

Fastcampus

Computer Science Extension School

Python Basic_Day4

Review

- String
- List
- Tuple
- Conditional Statement

Dictionary, Set

dictionary의 선언

```
dict1 = {}  
print(dict1)
```

dictionary는 key와 value로 이루어져 있으며, 추가하는 법은 다음과 같습니다.

```
dict1 = {'name': 'foo bar'}  
print(dict1)
```

```
dict1 = {'korean': 95, 'math': 100, 'science': [80, 70, 90, 60]}  
print(dict1)
```

```
dict1['english'] = "pass"  
print(dict1)
```

요소 삭제는 del을 활용합니다.

```
del dict1['math']  
print(dict1)
```

key를 활용해 value를 출력하는 법을 알아봅시다.

```
print(dict1['korean'])
```

key만 출력하는 법을 알아봅시다.

```
print(dict1.keys())
```

value만 출력할땐 이렇게 합니다.

```
print(dict1.values())
```

key와 value를 함께 출력합니다.

```
print(dict1.items())
```

Small Quiz

$A = \text{'fastcampus'}$

$B = \text{'python'}$

$A \cup B$

$A \cap B$

$A - B$

$A \Delta B$

Set

- 수학 집합 연산을 쉽게 하기 위해 만든 자료형
- 순서없음
- 중복없음

Set

Set 선언

```
ppap = {'pen', 'apple', 'pineapple', 'pen'}  
print(ppap)
```

```
'apple' in ppap  
'applepen' in ppap
```

```
pineapple = set('pineapple')  
pineapple
```


Set

```
A = set('fastcampus')
```

```
B = set('python')
```

```
A ∪ B == A | B
```

```
A ∩ B == A & B
```

```
A - B == A - B
```

```
A Δ B == A ^ B
```

For, while

```
for 변수 in (리스트 or 문자열):  
    실행문1  
    ...
```

```
for i in ["python", "java", "golang"]:  
    print(i)
```

For, while

```
sum = 0
for i in range(1,11):
    sum += i
    sum = sum + i
    print(sum)
```

For, while

```
while 조건:  
    실행문1  
    ...
```

```
while name != "foo bar":  
    name = input("What's your name? ")  
    print("Hi, " + name + "So, where is foo bar?")
```

```
while 1:  
    print("Hello world!")
```

Fizzbuzz

1부터 100까지 반복

3의 배수 = "Fizz"

5의 배수 = "Buzz"

15의 배수 = "FizzBuzz"

나머지 = 그 숫자

Fizzbuzz

```
num = eval(input("type the number: "))

for i in range(1, num + 1):
    if i % 15 == 0:
        print("fizzbuzz")
    elif i % 3 == 0:
        print("fizz")
    elif i % 5 == 0:
        print("buzz")
    else:
        print(i)
```

Refactoring numguess

```
import random

answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")

while True:
    guess = eval(input("Hi "+ username + ", guess the number"))

    if guess == answer:
        print("Correct! The answer was ", str(answer))
        break
    else:
        print("That's not what I wanted!! Try again!!")
```

give a hint!!

```
import random

answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")

while True:
    guess = eval(input("Hi, "+ username + "guess the number: "))

    if guess == answer:
        print("Correct! The answer was ", str(answer))
        break
    elif guess > answer:
        print("Too high!! Try again!!")
    elif guess < answer:
        print("Too Low!! Try again!!")
```


