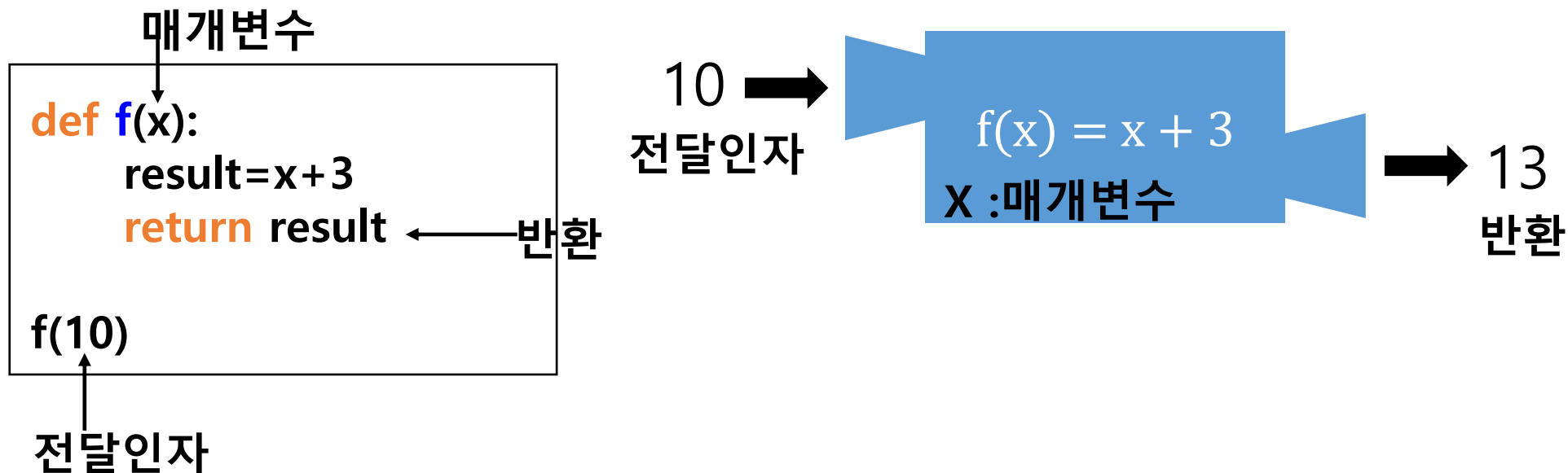


함수

- 매개변수 & 전달인자 & 반환
 - 함수 호출 파트(외부) 에서 함수 정의 파트(내부)로 값을 전달할 때 **전달인자**를 사용한다.
 - 함수 정의 파트에서는 값을 받을 때 **매개변수**로 받는다.
 - 함수 정의 파트에서 함수 호출 파트로 값을 내 보낼 때에는 **반환(return)**을 사용한다.



- Quiz

```
def f_a(x, y, z):  
    result=x+y+z  
    return result
```

```
result = f_a(10, 20, 30)  
print (result)
```

```
def f_b( ):  
    for count in range(100):  
        result+=count  
    print ('result:', result)
```

```
result = f_b( )  
print (result)
```

전달인자:

매개변수:

반환:

전달인자:

매개변수:

반환:

- None은 타입 자체가 None이다.

```
>>> None == False
False
>>> type(None)
NoneType
```

- Quiz

```
def f_a( ):
    result=10+20+30
    return result
```

```
result = f a( )
print ('result:', result)
```

```
def f_b( ):
    for count in range(100):
        result+=count
    return result
```

```
print ('result:', f_b( ) )
```

전달인자:

매개변수:

반환:

전달인자:

매개변수:

반환:

- 지역변수
 - 함수 정의 파트(내부)에서만 유효한 변수
 - 함수가 호출되었을 때 생성되며 함수를 벗어날 때 사라짐
- 전역변수
 - 프로그램 전체에서 유효한 변수
 - 프로그램이 시작될 때 생성되며, 프로그램이 끝날 때 사라짐

