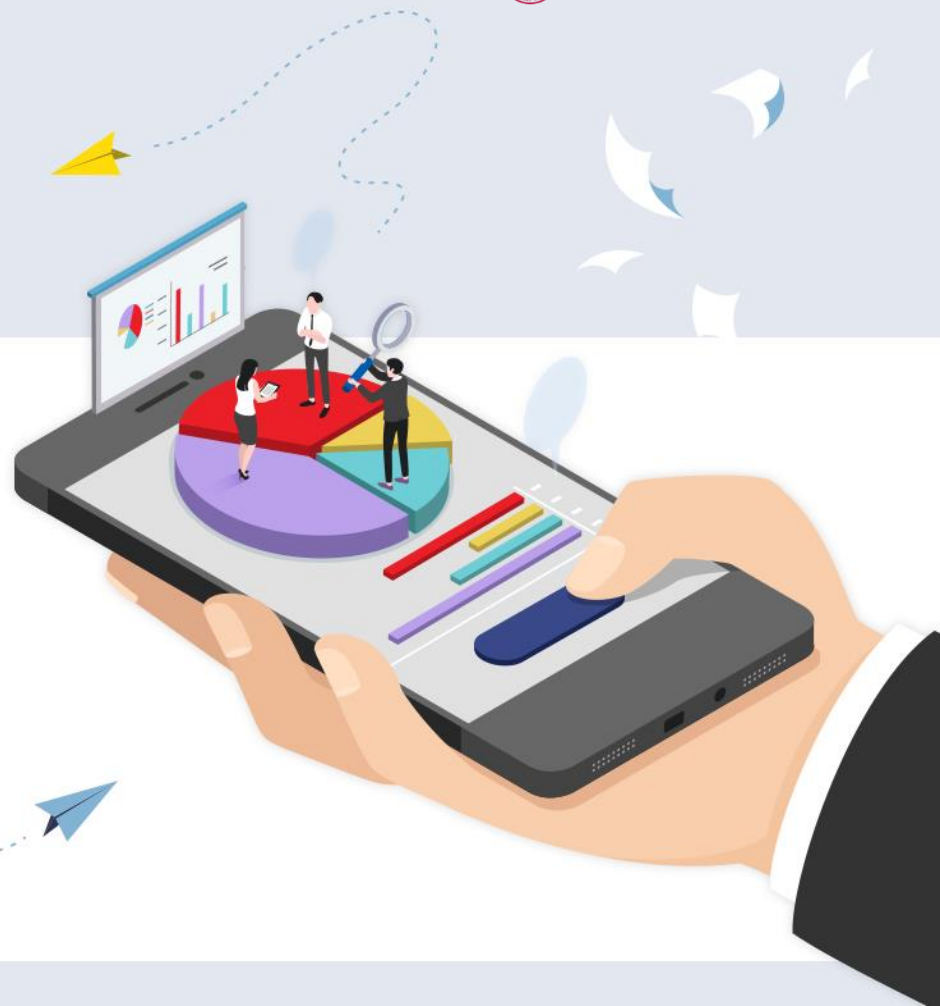




01 for문과 선택문 활용



Q

수강하는 과목의 점수를 리스트에 저장하고
점수가 70점 이상인 과목의 수를 출력하는 프로그램

```
score=[70,90,60,100,50]
cnt=0
for i in range(5):
    if score[i]>=70:
        cnt += 1
print('count:',cnt)
```

실행결과
>>>
count : 3

Q 사용자로부터 시작, 끝 값을 입력 받은 후
전체 합계, 짝수의 합계, 홀수의 합계를 출력하는 프로그램

```
s=int(input('start num: '))
e=int(input('end num: '))
total, even, odd=0,0,0
for i in range(s,e+1):
    total += i
    if i % 2 == 0:
        even += i
    else:
        odd += i
print('전체합계: ',total)
print('짝수의 합: ',even)
print('홀수의 합: ',odd)
```

실행결과

>>>

```
start num: 1
end num: 10
전체합계: 55
짝수의 합: 30
홀수의 합: 25
```



레벨판별프로그램

학생들의 코딩 점수가 리스트에 저장되어 있다.

