



가천대학교 컴퓨터공학전공 '파이썬' 기말고사 대비 추가 연습 5제

본 연습문제는 가천대학교 컴퓨터공학전공의 '파이썬' 기말고사를 대비하기 위함으로 제작되었습니다.

모든 문제는 월, 화 11 ~ 12:15의 오기욱 교수님 수업 기준으로 제작되었습니다.

본 연습문제의 저작권은 임규연 (lky473736) 아래에 있으며, 모든 문제의 예시 풀이 소스 코드는 '[lky473736/practice-learningpython](https://github.com/lky473736/practice-learningpython)'와 '[lky473736/problemsolving](https://github.com/lky473736/problemsolving)'에서 MIT 라이선스가 적용된 소스코드입니다.

* 질문 및 오류 제보 : lky473736@icloud.com

Profile : <https://url.kr/rvq7sy>

GitHub : <https://github.com/lky473736>

중간고사 대비 연습 5제 : <https://github.com/lky473736/problemsolving/blob/main/self-made/quiz%20for%20midterm%20-%20learningpython%20.pdf>

기말고사 대비 연습 15제 : <https://github.com/lky473736/problemsolving/blob/main/self-made/quiz%20for%20endterm%20-%20learningpython.pdf>

본 문서의 구성은 아래와 같습니다.

1p : 개괄

2p : 그동안의 여담 및 추가 연습문제에 관한 설명

3p ~ : 연습 5제

2023년 6월 3일 토요일

가천대학교 컴퓨터공학전공 '파이썬' 기말고사 대비 추가 연습 5제

그동안의 여담 및 추가 연습문제에 관한 설명)

안녕하세요, 저는 컴퓨터공학전공의 임규연입니다.

요즘 저는, '이곳에서 정말 멋진 친구들과 같이 꿈을 키워나가고 있구나'라는 생각에 모두에게 고마움을 느끼며 하루하루를 보내고 있습니다. 현재 연습 15제의 조회수는 800회를 웃돌고, 제 메일에는 코드와 문제에 관한 질문들이 50통 가까이 쌓이고 있습니다.

이 자리를 빌어서 제 자작문제에 많은 관심을 주셔서 진심으로 감사드립니다.

파이썬은 기본적으로 제공하는 내장 함수가 다른 어떤 언어보다도 많습니다. 따라서 편하면서도 직관적인 것 때문에 저는 프로그래밍 언어 중 파이썬을 가장 좋아합니다.

이건 엄연히 제 생각이지만, 파이썬을 공부할 때 만큼은 내장 함수를 한번 직접 구현해 보는 것이 앞으로 다른 언어를 배울 때에도 훨씬 도움이 된다고 생각합니다. 내장 함수의 메커니즘을 스스로 작성해보신 후에 어떻게 하면 코드가 간결해지고 보기 편할 지를 거듭적으로 고민하신다면 분명 실력이 느낄 겁니다.

예를 들어서,

```
slist = [1, 2, 3]
print (sum(slist))
```

라는 코드를 실행시켜보면, sum 함수가 $1 + 2 + 3$ 을 해줘서 6이 출력됩니다. 하지만 sum을 사용하지 않고 스스로가 직접 sum 함수의 메커니즘을 구현해 보는 것이죠. 구현할 땐, sum 이외의 어떠한 내장함수를 사용해도 괜찮습니다. sum만 사용하지 않고 sum을 구현시키는 겁니다. 아래와 같이 말입니다.

```
slist = [1, 2, 3]
s = 0

for i in range (len(slist)) :
    s = s + slist[i]

print (s)
```

따라서 본 추가 연습문제에서는 자주 사용되는 내장 함수 12개의 메커니즘을 스스로 구현하는 문제를 Q1에 배치하였습니다. Q2부터는 객체와 class, tkinter, 함수 심화를 범위로 하여 문제를 구성하였습니다.

Q2부터는 문제가 좀 어려울 수도 있습니다. 코드의 줄개수가 최소한 70이 넘어가도록 복잡한 상황을 구현하였습니다. 또한 Q4부터는 Hint.가 제공되지 않습니다. 도저히 문제 해결이 되지 않으시면 풀이 예시 및 참고를 적극적으로 활용해주세요. 또한 언제든지 제 메일 (lky473736@icloud.com)으로 문의 주시면 친절히 답변해드리겠습니다.

