

비주얼 프로그래밍

김승태 교수님

과제2

20176359 신수현

[문제 1]

과제1- 7에서 가위바위보 게임을 작성하였다.

이 프로그램을 다음 사항에 맞도록 수정하라.

프로그램 내에 ("가위", "바위", "보")로 구성된 튜플을 하나 만들어라.

프로그램 내에 이 튜플 외에는 "가위", "바위", "보 " 라는 문장이 없게 수정하라. 예를 들어,

if user == "가위" : 와 같이 "가위 " 라는 단어가 소스코드에 있어서는 안된다.

써야 한다면 튜플을 이용하라.

1. 문제의 해결 방안

과제 1- 7와 같이 가위바위보 게임 코드를 작성했습니다. 저번 과제와의 차이점이 있다면, 튜플을 사용하였고, 두번째 if문에서 부터 elif와 else를 이용하여, 지난 과제에서는 else를 사용하지 않고, if elif만을 사용하여 길게 조건문을 쓴것을 else를 통해 생략할 수 있었습니다.

2. 소스코드

```
import random
game = ('가위','바위','보')
n=0
i=0
j=0
l=0
k=0
while True:
    com = random.choice(game)
    n +=1
    if i ==3 or j==3:
        break

    print('라운드',n,')
    print('컴퓨터가 결정했습니다.')
    user = input('무엇을 내시겠습니까?' + str(game))

    if com == user:
        print('컴퓨터는 %s 당신은 %s 비겼습니다' %(com ,user))
```

```
elif (com == game[0] and user ==game[2]) or (com==game[2] and user==game[1]) or  
(com==game[1] and user ==game[0]):
```

```
    print('컴퓨터는 %s 당신은 %s 컴퓨터가 이겼습니다' %(com,user))
```

```
    i+=1
```

```
    k+=1
```

```
    print('컴퓨터: %d 승 %d 패 당신 %d 승 %d 패' %(i,l,j,k))
```

```
else:
```

```
    print('컴퓨터는 %s 당신은 %s 당신이 이겼습니다' %(com,user))
```

```
    j+=1
```

```
    l+=1
```

```
    print('컴퓨터: %d 승 %d 패 당신 %d 승 %d 패' %(i,l,j,k))
```

3.결과 화면 캡처

```
=====
(라운드 1 )
컴퓨터가 결정했습니다.
무엇을 내시겠습니까?('가위', '바위', '보')가위
컴퓨터는 보 당신은 가위 당신이 이겼습니다
컴퓨터: 0 승 1 패 당신: 1 승 0 패
(라운드 2 )
컴퓨터가 결정했습니다.
무엇을 내시겠습니까?('가위', '바위', '보')바위
컴퓨터는 바위 당신은 바위 비겼습니다
(라운드 3 )
컴퓨터가 결정했습니다.
무엇을 내시겠습니까?('가위', '바위', '보')보
컴퓨터는 보 당신은 보 비겼습니다
(라운드 4 )
컴퓨터가 결정했습니다.
무엇을 내시겠습니까?('가위', '바위', '보')바위
컴퓨터는 보 당신은 바위 컴퓨터가 이겼습니다
컴퓨터: 1 승 1 패 당신: 1 승 1 패
(라운드 5 )
컴퓨터가 결정했습니다.
무엇을 내시겠습니까?('가위', '바위', '보')바위
컴퓨터는 가위 당신은 바위 당신이 이겼습니다
컴퓨터: 1 승 2 패 당신: 2 승 1 패
(라운드 6 )
컴퓨터가 결정했습니다.
무엇을 내시겠습니까?('가위', '바위', '보')보
컴퓨터는 보 당신은 보 비겼습니다
(라운드 7 )
컴퓨터가 결정했습니다.
무엇을 내시겠습니까?('가위', '바위', '보')가위
컴퓨터는 보 당신은 가위 당신이 이겼습니다
컴퓨터: 1 승 3 패 당신: 3 승 1 패
>>> |
```

[문제 2]

자기소개서 이름 추출하기

1. 문제 해결 방안

첫 시작으로는, 자기소개서 문장을 받을 수 있는 input함수를 설정하였습니다. 자기소개서의 첫 줄은 항상 “저는 홍길동이라고 합니다. 제 이름은 심청입니다. 저는 이순신입니다.” 셋 중 하나를 무조건 포함해야 하므로, ‘저는’과 ‘제 이름은’ 단어 바로 뒤에 이름이 온다는 것을 확인할 수 있습니다. 또한, 이름은 항상 문자열의 0번째부터 10번째 사이에 존재하므로, 범위는 [0:10]으로 설정했습니다. 따라서 문자열의 index로 이름을 뽑는 식으로 구성했습니다.