- 90점 이상이면 'LEVEL 1'
- 80점 이상이면 'LEVEL 2'
- 70점 이상이면 'LEVEL 3'
- 나머지는 'FAIL' 메시지를 출력

```
score= [85,95,70,50,100]
n=0
print('코딩 시험 결과')
print('-'*30)
for s in score:
    n += 1

    if s>=90:
        result='LEVEL1'
    elif s>=80:
        result='LEVEL2'
    elif s>=70:
        result='LEVEL3'
    else:
        result='FAIL'

print (' {}번 학생의 점수는 {}입니다.'.format(n,s))
print ('결과는 {}입니다.'.format(result))
print ()
```

실행결과

>>>

코딩 시험 결과

1번 학생의 점수는 85입니다.
결과는 LEVEL2입니다.

2번 학생의 점수는 95입니다.
결과는 LEVEL1입니다.

3번 학생의 점수는 70입니다.
결과는 LEVEL3입니다.

4번 학생의 점수는 50입니다.
결과는 FAIL입니다.

5번 학생의 점수는 100입니다.
결과는 LEVEL1입니다.

NOTE

문자열 뒤에
점 입력

```
print ('{}번 학생의 점수는 {}입니다.'.format(n,s))
```

자료 지정

괄호 () 안에
{ }에 대응하는
자료 지정

```
print ('결과는 {}입니다.'.format(result))  
print ()
```

결과는 LEVEL1입니다.

NOTE

`print ('{}번 학생의 점수는 {}입니다.'.format(n,s))`

- 정의되는 자료는 인덱스 번호로 문자열 안의 {}에 적용
- 인덱스 번호는 0부터 시작
- 인덱스 번호가 생략되면 `format()`의 값이 차례대로 {}에 대입

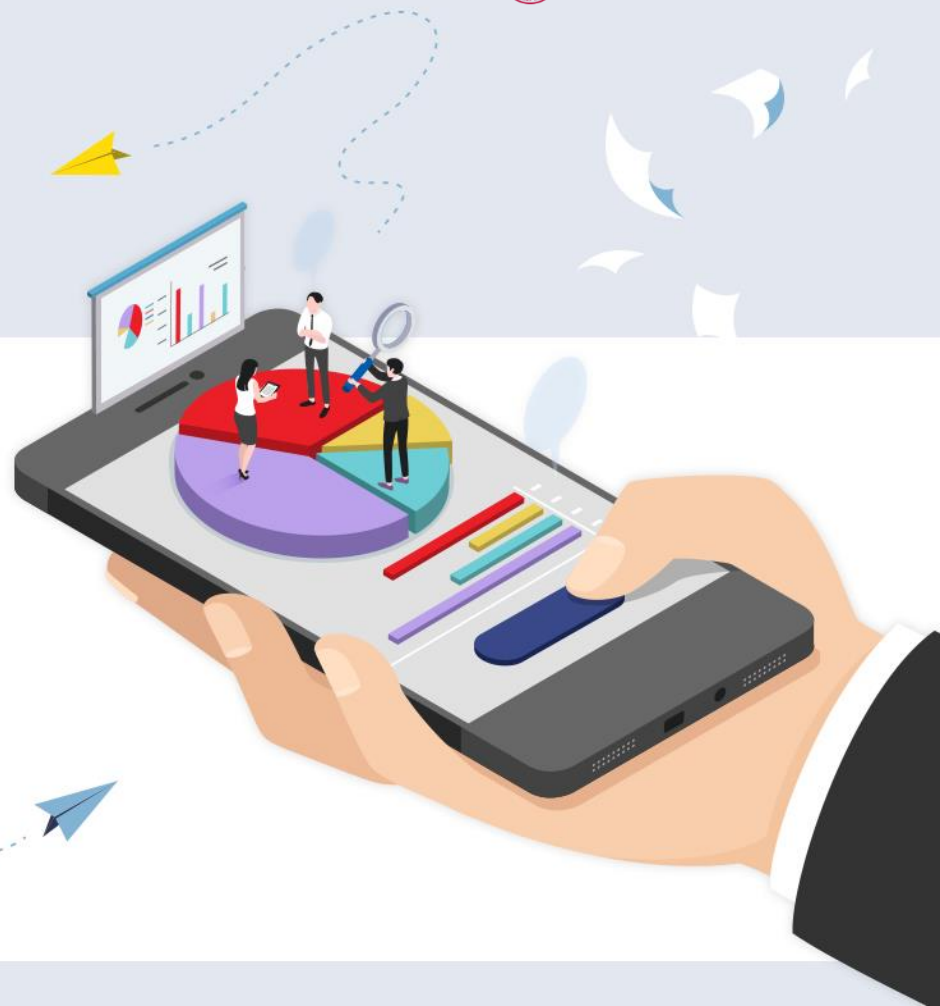
```
print ('결과는 {}입니다.'.format(result))  
print ()
```