전역변수와 지역변수는 변수 이름이 같아도 각각 별개의 변수입니다.



```
def add ():  
    number = 5  
    number = number + 3  
    print (number)
```

지역변수
number

```
number = 5  
add()  
print (number)
```

전역변수
number

• 실행결과

```
8  
5  
>>>
```

- 지역변수, 전역변수의 사용
 - 전역변수를 함수 내부에서 수정 또는 선언하고자 할 때에는 **global** 키워드를 사용함

```
def add():  
    global number  
    number = number + 3  
    print(number)
```

전역변수
number

```
number = 5  
add()  
print(number)
```

전역변수
number

• 실행결과

```
8  
8  
>>>
```

- 지역변수, 전역변수의 사용



```
number = 5
def add():
    global number
    number = number + 3

add()
print(number)
```



```
number = 5
def add():
    number = number + 3

add()
print(number)
```


- 지역변수, 전역변수의 사용

```
number = 5  
def add():  
    print (number)  
  
add()
```

- Tip

전역변수의 값을 변경하는 것 없이, 단순히 전역변수의 값을 참조만 하는 경우에는 함수 내부에서 global로 선언하지 않아도 함수에 정의된 지역변수가 없다면 전역변수를 찾아 참조한다.

에러가 나올까요?
안나올까요?



- Local Variables and Global Variables with the Same Name

```
def spam(myName):  
    print ('Hello, ' + myName)  
    myName = 'Waffles'  
    print('Your new name is ' + myName)
```

```
myName = 'Albert'  
myName = spam(myName)  
print ('Howdy, ' + myName)
```

- Quiz

```
def spam(myName):  
    print ('Hello, ' + myName)  
    myName = 'Waffles'  
    print('Your new name is ' + myName)  
    return myName  
  
myName = 'Albert'  
myName = spam(myName)  
print ('Howdy, ' + myName)
```

```
def roots(value):  
    from math import sqrt  
    result = sqrt(value)  
    return result, -result
```

```
roots(2)
```

```
def a():  
    return 0
```

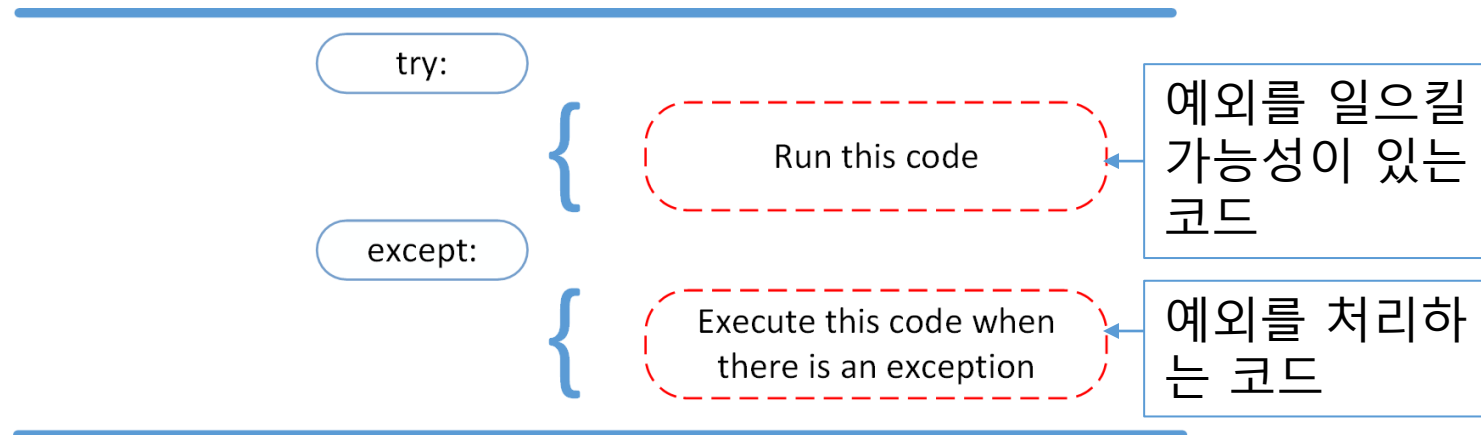
```
def b():  
    print (0)
```

```
a() + 1    #??  
b() + 1    #??
```