limit trial

```
import random

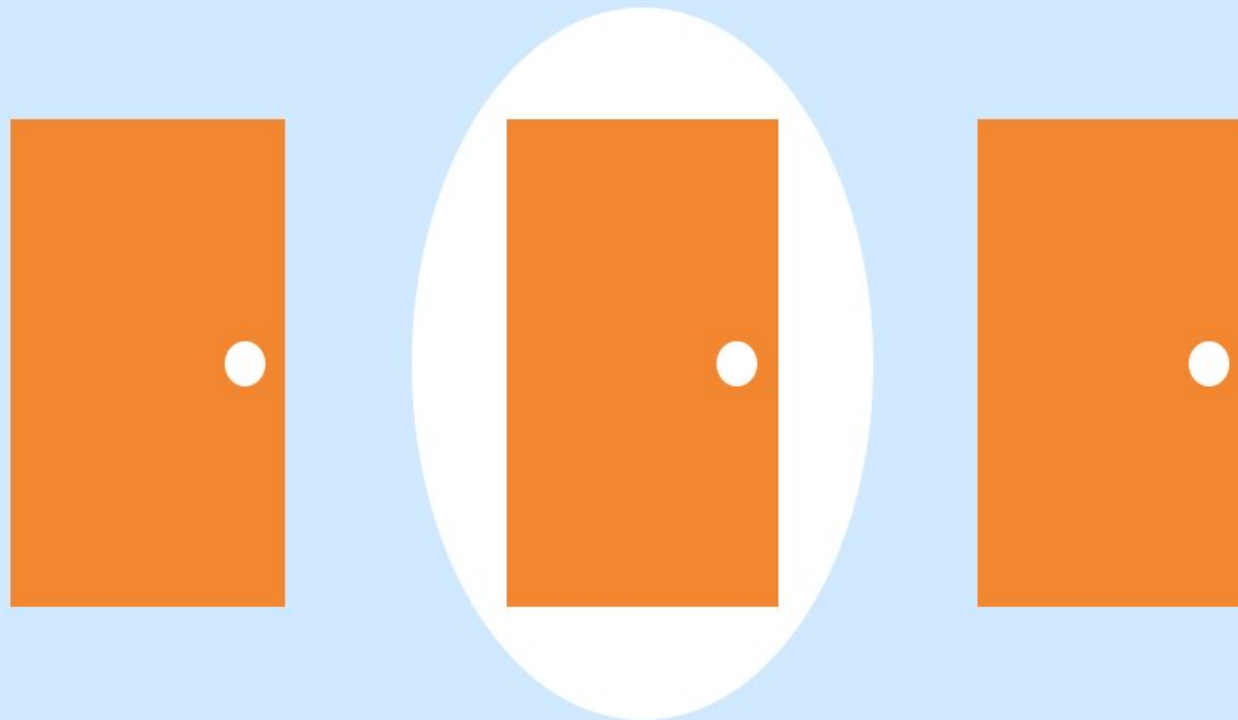
answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")
trial = 5
while trial:
    guess = eval(input("Hi, "+ username + ". guess the number: "))

    if guess == answer:
        print("Correct! The answer was ", str(answer))
        break
    elif guess > answer:
        trial -= 1
        print("Too high!! Try again!!(%d times left)" % (trial))
    elif guess < answer:
        trial -= 1
        print("Too Low!! Try again!!(%d times left)" % (trial))

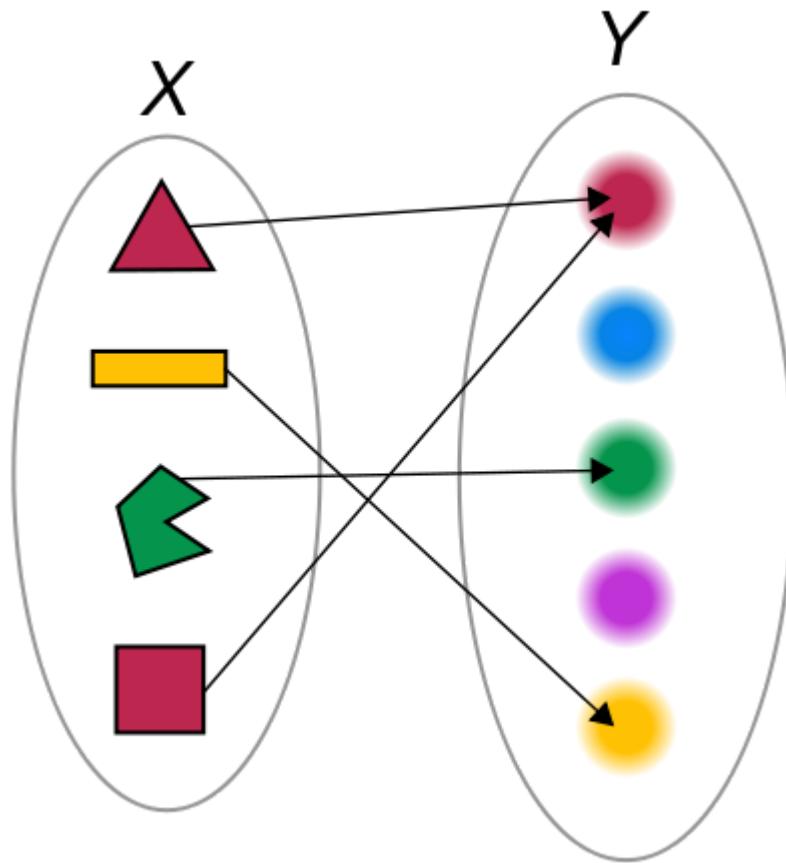
if trial == 0:
    print("You are Wrong! The answer was ", str(answer))
```

Monty Hall Problem

MONTY HALL PROBLEM

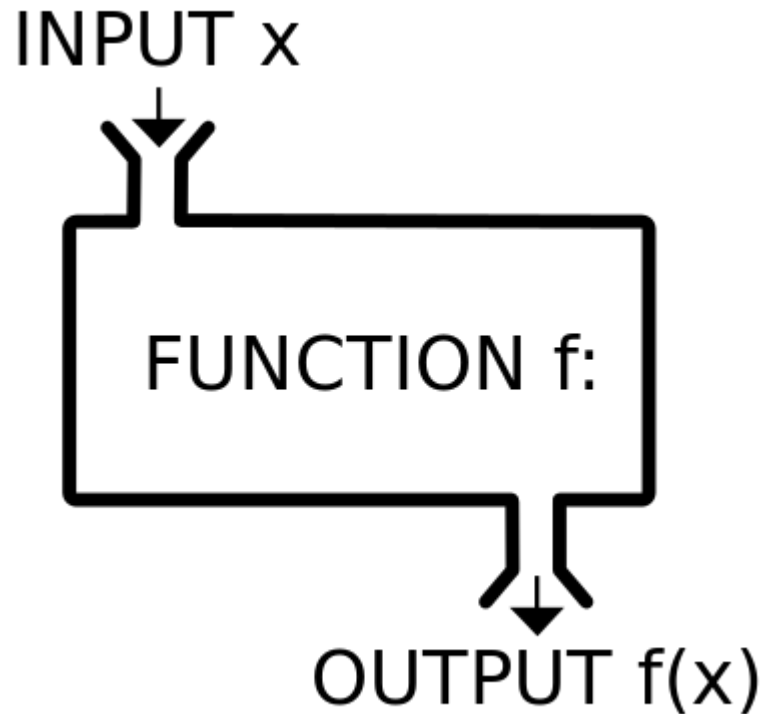


function



- 수학적 정의: 첫 번째 집합의 임의의 한 원소를 두 번째 집합의 오직 한 원소에 대응시키는 대응 관계
- x : 정의역 y : 공역

function



- 프로그래밍에서의 함수: 입력값을 내부에서 어떤 처리를 통해 결과값을 출력하는 것

function

```
def function(parameter):  
    실행문1  
    실행문2  
    ...  
    return output
```

function

```
def awe_sum(a,b):  
    result = a + b  
    return result  
  
a = 2  
b = 3  
print(awe_sum(a,b))
```