하세요!")

draw_christmas_tree(height)


```