- 함수로 바꾸어보기
 - displayIntro() 함수 : 게임 시작에 나오는 인트로 메시지 부분
 - chooseCave() 함수 : 동굴번호 선택 코드 부분
 - checkCave() 함수 : 동굴번호와 랜덤번호 비교하는 코드 부분

예외처리

- **Syntax errors** : incorrect statement.
- **Exception error** : 실행중에 발생하는 오류



```
prompt = ('What is the airspeed velocity of an unladen swallow? ')
try:
    response = input(prompt)
except:
    response = ''
```

```
try:
    response = input(prompt)

except <ExceptionType1>:
    ...
except <ExceptionTypeN>:

except:
    print ('문제가 있습니다.')
else:
    print ('예외가 없습니다.')
finally:
```



```
try:
    number1, number2 = eval(input("두 숫자를 콤마로 분리하여 입력하세요"))
    result = number1/number2
    print("결과는", result, "입니다.")
except ZeroDivisionError:
    print("0으로 나누기")
except SyntaxError:
    print("입력에 콤마가 빠졌습니다.")
except:
    print("입력에 문제가 있습니다.")
else:
    print("예외는 없습니다.")
finally:
    print("finally절이 실행되었습니다.")
```

두 숫자를 콤마로 분리하여 입력하세요3,4
결과는 0.75 입니다.
예외는 없습니다.
finally절이 실행되었습니다.

- 다음과 같이 두수를 입력 받아 계산하여 출력하는 프로그램을 addition(x,y) 함수를 정의하여 작성하시오.

```
=====
첫번째 정수를 입력하세요:
```

```
3
```

```
두번째 정수를 입력하세요:
```

```
6
```

```
addition result 9
```

- subtraction(x,y) 함수를 추가하여 아래와 같이 출력되도록 작성하시오.

```
===== RESTART:
```

```
첫번째 정수를 입력하세요:
```

```
6
```

```
두번째 정수를 입력하세요:
```

```
3
```

```
subtraction result: 3
```

```
>>>
```

- multiplication(x,y)함수를 추가하여 아래와 같이 출력되는 프로그램을 작성하시오.

첫번째 정수를 입력하세요:

10

두번째 정수를 입력하세요:

20

multiplication result: 200

- division(x,y)함수 추가하여 아래와 같이 출력되는 프로그램을 작성하시오.

첫번째 정수를 입력하세요:

10

두번째 정수를 입력하세요:

20

division result: 0

- 연산자를 함께 입력 받도록 수정한 후, 연산자에 따라 필요한 함수를 호출하도록 프로그램을 수정하시오.

첫번째 정수를 입력하세요:

10

두번째 정수를 입력하세요:

5

연산자를 입력하세요 (+, -, *, /):

+

addition result : 15

>>>

===== RESTART: C:,

첫번째 정수를 입력하세요:

50

두번째 정수를 입력하세요:

10

연산자를 입력하세요 (+, -, *, /):

-

subtraction result: 40

- 연산자에 all을 추가하고 all을 입력 받았을 때 모든 결과의 합을 출력하도록 수정해보세요.(all이라는 연산자는 실제 없음!)