저는 앞으로도 lky473736/problemsolving을 통해 저희가 배울 sqlite, C, C++, Java, HTML, Javascript, Linux 명령어 등 여러 가지 언어들을 대비할 수 있는 연습문제를 배포하려고 합니다. (해당 학기 시험기간 2주 전에 배포)

잘 부탁드립니다. 감사드립니다.

2023년 6월 3일 토요일

가천대학교 컴퓨터공학전공 '파이썬' 기말고사 대비 추가 연습 5제

Q1) 내장 함수 구현하기

본 문제에서 구현할 내장 함수의 목록은 아래와 같다.

abs, all, any, eval, sum, len, list/tuple, max/min, map, enumerate, filter, sort

*** Q1의 풀이 예시 및 참고는 <https://github.com/lky473736/problemsolving/tree/main/solution%20for%20self-made/quiz%20for%20learningpython-plus/S-01>에 있습니다.**

1) abs()는 어떠한 수의 절댓값을 반환한다.

```
ex) print (abs(-2004))  
>>> 2004
```

abs()를 구현하시오.

2) all()은 sequence의 모든 component가 참일 때 True를 반환한다. 마치 논리 연산자 and라고 생각할 수 있다.

```
ex) slist = [1, 2, 3, 4, 0]  
print (all(slist))  
>>> False (0은 LOW이기 때문에 False가 나옴)
```

숫자로 구성된 list를 인수로 받는 all()를 구현하시오.

3) any()는 all()과는 다르게 논리 연산자 or라고 생각할 수 있다. all()의 예시에서 all(slist) 대신 any(slist)를 집어넣고 실행시키면 True가 출력된다.

숫자로 구성된 list를 인수로 받는 any()를 구현하시오.

(뒷장에 계속.)

4) eval()은 문자열로 이루어진 수식을 계산해준다.

```
ex) formula1 = str(10*3)
print (eval(formula1))
>>> 30
```

한 자리수와 한 자리 수의 연산식을 인수로 받는 **eval()**를 구현하시오. 이때 연산식엔 띄어쓰기가 없다.

5) sum()은 **sequence**의 모든 **component**를 합산해준다.

```
ex) slist = [1, 2, 3, 4, 5]
print (sum(slist))
>>> 15
```

sum()을 구현하시오.

6) len()은 어떠한 객체의 길이를 계산해준다. 길이란 객체가 가지고 있는 **component**의 갯수나 문자의 수, 자료의 수 등이 될 수 있다.

```
ex) k = "gachonuniv"
print (len(k))
>>> 10
```

list의 길이를 재는 **len()**을 구현하시오.

(뒷장에 계속.)

7) tuple()과 list()는 어떠한 객체를 tuple 혹은 list로 생성해준다. 참고로, tuple과 list는 서로 변환이 가능하다.

```
ex1) stuple = (1, 2, 3, 4, 5)
stuple = list(stuple)
print (stuple)
>>> [1, 2, 3, 4, 5]
```

```
ex2) k = "macbook"
k = list(k)
print (k)
>>> ['m', 'a', 'c', 'b', 'o', 'o', 'k']
```

ex2의 list()를 구현하시오.

8) max()는 객체의 최댓값, min()는 객체의 최솟값을 반환한다.

```
ex) slist = [1, 2, 3, 4, 5]
print(max(slist))
print(min(slist))
>>> 5
1
```

list를 인수로 받는 max()와 min()을 구현하시오.

9) map()은 반복이 가능한 객체 (리스트, 튜플 등)의 각 항목에 주어진 함수를 적용한 후 결과를 반환해준다. 이때 map의 첫 인수는 함수명, 두번째 인수는 반복 가능한 객체이다. 어려워니 예시를 보자.

```
ex) def square (n) :
    return n ** 2
klist = [1, 2, 3, 4, 5]
result = list(map(square, klist)) # 각 component가 square라는 함수를 전부 적용되었다.
print (result)
>>> [1, 4, 9, 16, 26]
```

map()을 구현하시오.

