



문화 A0019

파이썬프로그래밍

김 태 완

kimtwan21@dongduk.ac.kr

오늘 배울 내용

- 시험 풀이
- 과제 풀이
- 문자열

파이썬프로그래밍

중간 고사

1번 문제

(1점)

- 세 개의 문자형 변수를 출력하여 아래와 같이 결과가 나오도록 완성해보자.

< 코드 >

```
x = '010'  
y = '1234'  
z = '5678'  
  
print()
```

< 실행결과 >

010-1234-5678

2번 문제

(2점: 각 1점)

- 학번을 입력 받아 짝수이면 "Group1 " 를 출력, 홀수이면 "Group2 " 를 출력해보자.
 - 학번은 "id"라는 변수에 입력 받습니다.

```
id = int(input("학번 입력:"))  
  
if     print("Group1")  
else: 
```

3번 문제

(2점)

- 리스트 name에 있는 원소의 이름과 길이를 함께 출력하는 코드를 완성해보자.

< 코드 >

```
name = ['hello', 'trip', 'pho', 'airplane', 'zebra']  
  
for i in name:  
    print()
```

< 실행결과 >

```
hello의 길이는 5  
trip의 길이는 4  
pho의 길이는 3  
airplane의 길이는 8  
zebra의 길이는 5
```

4번 문제

(5점)

- 리스트 a에 있는 원소 중 30보다 작고 10보다 큰 원소의 개수를 출력해보자.

```
a = [1, 11, 40, 24, 67, 22, 19, 24, 88]  
res=0
```



```
print("원소의 개수는 %d 입니다"%res)
```

5번 문제

(5점)

- While문을 이용하여 다음과 같이 30이하의 자연수 중 3의 배수를 제외하고 출력해보자.
 - 단 출력 결과와 결과 사이는 쉼표(,)로 연결한다. (줄 바꿈 없이 계속 옆으로 출력)

< 코드 >

```
i = 1
while 1점:
    if i%3==0:
        2점
    else:
        2점
    i+=1
```

< 실행결과 >

1, 2, 4, 5, 7, 8, 10, 11, 13, 14
, 16, 17, 19, 20, 22, 23, 25, 2
6, 28, 29,

6번 문제

(5점)

- for문을 이용하여 오른쪽의 실행결과가 나오도록 코드를 구현해보자.

< 실행결과 >

```
1
22
333
4444
55555
666666
```

7번 문제

(5점)

- 딕셔너리 자료 구조를 이해하여 아래와 같이 출력해보자.
 - 국가명의 길이가 3인 국가의 수도만 아래와 같이 출력되도록 코드를 구현하세요.

< 코드 >

```
capital = {"대한민국": "서울", "미국": "워싱턴", "프랑스": "파리", "영국": "런던", "스위스": "베른", "베트남": "하노이", "덴마크": "코펜하겐"}
```

```
### coding here ###
```

< 실행결과 >

```
프랑스의 수도는 파리 입니다.  
스위스의 수도는 베른 입니다.  
베트남의 수도는 하노이 입니다.  
덴마크의 수도는 코펜하겐 입니다.
```

8번 문제

(5점)

- 오늘 먹은 음식 리스트 : eat
- 먹고 싶은 음식 리스트 : wish
- 먹고 싶은 리스트 중에서 오늘 먹은 음식을 찾아 아래와 같이 출력해보자.

< 코드 >

```
eat = ['미역국', '김치', '햄버거', '피자', '떡', '순대', '떡볶이']  
wish = ['햄버거', '떡볶이']  
  
## coding here ##  
  
print(result)
```

< 실행결과 >

```
['햄버거', '떡볶이']
```

파이썬프로그래밍

중간 과제

1번 문제

- 사용자로부터 파일이 위치한 드라이브 이름(D:\), 디렉토리 이름 (python\), 파일이름(homework), 확장자(.py)를 입력 받아서 완전한 파일이름(D:\python\homework.py)으로 만드는 프로그램을 작성해보자.

드라이브 이름: D\
디렉토리 이름: python\
파일 이름: homework
확장자: .py

>>>완전한 파일이름: D:\python\homework.py

2번 문제

- 정답은 25이며, 숫자를 입력 받아 UP/DOWN을 출력하여 정답이 나올 때 까지 반복하는 프로그램
 - 정답은 25로 고정
 - 정답을 맞추기 전에는 프로그램이 종료 x
 - 정답을 입력하면 “정답!” 출력 후 프로그램 종료

```
>>> 숫자를 입력하세요 : 90  
>>> DOWN !
```

```
>>> 숫자를 입력하세요 : 10  
>>> UP !
```

```
>>> 숫자를 입력하세요 : 30  
>>> DOWN !
```

```
>>> 숫자를 입력하세요 : 20  
>>> UP !
```

```
>>> 숫자를 입력하세요 : 24  
>>> UP !
```

```
>>> 숫자를 입력하세요 : 25  
>>> 정답 !
```

3번 문제

- 자동 판매기를 시뮬레이션하는 프로그램을 작성해보자.
- 사용자는 1000원짜리 지폐만 사용할 수 있으며, 거스름돈을 계산해서 동전으로 반환한다.

물건 값을 입력 : 1650

받은 금액 : 2000

>>>거스름돈은 아래와 같습니다.

>>>500원 = 0개, 100원 = 3개, 10원 = 5개, 1원 = 0개

4번 문제

- while문을 이용하여 아래와 같이 출력해보자.