첫번째 정수를 입력하세요:

30

두번째 정수를 입력하세요:

10

연산자를 입력하세요 (+, -, *, /, all)

all

addition result : 40

subtraction result: 20

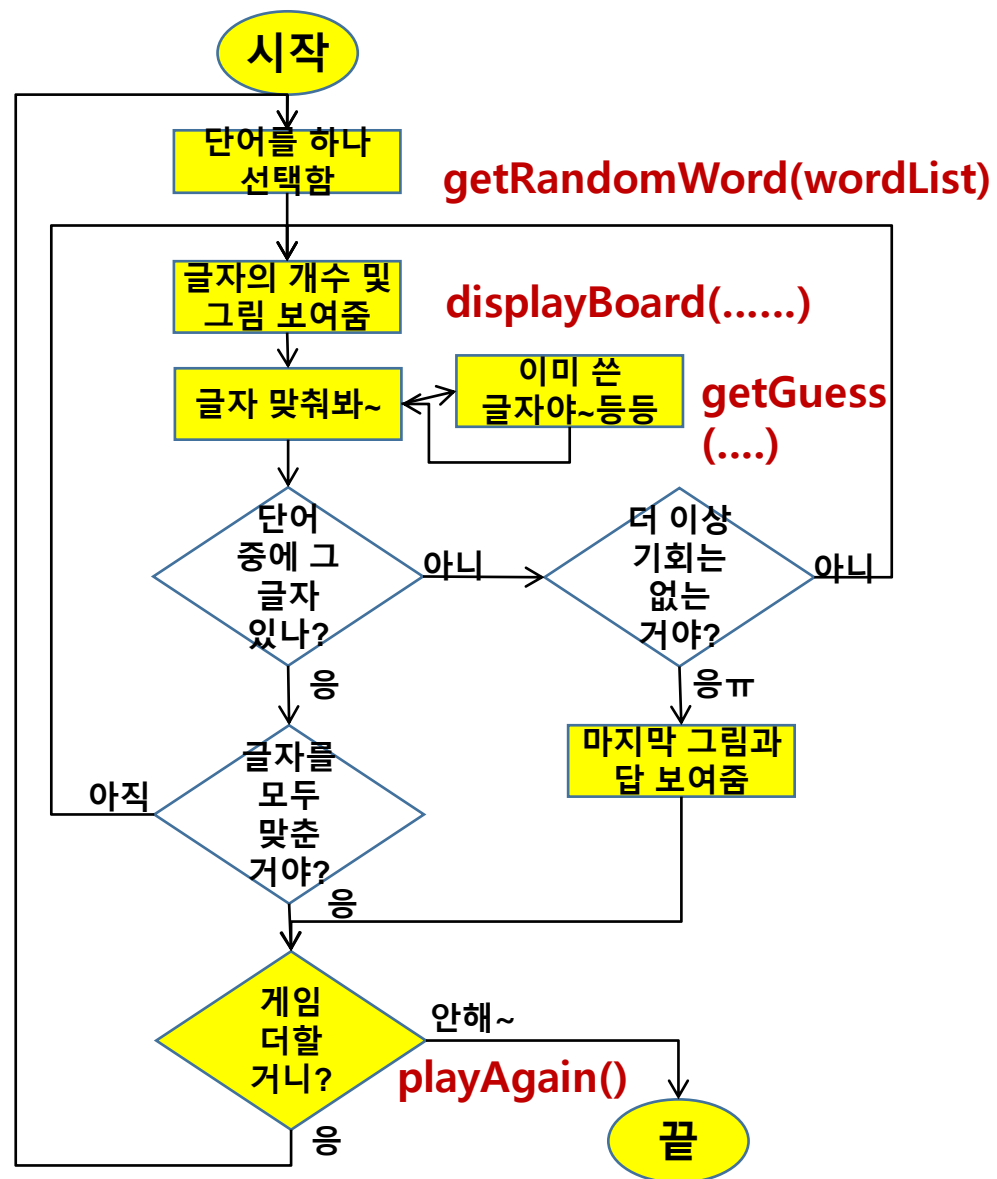
multiplication result: 300

division result: 3

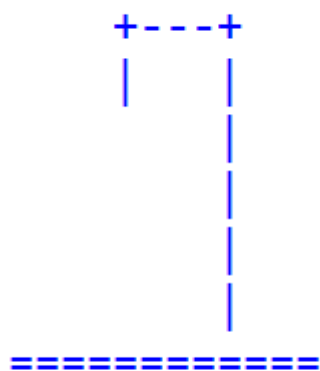
모든 결과의 합은 363 입니다.