2. 소스 코드

while True:

```
    sen = input('문장을 입력하세요.')
```

```
    if '저는' in sen[0:10]:
```

```
        print("이름: ", end="")
```

```
        for i in range(3,6):
```

```
            print(sen[i], end="")
```

```
        print()
```

```
        continue
```

```
    if '제 이름은' in sen[0:10]:
```

```
        print("이름: ", end="")
```

```
        for j in range(6,8):
```

```
            print(sen[j], end="")
```

```
        print()
```

```
        continue
```

3. 결과 화면 캡처

```
===== RESTART: C:\Users\queenSSH\Desktop\학교\수학과\비주얼프로그래밍\파이썬\prac.py
=====
문장을 입력하세요.제 이름은 심청입니다. 이러쿵 저러쿵 저는 화목한 가정
이름: 심청
문장을 입력하세요.저는 홍길동이라고 합니다. 어찌구 저찌구
이름: 홍길동
문장을 입력하세요|
```

[문제3]

수식 계산기

1, 문제 해결 방안

수식에 + 만 있다는 가정과 +, - 둘다 있는 가정을 생각해 보았습니다. - 를 해주어야 하는 숫자들은 sublist에 담고, +를 해주어야 하는 숫자들은 addlist에 담아야겠다는 생각으로 시작했습니다. 먼저, '+'으로 split함수를 이용하여 쪼개고, 쪼개진 문자열 안에 '-'가 있다면 다시 '-'로 split함수를 이용하여 쪼갬니다.

만약 334+123- 32+245- 6- 32 을 예시로 든다면, + 로만 쪼갬을 때 ['334','123- 32','245- 6- 32'] 가 addlist가 됩니다. 이후 '-'로 쪼갬다면, addlist[1] 은 ['123','32']로, addlist[2]는 ['245','6','32']로 쪼개집니다 (구해진 리스트들을 임시로 빈 리스트 strs_list에 추가했습니다). strs_list에서 항상 index가 0인 숫자들은 addlist에 들어가야할 숫자이므로 for문을 이용하여, addlist[i]에 각각의 strs_list[0]으로 덮어주었습니다. 0번째를 제외한 나머지는 다 sublist에 추가되어야하므로, append함수를 이용하여 sublist에 넣어주었습니다.

마지막으로 for문을 이용하여 subNum 과 sumNum을 구하고 sumNum - subNum을 통해 최종 결과를 얻었습니다.

2. 소스 코드

while True:

```
    sublist=[]
```

```
    strs_list=[]
```

```
    sumNum =0
```

```
    subNum=0
```

```
    strs = input("수식을 입력하시오")
```

```
    addlist=strs.split('+')
```

```
    for i in range(0, len(addlist)):
```

```
        if "-" in addlist[i]:
```

```
            strs_list = addlist[i].split('-')
```

```
            addlist[i] = strs_list[0]
```

```
        for j in range(1,len(strs_list)):
```

```
            sublist.append(strs_list[j])
```

```

for i in addlist:
    sumNum += int(i)

for i in sublist:
    subNum += int(i)

print(sumNum - subNum)

```

3. 결과 화면 캡처

```

===== RESTART: C:\Users\queenSSH\Desktop\학교\수학과\비주얼프로그래밍\파이썬\prac.py
=====
수식을 입력하시오12+345+6789
7146
수식을 입력하시오34-12+5
27
수식을 입력하시오334+123-32+245-6-32
632
수식을 입력하시오

```

Windows 정품

[문제 4]

전화번호부

1.문제 해결 방안

첫번째로, 이름을 입력할 input함수를 사용했습니다. 그리고 이름에 add를 입력했을때 딕셔너리에 추가 될 수 있게, if문을 이용하여 코드를 짰습니다(if name == 'add'). 그리고 add가 아닌 나머지 (즉, 이름을 검색했을 때 인 경우를 위해 else를 사용했습니다. 이름을 입력했을때, key값들만 모아놓은 리스트에서 비교하기 위해서(이름이 teldic의 key값에 있는지 확인하기 위해서) list_keys = list(teldic.keys()) 를 이용하여 key만 모아놓은 list를 만들었습니다. for문을 이용하여 이름을 검색하고, 찾았으면 find = True로 설정하고 못찾았을 경우에는 find = False를 기본값으로 하여 '찾을 수 없습니다' 를 출력할 수 있게끔 설정 했습니다.

