



# Round 1

**PRESS  
START**



## 《 Round 1 》

- Python의 특징 파악
- 간단한 계산기 만들기



New  
Assignment



## 《 Round 1 》

- Python의 특징 파악 《《
- 간단한 계산기 만들기



Let's  
Go



?

```
23 def start() :
24     print("-Calculator v1.0-")
25     while 1 :
26         global result
27         if result == 0 :
28             result = int(input("주연산자(수)를 입력하세요. "))
29         else :
30             print(result)
31         unResult = result
32
33         sign = input("부호를 입력하세요. ! 를 입력하면 종료합니다.")
34         if sign == '!' :
35             break
36         try :
37             num = int(input("피연산자(수)를 입력하세요. "))
38         except :
39             print("입력이 잘못되었습니다.")
40
41         if sign == '+' :
42             add(num)
43             print(unResult, " ", sign, " ", num, " = ", result)
44
45         elif sign == '-' :
46             sub(num)
47             print(unResult, " ", sign, " ", num, " = ", result)
48
```

# Tab

10 팩토리얼을  
for문을 이용해  
짚다면...??



# in java && C...

```
# if you are programming 10! for C/java  
for(int i = 1; i <= 10; i++)  
{  
    result *= i  
}
```

10 팩토리얼을  
for문을 이용해  
짤다면...??



```
# if you are programming 10! for C/java
for(int i = 1; i <= 10; i++)
{
    result *= i
}
```

```
# in python...
for i in range(10):
    result *= i
```

파이썬의 기본적인 for문의 형태는  
타 언어의 foreach문과 흡사해!



파이썬의 전형적인 for문은  
**리스트(배열)**를 읽어오는 방식!!!

- in : 리스트를 읽어오는 연산자
- range() : 리스트를 만들어주는 함수
- List Comprehension 을 이용해서  
다양하고 가독성 높은 표현가능 !!!

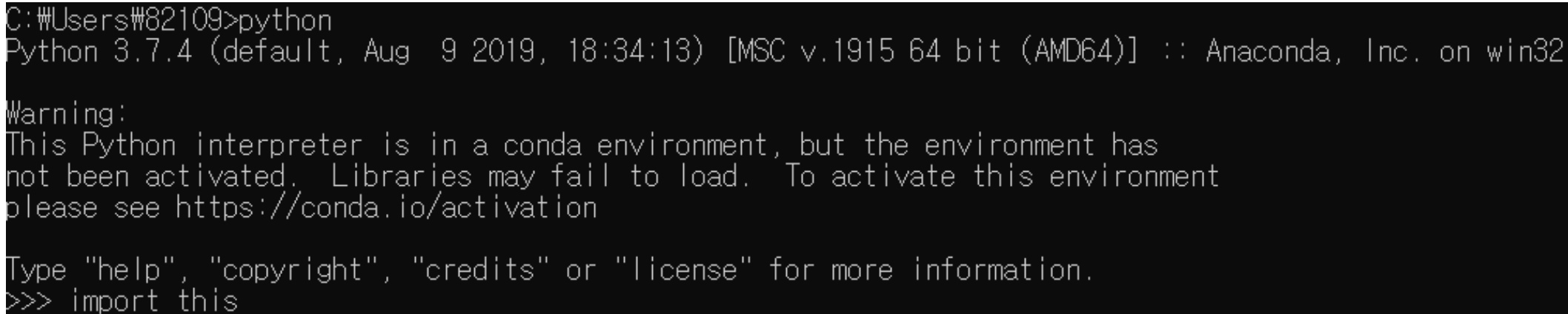
1	2	3	4		1	5	9
5	6	7	8	→	2	6	10
9	10	11	12		3	7	11
					4	8	12

```
# list Comprehension
matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]
[[row[i] for row in matrix] for i in range(4)]
# -> [[1, 5, 9], [2, 6, 10], [3, 7, 11], [4, 8, 12]]
```

2차원 행렬 전치가 무려 1줄 컷...세상에...







**cmd -> python -> import this**  
**직접 실행해보자!**

**파이썬은 하나의 간결하고 아름다운 코드를 지향해!  
파이썬 인터프리터엔 파이썬의 철학을 담은  
이스터에그(import this)가 존재할 정도!**



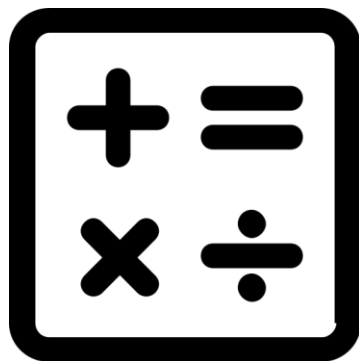
## 《 Round 1 》

- Python의 특징 파악 - Complete
- 간단한 계산기 만들기 《



Let's  
Go





계산기

표준

2977 + 654 =

3,631

MC MR M+ M- MS M\*

%	CE	C	↵
1/x	x <sup>2</sup>	<sup>3</sup> √x	÷
7	8	9	×
4	5	6	-
1	2	3	+
+/-	0	.	=

계산기

표준

2977 + 654 =

3,631

MC MR M+ M- MS M\*

2977 + 654 =

3,631

2983 - 6 =

2,977

2883 + 393 - 293 =

2,983

🗑️

input()

