

Python - 복습 Note

파이썬 복습 Note

사용자로부터 변수 x , 변수 y 를 입력받아 더하고 곱하는 프로그램을 작성하세요.

실행 :

x 값을 입력해주세요 : 20

y 값을 입력해주세요 : 30

결과 :

50

600

파이썬 복습 Note

사용자로부터 5개의 값을 입력 받아 리스트에 넣고 생성된 리스트에서 두번째, 네번째 값을 삭제한 뒤 출력하세요.

실행:

첫번째 값을 입력해주세요 : 10

두번째 값을 입력해주세요 : 20

세번째 값을 입력해주세요 : 30

네번째 값을 입력해주세요 : 40

다섯번째 값을 입력해주세요 : 50

[10, 30, 50]

파이썬 복습 Note

다음과 같은 number = [[1,2,3],[4,5,[7,8]]] 리스트를 만들고 모든값에 직접 접근하여 값을 모두 0으로 변경한 뒤 출력하세요.

실행:

```
[[0,0,0],[0,0,[0,0]]]
```

파이썬 복습 Note

다음과 같은 fruit = ["사과", "배", "딸기", "옥수수"] 리스트를 만들고 유저에게 두가지 과일을 입력 받아 첫번째 입력한 과일을 찾아 두번째 입력한 과일로 변경한 뒤 출력하세요.

실행 :

바꿀 과일을 입력하세요 : 배

어떤 과일로 바꾸실래요 : 복숭아

['사과', '복숭아', '딸기', '옥수수']

파이썬 복습 Note

문자를 입력받아 공백을 모두 제거하고 출력하세요.

실행 :

아무 문자나 입력해주세요 : 안녕하세요 반갑습니다.

안녕하세요반갑습니다.

파이썬 복습 Note

이름과 점수 3개를 입력 받아 아래와 같이 출력하세요.

실행 :

이름을 입력해 주세요 : **홍길동**

첫번째 점수를 입력해 주세요 : **80**

두번째 점수를 입력해 주세요 : **70**

세번째 점수를 입력해 주세요 : **60**

저의 이름은 홍길동 이고 총점은 210 입니다.

파이썬 복습 Note

range 함수를 이용하여 사용자가 입력한 수까지 2의 배수 값을 넣은 리스트를 만들고 리스트의 맨 마지막에 사용자가 입력한 추가해 주세요

실행 :

숫자를 입력해주세요 : 10

[2, 4, 6, 8, 10, 10]

실행 :

숫자를 입력해주세요 : 20

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 20]

파이썬 복습 Note

이름, 나이, 연락처를 입력받아 딕셔너리를 만들어 출력해주세요

실행 :

이름을 입력해주세요 : **홍길동**

나이를 입력해주세요 : **27**

연락처를 입력해주세요 : **010-3023-1223**

{'이름': '홍길동', '나이': '27', '연락처': '010-3023-1223'}

파이썬 복습 Note

두사람의 이름, 나이, 연락처를 입력받아 각각 딕셔너리를 만들어 리스트에 넣어주세요

실행 :

이름을 입력해주세요 : 홍길동

나이를 입력해주세요 : 27

연락처를 입력해주세요 : 010-3023-1223

이름을 입력해주세요 : 이몽룡

나이를 입력해주세요 : 30

연락처를 입력해주세요 : 010-3030-4434

```
[{'이름': '홍길동', '나이': '27', '연락처': '010-3023-1223'}, {'이름': '이몽룡', '나이': '30', '연락처': '010-3030-4434'}]
```

파이썬 복습 Note

사용자로부터 점수를 3개 입력 받아 모든 점수가 65점보다 클 경우 합격 아닐 경우 불합격을 출력하세요

단, 0~100이 아닌 숫자가 입력된 경우 “잘못된 점수가 입력되었습니다”를 출력하세요

실행 :

첫번째 점수를 입력해주세요 : 120

두번째 점수를 입력해주세요 : 90

세번째 점수를 입력해주세요 : 80

잘못된 점수가 입력되었습니다

실행 :

첫번째 점수를 입력해주세요 : 80

두번째 점수를 입력해주세요 : 90

세번째 점수를 입력해주세요 : 75

합격

실행 :

첫번째 점수를 입력해주세요 : 50

두번째 점수를 입력해주세요 : 60

세번째 점수를 입력해주세요 : 90

불합격

파이썬 복습 Note

```
fruit = ['사과', '오렌지']
```

```
vegetable = ['당근', '호박']
```

위와 같은 리스트 두개를 만들고 유저로부터 카테고리나 상품명을 입력 받아 카테고리가 과일일때는 fruit 리스트에 카테고리가 채소일때는 vegetable 리스트에 상품을 추가하고 리스트의 모든 내용을 출력해주세요.