응용실습

- Hangman game 요구사항
 - getRandomWord() : 단어 가운데 하나를 골라 리턴하는 함수
 - displayBoard() :
 - 틀린 문자의 수에 해당하는 그림을 나타냄
 - Missed letters:틀린 글자를 나타냄
 - 맞힌 글자는 글자로 아직 못 맞힌 글자는 언더바(_) 로 나타냄
 - getGuesss():
 - 글자를 입력받아 입력 글자가 올바르면 반환함
 - 입력글자가 한글자가 아니면 Please enter a single letter. 를 출력함
 - 입력글자가 이미 이전에 입력한 글자이면 You have already guessed that letter. Choose again. 를 출력함
 - 입력글자가 알파벳이 아니면 Please enter a LETTER. 를 출력함
 - 올바른 글자를 입력 받을 때까지 반복함.
 - playAgain ():
 - 'Do you want to play again? (yes or no)' 메시지 출력, 값을 입력받아 반환

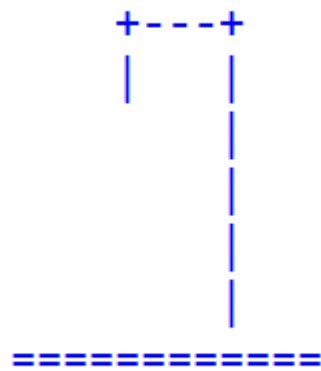


H A N G M A N



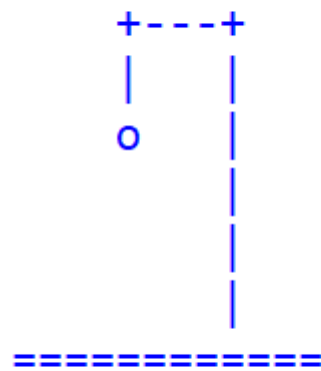
Missed letters:

Guess a letter.
d



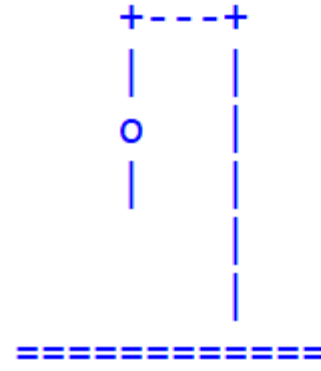
Missed letters:
__d__

Guess a letter.
y



Missed letters:y
__d__

Guess a letter.
f



Missed letters:yf
__d__

Guess a letter.

g

```
  +---+
  |   |
  o   |
 /|\  |
 / \  |
      |
=====
```

Missed letters:yfzhkg

_a_da

You have run out of guesses!

After 6 missed guesses and 2 correct guesses, the word was "panda"

Do you want to play again? (yes or no)

no

HANGMANPICS = ['''

+---+
|
|
|
|
|

=====', '''

+---+
|
o
|
|
|

=====', '''

+---+
|
o
|
|
|

=====', '''

+---+
|
o
/|
|
|

=====', '''

+---+
|
o
/|\
|
|

=====', '''

+---+
|
o
/|\
/
|

=====', '''

+---+
|
o
/|\
/ \
|
|

=====']

- Hangman game에서 사용될 단어
 - ant baboon badger bat bear beaver camel cat clam cobra cougar
coyote crow deer dog donkey duck eagle ferret fox frog goat goose
hawk lion lizard llama mole monkey moose mouse mule newt otter
owl panda parrot pigeon python rabbit ram rat raven rhino salmon
seal shark sheep skunk sloth snake spider stork swan tiger toad
trout turkey turtle weasel whale wolf zebra