* * * * *

* * * * *

* * *

*

5번 문제

- 정답 리스트 내 있는 데이터인지 아닌지 판별하는 프로그램
- answer 리스트에 있으면 'O', 없으면 'X'를 출력
- 입력한 리스트가 A,B,C 가 아닌 경우 '리스트에 없습니다.' 출력
- `answer = ['apple', 39, 'music', 568.2, 'Dongduk', 145, 'hello']`
- `A = ['hello', 62, 'umbrella', 145]`
- `B = ['September', 512.3, 'coffee', 39, 'keyboard', 'notebook', 0.5, 'f12']`
- `C = ['computer', 568.2, 39, 'aPple', 111, 'Dongduk', 'water']`

```
>>> 리스트를 입력하세요 : A  
>>> OXXO
```

```
>>> 리스트를 입력하세요 : B  
>>> XXXOXXXX
```

```
>>> 리스트를 입력하세요 : C  
>>> XOOXXOX
```

```
>>> 리스트를 입력하세요 : F  
>>> 리스트에 없습니다.
```

문자열

문자열

- 문자열 : 큰 따옴표, 작은 따옴표로 표현
 - “파이썬프로그래밍”, ‘데이터사이언스전공’
- 리스트와 문자열 비교
 - 리스트는 대괄호 []로 묶고 문자열은 작은(큰) 따옴표로 묶어 출력
 - A = [1, 2, 3, 4, 5]
 - print(A[0]) : 1
 - print(A[1:3]) : [2, 3]
 - print(A[2:]) : [3, 4, 5]
 - B = “파이썬프로그래밍”
 - print(B[0]) : 파
 - print(B[1:3]) : 이썬
 - print(B[2:]) : 썬프로그래밍

문자열

- 문자열 기본
 - 덧셈과 곱셈 연산자
 - A = “파이썬” + ‘프로그래밍’
 - print(A) : 파이썬프로그래밍
 - B = ‘안녕’
 - print(B *3) : 안녕안녕안녕
 - print('*' * 10) : **********

문자열

- len() 함수
 - 리스트나 문자열의 개수를 셀 때 사용
 - 띄어쓰기도 길이 포함
 - 예시
 - word = 'hello everyone'
 - `print(len(word)) = 14`
 - word2 = '김태완'
 - `print(len(word2)) = 3`
 - `for i in range(0, len(word2)):`
`print(word2[i], end="")`

문자열

- 영문자열 대소문자 변환
 - 문자열을 대문자 로 바꿀 수 있는 upper()
 - 문자열을 소문자 로 바꿀 수 있는 lower()
 - 문자열을 대소문자를 서로 바꿀 수 있는 swapcase()
 - 문자열을 단어 앞만 대문자로 바꿀 수 있는 title()
- 영문자열 대소문자 확인
 - islower(): 소문자로 구성되어 있는지 확인
 - isupper(): 대문자로 구성되어 있는지 확인
 - isdigit(): 숫자로 인식될 수 있는지 확인
 - isspace(): 공백으로만 구성되어 있는지 확인

```
word = 'hEllo eveRyone'  
  
print(word.upper())  
print(word.lower())  
print(word.swapcase())  
print(word.title())
```

```
word = 'hello'  
word2 = '12345'  
word3 = " "  
  
print(word.islower())  
print(word2.isdigit())  
print(word3.isspace())
```

문자열

- 문자열 찾기 함수

- count()

```
word = '공부가 너무 재미있어서 공부도 공부만 하고 싶어요'  
  
print(word.count('공부'))
```

- find()

```
word = '나는 공부가 너무 재미있어서 공부도 공부만 하고 싶어요'  
  
print(word.find('너무'))
```

- find('문자열', 시작 index) : 문자를 찾을 때 어디서 부터 찾을지 시작 index. 생략 시 0
 - find('문자열', 시작 index, 끝 index) : 문자를 찾을 때 어디 까지 찾을지 index, 생략 시 마지막 index
 - 찾는 문자가 존재 하지 않는다면 -1 을 반환

문자열

- 문자열 찾기 함수
 - rfind() : 오른쪽 부터 찾기

```
word = '나는 공부가 너무 재미있어서 공부도 공부만 하고 싶어요'  
print(word.rfind('공부'))
```

- index()
 - find 함수와 동일, rindex 등 동일하게 사용 가능
 - 찾는 문자열이 없을 경우 find 함수는 -1, index 함수는 ValueError 출력
 - find 함수는 문자열만 가능하나 index함수는 리스트, 튜플에서 사용 가능 (딕셔너리는 둘 다 사용 x)

```
list = ['나는', '파이썬', '공부가', '좋아요']  
print(list.index('공부가'))
```


문자열

- 문자열 공백 삭제, 변경하기
 - `strip()` : 인자로 전달된 문자를 String의 왼쪽과 오른쪽에서 제거
 - `lstrip()` : 인자로 전달된 문자를 String의 왼쪽에서 제거
 - `rstrip()` : 인자로 전달된 문자를 String의 오른쪽에서 제거

```
word = ' 나는 공부가 너무 재미있어서 공부도 공부만 하고 싶어요 '  
print(word.strip())
```

- `replace()` : 문자열 변경

```
word = '나는 공부가 너무 재미있어서 공부도 공부만 하고 싶어요'  
print(word.replace('공부', '운동'))
```

문자열

- 문자열 분리, 결합하기
 - split()

```
word = '나는 공부가 너무 재미있어서 공부도 공부만 하고 싶어요'  
  
print(word.split())
```

```
word = '하나:둘:셋'  
  
print(word.split(':'))
```

- join()

```
data = ['A', 'B', 'C', 'D', 'E']  
data_join = '-'.join(data)  
print(data_join)
```

감사합니다

kimtwan21@dongduk.ac.kr

김 태 완