단, 카테고리명이 채소나 과일이 아닐경우 “존재하지 않는 카테고리입니다.”

이미 등록되어있는 경우 “이미 등록된 과일입니다.” or “이미 등록된 채소입니다”를 출력해주세요.

실행 :

등록할 카테고리를 선택해주세요 (과일, 채소) : **채소**

등록할 채소를 입력해주세요 : **당근**

이미 등록된 채소 입니다.

실행 :

등록할 카테고리를 선택해주세요 (과일, 채소) : **야채**

등록할 야채를 입력해주세요 : **당근**

존재하지 않는 카테고리입니다.

실행 :

등록할 카테고리를 선택해주세요 (과일, 채소) : **과일**

등록할 과일을 입력해주세요 : **바나나**

['사과', '오렌지', '바나나']

파이썬 복습 Note

(1) $x = [3, 6, 9, 20, -7, 5]$ 의 값의 모든 요소에 10을 곱하여 저장한 뒤 출력하세요

실행 :

`[30, 60, 90, 200, -70, 50]`

(2) $y = \{\text{"math": 70, "science": 80, "english": 20}\}$ 의 값의 모든 요소에 10을 더하여 저장한 뒤 출력하세요

실행 :

`{'math': 80, 'science': 90, 'english': 30}`

(3) 숫자를 입력 받고 입력 받은 수의 구구단을 출력하세요

실행 :

몇단을 출력하시겠습니까? **3**

`3 * 1 = 3`

`3 * 2 = 6`

...

`3 * 9 = 27`

파이썬 복습 Note

(1) word = ["school", "game", "piano", "science", "hotel", "mountain"] 중 글자수가 6 이상인 문자를 모아 새로운 리스트를 생성하세요

실행 :

```
['school', 'science', 'mountain']
```

(2) 구구단을 1단부터 9단까지 출력하세요

실행 :

```
1 * 1 = 1
```

```
1 * 2 = 2
```

```
1 * 3 = 3
```

```
....
```

```
9 * 9 = 81
```

파이썬 복습 Note

(1) [3, 6, 9, 20, -7, 5] 리스트를 sort 함수를 사용하지 않고 for문을 활용하여 오름차순으로 정렬해주세요.

실행 :

[-7, 3, 5, 6, 9, 20]

(2) 1-100까지 숫자 중 3과 5의 공배수일 경우 "3과 5의 공배수" 나머지 숫자 중 3의 배수일 경우 "3의 배수" 나머지 숫자 중 5의 배수일 경우 "5의 배수" 모두 해당되지 않을 경우 숫자 를 출력하세요

실행 :

1

2

3의 배수

...

14

3과 5의 공배수

16

파이썬 복습 Note

사용자로부터 숫자를 계속 입력받다가 0을 입력하면 합계를 출력해주세요

실행 :

값을 입력해주세요 : 30

값을 입력해주세요 : 20

값을 입력해주세요 : 50

값을 입력해주세요 : 40

값을 입력해주세요 : 30

값을 입력해주세요 : 0

170

파이썬 복습 Note

가위바위보 게임 만들기

1~3 을 입력하면 게임을 진행하고, 이외의 숫자를 입력하면 게임종료하고,
진행 횟수와 승리 횟수를 출력해주세요.