결과는 LEVEL1입니다.



02

while문과 선택문 활용



Q

주사위 게임

컴퓨터가 생성한 주사위 숫자와
사용자가 입력한 숫자를 비교하여 높은 숫자가 나오면
이기는 주사위게임 프로그램

- '0'을 입력하면 게임은 종료되고
아무거나 입력하면 프로그램은 계속됨
- 랜덤숫자를 생성하는 방법은
random모듈을 사용함

```
import random
```

```
print('주사위 게임 시작')
```

```
throw=input('Enter를 치세요')
```

```
while throw != '0':
```

```
    com=random.randint(1,6)
```

```
    user=int(input('num: '))
```

```
    if com>user:
```

```
        win='com'
```

```
    else:
```

```
        win='user'
```

```
    print ('com {} : user {}, {} win'.format(com,user,win))
```

```
    throw=input ('재시작:Enter,종료:0 ')
```

실행결과

>>>

주사위 게임 시작

Enter를 치세요

num: 5

com 3 : user 5, user win

재시작:Enter,종료:0

num: 4

com 1 : user 4, user win

재시작:Enter,종료:0 0

Q

막대그래프 그리기

사용자로부터 숫자를 입력 받아 숫자만큼 그래프 그리는 프로그램

- 단, 입력되는 숫자가 0이 들어오면 막대그래프 그리는 것을 멈춤

```
while True:
    n=int(input('n:'))
    if n==0:
        break
    for i in range(n):
        print('*',end=' ')
    print()
```