-> int(num1)  
-> String(sign)  
-> int(num2)

Function(add)

Function(mul)

Function(sub)

Function(div)

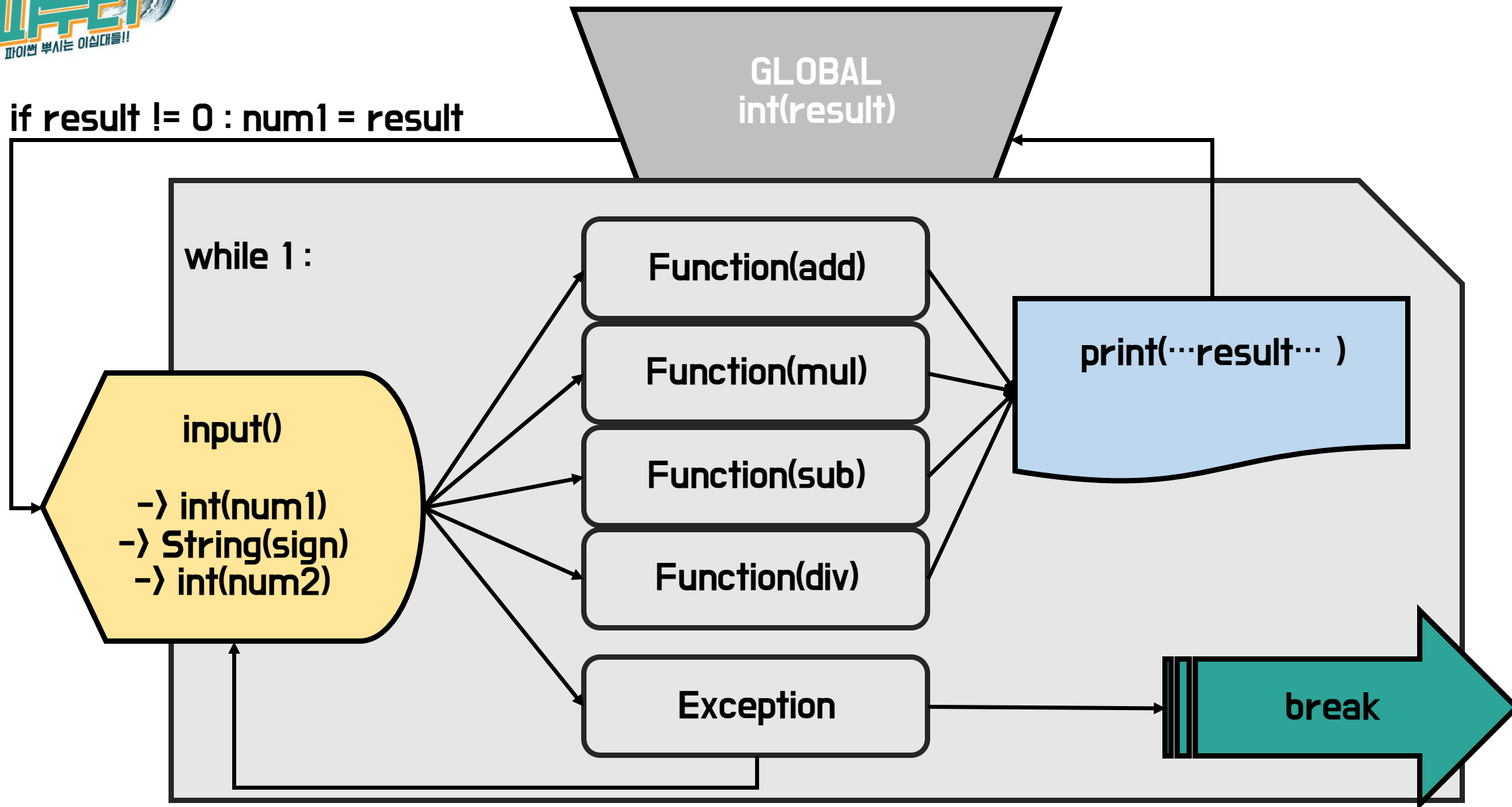


break

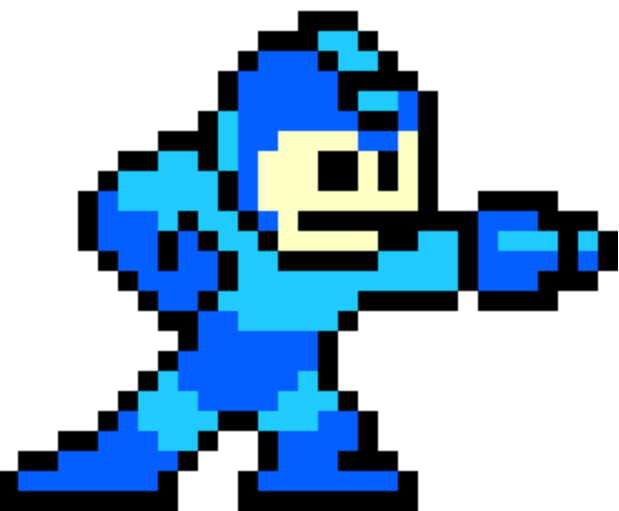
print(...result... )