실행 :

가위(1), 바위(2), 보(3) 을 입력해주세요 : 3

유저 : 보, 컴퓨터 : 보

가위(1), 바위(2), 보(3) 을 입력해주세요 : 2

유저 : 바위, 컴퓨터 : 보

가위(1), 바위(2), 보(3) 을 입력해주세요 : 1

유저 : 가위, 컴퓨터 : 보

가위(1), 바위(2), 보(3) 을 입력해주세요 : 4

게임종료 (전체:3 ,승리:1)

random 모듈 사용방법 - 랜덤 값 받기

```
import random #random 모듈을 가져온다
```

```
random.random() # 0.00202302032
```

```
random.randint(1,3) # 3
```

파이썬 복습 Note

(1) word = ["school", "game", "piano", "science", "hotel", "mountian"] 중 글자수가 6 이상인 문자를 모아 새로운 리스트를 생성하세요 (리스트 컴프리헨션을 사용해주세요)

실행 :

['school', 'science', 'mountian']

(2) word = ["school", "game", "piano", "science", "hotel", "mountian"] 리스트의 글자수가 들어가있는 새로운 리스트를 생성하세요 (리스트 컴프리헨션을 사용해주세요)

실행 :

[6, 4, 5, 7, 5, 8]

파이썬 복습 Note

아래 두 리스트를 곱해 새로운 리스트 C를 만드세요

a = [[10, 20], [30, 40], [50, 60]]

b = [[2, 3], [4, 5], [6, 7]]

실행 :

[[20, 60], [120, 200], [300, 420]]

파이썬 복습 Note

아래의 학습코드를 가지고 a 리스트가 [[1,2],[3,4],[5,6]]와 같이 만들어지도록 수정하세요

```
--- 학습코드
a = []
for i in range(3):
    temp = []
    for j in range(2):
        temp.append(0)
    a.append(temp)
print(a)
---
```

실행 :

[[1,2],[3,4],[5,6]]

파이썬 복습 Note

별 출력하기

사각형별	계단식별	대각선별	계단식(역순)	계단식(역순)
* * * * *	*	*	* * * * *	* * * * *
* * * * *	* *	*	* * * *	* * * *
* * * * *	* * *	*	* * *	* * *
* * * * *	* * * *	*	* *	* *
* * * * *	* * * * *	*	*	*

파이썬 복습 Note

아래 예시코드의 while 문을 완성하여 사용자가 입력한 숫자만큼 출력해주세요. (break 사용)

```
user = int(input("숫자를 입력하세요:"))  
cnt = 0  
while True:  
    (여기 코드를 완성해 주세요)
```

실행 :

숫자를 입력하세요: 20

0

1

...

19

20

파이썬 복습 Note

아래 예시코드의 for 문을 완성하여 사용하여 사용자가 입력한 숫자까지의 짝수를 출력하기 (continue 사용)

```
user = int(input("숫자를 입력하세요:"))  
for i in range(user+1):  
    (여기 코드를 완성해 주세요)  
    print(i)
```

실행 :

숫자를 입력하세요: 20

0

2

...

16

18

20

파이썬 복습 Note

가위바위보 게임 업그레이드

이전에 만든 가위바위보 게임을 총 게임횟수와 승리횟수를 게임을 다시 실행해도 유지되도록 수정하세요

파이썬 복습 Note

성적관리 프로그램 개발

메뉴를 선택해주세요 1 - 입력, 2 - 조회, 3 - 삭제, 0 - 종료) : 1

이름 : 최재진

수학 : 30

과학 : 40

영어 : 50

메뉴를 선택해주세요 1 - 입력, 2 - 조회, 3 - 삭제, 0 - 종료) : 2

[0] 이름 : 다비드, 수학 : 50, 과학 : 70, 영어 : 80

[1] 이름 : 최재진, 수학 : 30, 과학 : 40, 영어 : 50

메뉴를 선택해주세요 1 - 입력, 2 - 조회, 3 - 삭제, 0 - 종료) : 3

[0] 이름 : 이민호, 수학 : 40, 과학 : 50, 영어 : 30

[1] 이름 : 이진범, 수학 : 30, 과학 : 40, 영어 : 50

삭제할 번호를 입력해주세요 : 0

메뉴를 선택해주세요 1 - 입력, 2 - 조회, 3 - 삭제, 0 - 종료) : 0

종료되었습니다

다음과 같이 동작하는 프로그램을 개발하세요

프로그램 종료 후 다시 실행할 때 이전에 입력했던 값을 파일에서 다시 불러와주세요

파이썬 복습 Note

리스트와 찾고 싶은 값을 모두 입력하면 입력한 값의 인덱스(위치)를 반환하는 함수 만들기

실행 :

```
lis = [1, 2, 3, 1, 4, 2, 1]
```

```
allindex(lis, 1)
```

[0, 3, 6]