실행결과

>>>

n:5

n:4

n:1

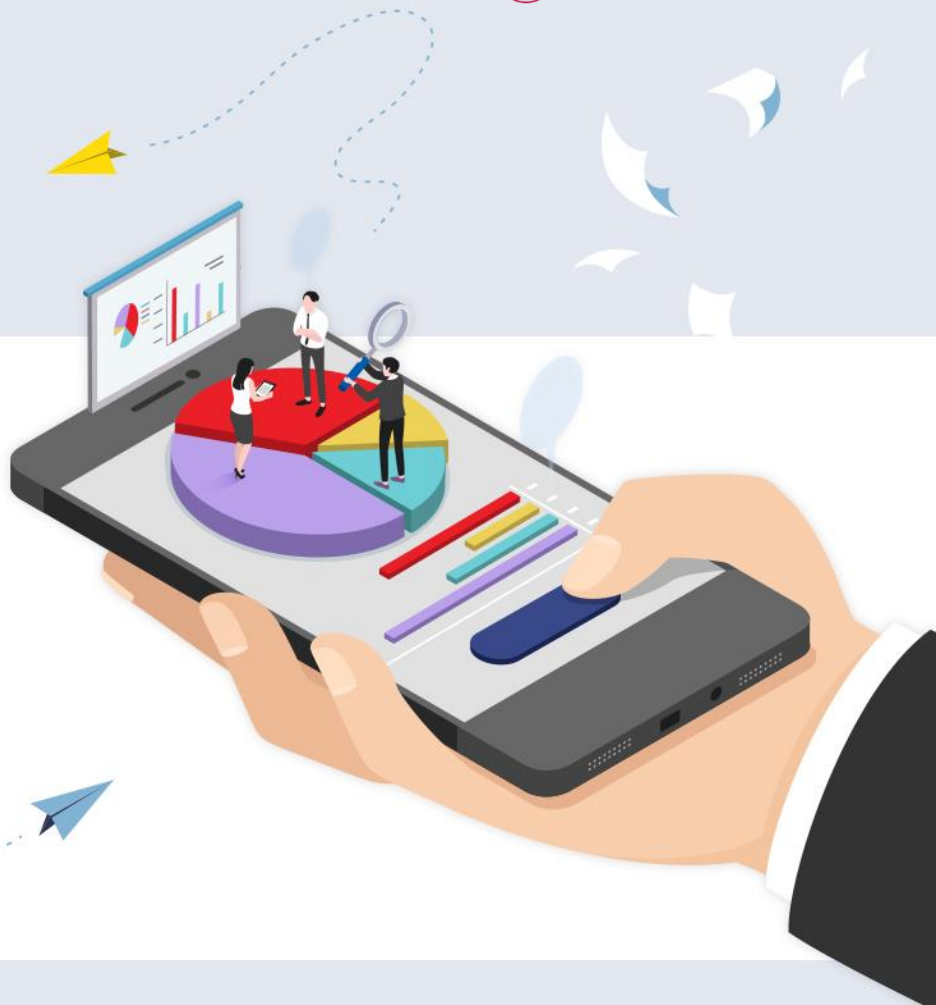
*

n:0



03

효율적인 자료구조와 반복문 활용



Q

합격/불합격 판단하기

리스트에 저장된 점수를 차례대로 읽어 와서
리스트 저장된 번호 순으로 '합격' 또는 '불합격'을 출력하고
합격한 학생 수를 출력하는 프로그램

- 합격의 기준은 점수가 80점 이상

Q

합격/불합격 판단하기

```
score= [60,80,90,70,95]
cnt=0
print(score)
for i in range(len(score)):
    if score[i]>80:
        result='합격'
        cnt += 1
    else:
        result='불합격'
    print('{}번 학생은 {}입니다.'.format(i+1,result))

print('합격한 학생 수: ', cnt)
```

실행결과

>>>

[60, 80, 90, 70, 95]

1번 학생은 불합격입니다.

2번 학생은 불합격입니다.

3번 학생은 합격입니다.

4번 학생은 불합격입니다.

5번 학생은 합격입니다.

합격한 학생 수: 2

Q

메뉴와 가격정보 출력하기

딕셔너리에 학생 식당 메뉴(키)와 가격(값)을 저장하고
전체 정보를 출력한 후 메뉴에 대한 가격을 검색해서
출력해주는 프로그램

- 사용자가 'q'를 입력하면 종료

Q

메뉴와 가격정보 출력하기

```
menu={}  
  
menu['라면']=3000  
menu['떡볶이']=4000  
menu['김밥']=2000  
menu['햄버거']=5000  
  
for i in menu.keys():  
    print('{}-{}'.format(i,menu[i]))  
  
while True:  
    s=input('seach menu : ')  
    if s=='q':break  
    print(menu.get(s, 'not Found'))
```

실행결과

>>>

라면 - 3000

떡볶이 - 4000

김밥 - 2000

햄버거 - 5000

seach menu : 김밥
2000seach menu : 햄버거
5000

seach menu : q