(뒷장에 계속.)

10) enumerate()는 **sequence** 객체를 입력받은 후, (**component**의 위치, **component**)의 **list**를 반환한다. 이것도 어려우니 예시를 보자.

```
ex) slist = [1, 2, 3, 4, 5]
print (list(enumerate(slist)))
>>> [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5)] # 각 component의 위치를 앞에, 그
component를 뒤에 써줌
```

enumerate()를 구현하시오.

11) filter()은 특정 조건을 만족하는 **component**를 뽑아낼 수 있다.

```
ex) def condition(x) :
    return x > 3

flist = [1, 2, 3, 4, 5]
print(list(filter(condition, flist))) # filter (조건을 반환하는 함수 이름, sequence 이름)
>>> [4, 5]
```

filter()를 구현하시오.

12) sort()는 **sequence**를 오름차순 정렬해준다.

```
ex) slist = [1, 4, 3, 0, 2]
sort(slist)
print(slist)
>>> [0, 1, 2, 3, 4]
```

sort()을 구현하시오.

가천대학교 컴퓨터공학전공 '파이썬' 기말고사 대비 추가 연습 5제

Q2) 야옹

`random` 라이브러리의 `random.randint`를 이용하여 0~100 범위를 가지는 수를 4개 선언해라. 또한 수를 연결하는 연산자 `+`, `-`, `*`와 수 4개가 이를 수 있는 모든 수식의 경우의 수를 콘솔에 출력해라. (이때 수식은 전부 문자열 취급한다.)

수식의 예시) `30 + 21 - 49 * 92, 1 * 2 * 99 - 99 ...`

`random` 라이브러리로 만들어 낸 수식 중 하나를 컴퓨터가 뽑고, 뽑아진 수식을 tk 창에 표시하여 사용자가 문제를 풀어 입력할 수 있도록 해라. 사용자가 답을 입력할 경우 **좋아요 고양이 이미지**를 출력하고, 오답을 입력할 경우 **시무룩 고양이 이미지** 그리고 **정답**을 출력해라.

프로그램의 예시) <https://www.youtube.com/watch?v=35m59g5aLTQ>

좋아요 고양이 이미지) <https://github.com/lky473736/problemsolving/blob/main/solution%20for%20self-made/quiz%20for%20learningpython-plus/good-uploading.jpeg>

시무룩 고양이 이미지) <https://github.com/lky473736/problemsolving/blob/main/solution%20for%20self-made/quiz%20for%20learningpython-plus/sad-uploading.jpeg>

- Hint.

- 모든 수식의 경우의 수를 출력하는 부분은 결국엔 반복문을 계속 거듭하여 쓰는 것.
- 수식을 문자열 취급하는 이유는 `eval` 때문이다.
- 그래서, 고양이는 액체인가, 고체인가?
- 풀이 예시 및 참고 : <https://github.com/lky473736/problemsolving/blob/main/solution%20for%20self-made/quiz%20for%20learningpython-plus/S-02.py>

가천대학교 컴퓨터공학전공 '파이썬' 기말고사 대비 추가 연습 5제

Q3) 숫자 6개 중 4개가 맞는 복권

복권은 1~45까지의 수 중 6개의 수를 맞춰 상금을 타는 게임이다.

tkinter로 메모장을 구현한다. 이때 menu는 '메뉴' 하나로 구성되며, '메뉴' 안에 command는 '프로그램 정보'와 '숫자 작성 완료'로 구성한다. '프로그램 정보'를 누를 시엔 '숫자 6개 중 4개의 방법'이라는 창이 뜬다.

프로그램을 실행시키면 처음에 특정 경로에 특정 이름으로 파일을 저장하라는 알림창이 뜬다. 사용자는 메모장에 1~45 사이의 6개의 숫자를 줄바꿈하여 작성한다. '숫자 작성 완료'를 누르면 `asksaveasfile()`이 작동되어 특정 경로에 특정 이름으로 파일을 저장한다.