파이썬 복습 Note

가변인수와 고정인수를 사용해 모든 값을 더하거나 곱하는 함수 작성

실행 :

```
calc("+", 1, 2, 3, 4, 5)
```

15

```
calc("*", 1, 2, 3, 4, 5)
```

120

파이썬 복습 Note

(1) numbers = [12, 32, 55, 12, 32, 4, 86, 50] 리스트에서 60보다 크면 합격 50~60점까지는 대기, 50보다 작으면 불합격이 들어간 리스트를 만드세요.

(람다와 map 이용)

(2) 파일명이 들어가있는 다음 리스트에서 files = ["memo.txt", "1.jpg", "32.png", "23.jpg", "223.jpg"] 리스트에서 확장자가 jpg 파일만 골라내 리스트를 만드세요.

(람다와 filter 이용)

find("문자열") 함수 사용 : 해당 문자열이 있을 경우 문자열의 인덱스, 없을 경우 -1 을 반환하는 함수

파이썬 복습 Note

Car 클래스를 만드세요

- 객체 생성시 차 이름, 배기량, 생산년도를 입력 받고 인스턴스 속성으로 만들어주세요
- 차이름을 확인하는 함수와 변경하는 함수를 만드세요
- 배기량에 따라 1000CC 보다 작으면 소형
1000CC 이상 2000CC 이하 중형
2000CC 보다 크면 대형을 출력하는 인스턴스 함수를 만드세요
- 객체 생성시마다 등록된 차량 개수를 기록하는 클래스 속성을 만들어주세요
- 총 등록된 차량개수를 출력하는 클래스 함수를 만드세요

파이썬 복습 Note

(1) Character 클래스를 만들어 주세요

Character 클래스는 Health 속성을 추가해주세요 (생성시 200)

Character 클래스는 Move() 메서드를 추가하고 메서드 사용시 Health 가 -10이 됩니다.

Character 클래스는 Rest() 메서드를 추가하고 메서드 사용시 Health가 +10 됩니다.

현재 Health를 알수있는 checkHealth() 메서드를 추가해주세요

(2) Knight와 Healer 클래스를 만들어주세요

Knight와 Healer 클래스는 Character 클래스를 상속합니다.

Knight 클래스는 Move() 사용시 Health가 -5 더 소모

Knight 클래스는 Attack() 추가하고 실행시 공격합니다를 출력해주세요

Healer 클래스는 Mana속성을 추가해주세요 (생성시 100)

Healer 클래스는 Heal(character) 메서드를 추가하고 해당 메서드는 character를 매개변수로 받습니다.

Healer 클래스는 Heal(character) 메소드 실행 시 Mana가 -10되고 전달받은 character 객체의 Rest() 메소드를 두 번 실행합니다.

Healer 클래스는 현재 마나속성을 확인할 수 있는 checkMana() 메서드를 추가해주세요

파이썬 복습 Note

아래의 코드를 수정하여 data3.p 파일이 없을 경우 사용자에게 name, address, email을 입력 받아 data3.p 파일을 생성하게 코드를 수정하세요

```
import pickle
with open("data3.p","rb") as file:
    name = pickle.load(file)
    address = pickle.load(file)
    email = pickle.load(file)

print(name, address, email)
```

파이썬 복습 Note

매개변수로 전달받은 숫자의 구구단을 출력해주는 함수를 만드는데 매개변수로 1~9 이외의 숫자를 전달하면
NotNumberException을 발생시켜 주세요

실행 :
gugudan (2)

2 X 1 = 2

2 X 2 = 4

...

2 X 9 = 18

gugudan(10)

NotNumber Traceback (most recent call last)

...

NotNumber: 잘못된 숫자입니다.

파이썬 복습 Note

아래와 같이 동작하도록 pklist 패키지를 만드세요

```
import pklist
```

```
pklist.list_max([1,2,3],[5,6,7],[4,3,2,5,6])
```

7

```
pklist.list_min([1,2,3],[5,6,7],[4,3,2,5,6])
```

1

```
pklist.list_avg([1,2,3],[5,6,7],[4,3,2,5,6])
```

4.0