



Round 3

**PRESS
START**



《 Round 3 》

- 모듈
- 예외처리
- 파이썬 자료구조



New
Assignment



《 Round 3 》

- 모듈 《
- 예외처리
- 파이썬 자료구조



Let's
Go



calculator.py

```
def add() :  
def sub() :  
def mul() :  
def div() :  
def rem() :  
  
def result_write() :  
def result_read() :  
  
def start() :  
    .  
    .
```

cal_module.py

```
def add :
```

```
def sub :
```

```
def div :
```

```
def mul:
```

cal_IO.py

```
def result_write() :
```

```
def result_read() :
```

calculator.py

```
import cal_module  
import cal_IO  
import cal_start
```

```
start()  
...
```

cal_start.py

```
def start() :
```



why ?

why ?

Looks good !



《 Round 3 》

- 모듈 - complete
- 예외처리 《
- 파이썬 자료구조



Let's
Go



example Java...

```
static Scanner sc = new Scanner(...);
int num1 ;
int num2 ;

try {
    num1 = sc.nextInt();
    num2 = sc.nextInt();
    System.out.println(num1/num2);
} catch (ArithmeticException e) {
    System.out.println("0으로 나누지 마!")
} catch (InputMismatchException e) {
    System.out.println("int형으로 넣어라")
} finally {
    System.out.println("try문이 끝났어.")
}
```



```
try :
    num1 = int(input())
    num2 = int(input())
    print(num1/num2)

except ZeroDivisionError as e :
    print("0으로 나누지 마!")

except ValueError as e:
    print("int형으로 넣어라")

finally :
    print("try문이 끝났어.")
```

《 Round 3 》

- 모듈 - complete
- 예외처리 - complete
- 파이썬 자료구조 《



Let's
Go



```
# 리스트 자료형
ex_list = [1, "이", , [3]]

# 튜플 자료형
ex_tuple = (1, "이", [3])

# 딕셔너리 자료형
ex_dictionary = {1 : 1, 2 : "이", 3 : [3]}

# 집합 자료형
ex_set = set([1, "이", [3]])
```

List

```
# 리스트 자료형  
ex_list = [1, "0", , [3]]
```

요소 수정	요소 형태	특징
0	Anything	대괄호

- array(배열)과 list(리스트)의 차이

배열은 크기가 정해져 있으며, 기능이 존재하지 않는다.
리스트는 크기가 가변적이며 여러 기능이 존재한다.

배열 인덱스는 값에 대한 유일무이한 식별자.
리스트 인덱스는 몇 번째 데이터인가 정도의 의미.

배열은 요소를 삭제해도 순서가 그대로이다.
리스트는 중간 요소를 삭제하면 순서가 바뀐다.

그렇다고 한다...

- ```
딕셔너리 자료형
ex_dictionary = {1 : 1, 2 : "0", 3 : [3]}
```

```
집합 자료형
ex_set = set([1, "0", [3]])
```



# NEXT STAGE