프로그램은 파일 유무를 따져, 만일 파일이 존재하면 파일을 연다. `random` 라이브러리를 통해 6개의 숫자 중 4개의 숫자를 랜덤으로 고른 후에 `slist`에 `append`한다. `random.randint()`를 통해 1~45 사이의 두 개의 수를 `slist`에 `append`한다. (이때 두 개의 수는 기존 4개의 숫자와 달라야 한다.) 마지막으로 `slist`와 컴퓨터가 고른 4개의 수를 출력한다.

위의 설명을 읽고 프로그램을 만드시오.

- Hint.

- 파일의 유무는 `os` 라이브러리를 사용해야 한다.
- 사용자 지정 함수에서 특정 변수가 지역 변수인지 전역 변수인지 잘 구분하자.
- `main()` 함수에서 먼저 `tk` 창을 만든 다음에 `__main__`으로 처리해주는게 깔끔하다.
- 풀이 예시 및 참고 : <https://github.com/lky473736/problemsolving/blob/main/solution%20for%20self-made/quiz%20for%20learningpython-plus/S-03.py>

가천대학교 컴퓨터공학전공 '파이썬' 기말고사 대비 추가 연습 5제

Q4) 회원가입 및 로그인 시스템

tkinter와 class로 회원가입 및 로그인 시스템을 구현한다. 아래처럼 tk 창을 구현하며, 만일 아이디 혹은 비밀번호가 일치하지 않으면 '아이디 혹은 비밀번호가 일치하지 않습니다.' 알림창이 뜨게 한다. 아이디와 비밀번호가 기존의 계정과 일치하면 시스템을 종료한다.



회원가입 버튼을 누르면 콘솔에 새로운 아이디를 입력하고, 그 후 비밀번호를 입력하게 한다. 이 때 아이디는 기존의 계정과 다르게 구성되어야 하며, 비밀번호는 8자 이상으로 구성되어야 한다. 기존의 계정과 같은 이름의 아이디를 입력할 경우 '다른 이름의 아이디를 사용해주세요.'를 출력 후 아이디를 다시 입력하게 한다. 또한 8자 이상이 아니게 입력될 시 '비밀번호는 8자 이상으로 구성하여 주십시오.'를 출력 후 비밀번호를 다시 입력하게 한다.

위의 설명을 읽고 프로그램을 만드시오.

프로그램의 예시) <https://www.youtube.com/watch?v=MUTYXU-5R48>

- 풀이 예시 및 참고 : <https://github.com/lky473736/problemsolving/blob/main/solution%20for%20self-made/quiz%20for%20learningpython-plus/S-04.py>

가천대학교 컴퓨터공학전공 '파이썬' 기말고사 대비 추가 연습 5제

Q5) 몬티홀은 진짜일까?

몬티홀 게임에 대한 설명은 아래 영상을 참고하세요.

<https://www.youtube.com/watch?v=AXB6r-hjsig>

세 개의 문이 있고, 문 뒤엔 고가의 슈퍼카와 2마리의 염소가 있다. 당신은 세 개의 문 중에서 하나를 택하였다. 그러더니 어딘가에서 임규연이 슬그머니 나와서 염소가 들어있는 문 중 하나를 보여준다. 임규연은 당연히 슈퍼카가 어디에 있는지를 알고 있다. 여기서 당신은 선택한 문을 바꿀 수 있다. 바꿀 것인가, 아니면 당신의 퍼스트 초이스를 믿을 것인가?

슈퍼카가 나올 확률은 반반 같지만, 사실은 그것이 아니라는 것이 놀랍다. 정답은, 문을 바꾸었을 때 차를 얻을 확률이 $2/3$ 이라는 것이다.

당신이 알고 있는 모든 파이썬 지식을 동원하여 몬티홀 게임을 구현해보자. 구현 후 반복 문을 통하여 컴퓨터에게 문을 바꾸는 경우만을 10,000번 시켜본 후 최종적으로 슈퍼카를 얻는 횟수와 얻지 못하는 횟수의 비율을 출력하게 하여 정말 확률이 $2/3$ 가 맞는지를 증명해라.

- 풀이 예시 및 참고 : <https://github.com/lky473736/problemsolving/blob/main/solution%20for%20self-made/quiz%20for%20learningpython-plus/S-05.py>

- 고생하셨습니다. -