

# data structure algorithm

made in fastcampus lecture by bools

### • 나머지 연산자

• 나눗셈 결과 나머지만 구함

```
result = num1 % num2
print('num1: {}, num2: {}, result: {}'.format(num1, num2, result))
num1: 10, num2: 3, result: 1
```

### • 몫 연산자

### • 나눗셈 결과 몫만 구함

```
result = num1 // num2
print('num1: {}, num2: {}, result: {}'.format(num1, num2, result))
num1: 10, num2: 3, result: 3
```

## • 나머지와 몫을 한번에 구하기

• divmod() 함수

```
result = divmod(num1, num2)
print('result: {}'.format(result))
print('몫: {}'.format(result[0]))
print('나머지: {}'.format(result[1]))
```



result: (3, 1)

몫: 3

나머지: 1

### • 실습

- 학급 전체 학생 수 입력
- 한 모둠에 속하는 학생 수 입력
- 전체 모둠 수와 남는 학생 수 출력

```
allStuCnt = int(input('전체 학생 수 : '))
stuCntOfGroup = int(input('한 모둠 학생 수 : '))
groupCnt = allStuCnt // stuCntOfGroup
overStuCnt = allStuCnt % stuCntOfGroup

print('전체 학생 수 : {}'.format(allStuCnt))
print('한 모둠 학생 수 : {}'.format(stuCntOfGroup))
print('모둠 수 : {}'.format(groupCnt))
print('남는 학생 수 : {}'.format(overStuCnt))
```



전체 학생 수 : 25 한 모둠 학생 수 : 4 전체 학생 수 : 25

선세 약생 수 : 25

한 모둠 학생 수 : 4

모둠 수 : 6

남는 학생 수 : 1

divmod() 함수도 이용해 보자.

# • 실습

### 123개의 사과를 4개씩 직원들한테 나누어 주려고 한다. 최대 나누어 줄수 있는 직원수와 남는 사과 개수를 출력해 보자.

```
employee = 123
apple = 4
result = divmod(employee, apple)
print('사과를 나주어 줄 수 있는 최대 직원 수 : {}명'.format(result[0]))
print('남는 사과 개수 : {}개'.format(result[1]))
```



사과를 나주어 줄 수 있는 최대 직원 수 : 30명

남는 사과 개수 : 3개