2. 소스 코드

```
teldic = {'홍길동': '010- 4444- 5555', '김중양': '010- 9191- 8181', '심청': '010- 3232- 5454'}
```

```
while True:
```

```
    list_keys = list(teldic.keys())
```

```
    find = False
```

```
    name = input('이름은?')
```

```
    if name == 'add':
```

```
        add_name = input('이름?')
```

```
        add_phone = input('번호?')
```

```
        teldic[add_name] = add_phone
```

```
        print('%s 전화번호가 추가되었습니다' % add_name)
```

```
    else:
```

```
        for i in range(len(list_keys)):
```

```
            if name in list_keys[i]:
```

```
                print(list_keys[i], ": ", teldic[list_keys[i]])
```

```
                find = True
```

```
        if find == False:
```

```
            print('찾을 수 없습니다')
```

3. 결과 화면 캡처

```
===== RESTART: C:\Users\queenSSH\Desktop\학교\수학과\비주얼프로그래밍\파이썬\prac.py
=====
```

```
이름은?홍길동
```

```
홍길동 : 010-4444-5555
```

```
이름은?홍길
```

```
홍길동 : 010-4444-5555
```

```
이름은?홍길서
```

```
찾을 수 없습니다
```

```
이름은?add
```

```
이름?강감찬
```

```
번호?010-5555-6666
```

```
강감찬 전화번호가 추가되었습니다
```

```
이름은?강감찬
```

```
강감찬 : 010-5555-6666
```

```
이름은?
```

Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증

[문제5]

1. 문제 해결 방안

먼저 천단위로 ';' 를 표시하는 함수를 생성했습니다. 문자열을 거꾸로 읽어서 3마다 (나누기 3했을 때 나머지가 0이 되는 때) ';'를 붙여주는 식으로 코딩했습니다. 한글로 숫자를 나타내기 위해 숫자가 1,2,3,4,5,6,7,8,9 일때마다 불러낼 한글 숫자를 적은 함수를 생성하고, 백천만억단위에 제일 처음이 1일 경우에 '일'을 생략하도록 설정했습니다. (예를들어 일만오천이 되지 않게끔) temp에는 일시적으로 숫자를 넣어주고, 최종적으로 ret문자열에 한국어로 변환된 숫자를 넣어, 함수마다 return ret을 했습니다. 그리고 자리수를 세기위해 cnt=0을 초기로 설정하여, 자리수가 올라가면서 +1씩 해주었습니다.

2. 소스 코드

```
def insertComma(num):
```

```
    numString = str(num)
```

```
    ret = ""
```

```
    cnt = 0
```

```
    for number in reversed(numString):
```

```
        if cnt != 0 and cnt % 3 == 0:
```

```
            ret = number + ';' + ret
```

```
        else:
```

```
            ret = number + ret
```

```
        cnt = cnt + 1
```

```
    return ret
```

```
def numberToKorean(num):
```

```
    if num == 1:
```

```
        return '일'
```

```
    if num == 2:
```

```
        return '이'
```

```
    if num == 3:
```

```
        return '삼'
```

```
    if num == 4:
```

```
        return '사'
```

```
    if num == 5:
```

```
        return '오'
```

```
    if num == 6:
```



```

        return '육'
    if num == 7:
        return '칠'
    if num == 8:
        return '팔'
    if num == 9:
        return '구'
    return "

```

```

def representSmallNumberByKorean(num):

```

```

    numString = str(num)
    length = len(numString)
    temp = ""
    ret = ""
    cnt = 0
    if length == 4:
        temp = numberToKorean(int(numString[cnt]))
        if temp == ":":
            pass
        elif temp == '일':
            ret = ret + '천'
        else:
            ret = ret + temp + '천'
        cnt = cnt + 1
    if length >= 3:
        temp = numberToKorean(int(numString[cnt]))
        if temp == ":":
            pass
        elif temp == '일':
            ret = ret + '백'
        else:
            ret = ret + temp + '백'
        cnt = cnt + 1
    if length >= 2:

```

```

temp = numberToKorean(int(numString[cnt]))
if temp == "":
    pass
elif temp == '일':
    ret = ret + '십'
else:
    ret = ret + temp + '십'
cnt = cnt + 1
if length >= 1:
    ret = ret + numberToKorean(int(numString[cnt]))

return ret

```

```

def representBigNumberByKorean(num):

```

```

    numString = str(num)
    length = len(numString)
    temp = ""
    ret = ""
    cnt = 0

```

```

if length >= 9:

```

```

    temp = representSmallNumberByKorean(int(numString[cnt : length - 8]))
    if temp != "":
        ret = ret + temp + '억'
    cnt = length - 8

```

```

if length >= 5:

```

```

    temp = representSmallNumberByKorean(int(numString[cnt : length - 4]))
    if temp != "":
        if temp == '일' and length == 5:
            ret = ret + '만'
        else:
            ret = ret + temp + '만'
    cnt = length - 4

```

```
ret = ret + representSmallNumberByKorean(int(numString[cnt:]))
```

```
return ret
```

```
while True:
```

```
    num = input('숫자는? ')

```

```
    print(insertComma(num))
```

```
    print(representBigNumberByKorean(num))
```

```
    print("")
```

3. 결과 화면 캡처

```
===== RESTART: C:\Users\queenSSH\Desktop\학교\수학과\비주얼프로그래밍\연습.py ==
=====
```

```
숫자는? 123456789
```

```
123,456,789
```

```
일억이천삼백사십오만육천칠백팔십구
```

```
숫자는? 10000006
```

```
10,000,006
```

```
천만육
```

```
숫자는? |
```