GLOBAL  
int(result)

if result != 0 : num1 = result



# WARNING



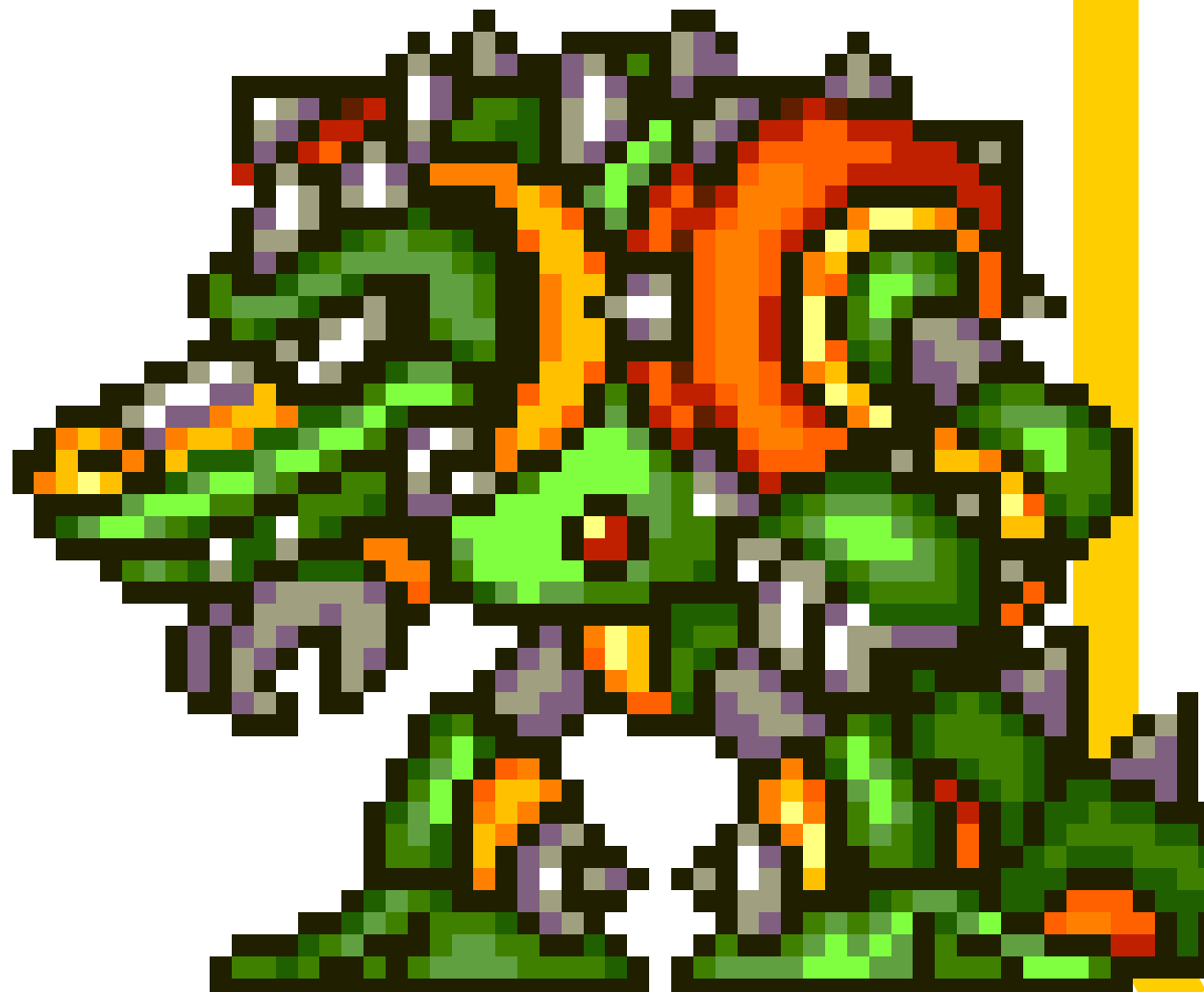


# WARNING

## 《 QUEST 》

- 다음 조건에 맞춰 계산기 완성 시키기 -

함수(def) 사용  
제곱(^) 구현  
자신만의 독특한 기능 구현



# NEXT STAGE

