



문화 A0019

파이썬프로그래밍

김 태 완

kimtwan21@dongduk.ac.kr

1. 함수 만들기

- 자연수를 입력 받아 약수를 출력하는 함수를 만들어 보자.
- 함수 이름 : divisor
 - 약수를 리스트 형태로 출력함
- 예시

```
divisor(10)  
>> [1, 2, 5, 10]
```

```
divisor(7)  
>> [1, 7]
```

2. 함수 만들기

- 나라 이름을 함수에 입력하면 수도를 출력해주는 함수
- 함수 이름 : find_capitalcity(capital, country)
 - 주어진 딕셔너리에 매칭되는 나라 - 수도 출력
- 예시

```
def find_capitalcity(capital, country):  
    ##### coding here #####  
    pass
```

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

```
print(find_capitalcity(capital, "대한민국"))  
print(find_capitalcity(capital, "덴마크"))  
print(find_capitalcity(capital, "일본"))
```

3. 파일 쓰기

- 파이썬프로그래밍 수업을 듣는 50명의 학생 중간고사/기말고사/과제/출석 점수 기록
 - 총 4개 파일 : mid_test.txt, final_test.txt, homework.txt, attendance.txt
 - 중간/기말고사 점수는 0~100점 만점, 과제는 0~20점 만점, 출석은 0~10점 만점
 - 각 텍스트 파일에는 한 줄에 한 명의 점수가 기록되도록 함
 - txt파일을 제출할 필요는 없음

mid_test.txt

```
97
100
56
36
...
88
91
```

final_test.txt

```
47
50
32
65
...
81
39
```

homework.txt

```
12
0
19
20
...
13
17
```

attendance.txt

```
5
5
9
10
...
8
10
```

4. 파일 읽기

- 파이썬프로그래밍 수업을 듣는 50명의 학생 중간고사/기말고사/과제/출석 점수 환산
 - 총 4개 파일 : mid_test.txt, final_test.txt, homework.txt, attendance.txt을 읽기
 - 중간 / 기말고사는 각 35%, 과제는 20%, 출석은 10% 반영 : 총 100%
 - 50명의 최종 점수를 list 또는 numpy array에 대입
 - list 명 : score
 - 참고 : len(list) = 50
- 이 list를 입력 받아 1등 점수를 반환하는 함수 “center”를 만들어 보자.

```
def center(list):  
    ## coding here ##  
  
print(center(score))
```

5. 클래스 만들기

- Student 클래스 생성해보자.
- 클래스 내 필수 사항
 - 필드 : 이름, 전공, 학년, 전화번호
 - 메서드 : 자기소개
- 출력 결과
 - “안녕하세요. 저의 이름은 정영희, 데사 전공 1학년이며, 연락처는 010-1234-5678입니다.”

```
class Student_DD:
    ##coding here
    pass

student1 = Student_DD("정영희", "데사", "1", "010-1234-5678")
student1.introduce()
```

감사합니다

kimtwan21@dongduk.ac.kr

김 태 완