

Fastcampus

컴퓨터공학 입문 스쿨

Pytuhon Basic_Day3

Recap

- Pythonic way
- input
- list, tuple
- string

Index

- dictionary
- set
- Conditional statement
- Iterations
- function

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 = 'fastcampus'

B = '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`

조건문

Let's get back to the Day1

배가 고프다!!!

- case 1: 집이라면
 - 밥이 있다면
 - 밥이 없다면
- case 2: 밖이라면
 - 현금이 10만원 초과라면
 - 현금이 5만원 초과라면
 - 현금이 없다면

If

if 조건:

실행문

if 조건1 **and** 조건2:

실행문

if 조건1 **or** 조건2:

실행문

if not 조건:

실행문

Comparison Operators

x == n

x != n

x < n

x > n

x <= n

x >= n

if

```
if 현금 > 100000:  
    레스토랑으로 간다
```

```
cash = 120000  
if cash > 100000:  
    print("go to restaurant")
```

else

```
if 조건:  
    실행문1  
else:  
    실행문2
```

```
cash = 120000  
if cash > 100000:  
    print("go to restaurant")  
else:  
    print("go to cvs")
```

else if

```
if 조건1:  
    실행문1  
else:  
    if 조건2:  
        실행문2  
    else:  
        실행문3
```

```
cash = 120000  
if cash > 100000:  
    print("go to restaurant")  
else:  
    if cash > 50000:  
        print("go to bobjib")  
    else:  
        print("go to cvs")
```

if in else in if in else in ..

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
else:
    if cash > 50000:
        print("go to bobjib")
    else:
        if cash > 30000:
            print("go to buffet")
        else:
            if cash > 20000:
                print("go to ramen store")
            else:
                if cash > 10000:
                    print("go to chinese restaurant")
                else:
                    print("go to cvs")
```

elif

```
if 조건1:  
    실행문1  
elif 조건2:  
    실행문2  
elif 조건3:  
    실행문3  
...  
else:  
    실행문n
```

elif

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
elif cash > 50000:
    print("go to bobjib")
elif cash > 30000:
    print("go to buffet")
elif cash > 20000:
    print("go to ramen store")
elif cash > 10000:
    print("go to chinese restaurant")
else:
    print("go to cvs")
```

numguess

```
import random

answer = random.randint(1,100)
print(answer)
```

numguess

```
username = input("Hi there, What's your name?? ")  
guess = eval(input("Hi, "+ username + "guess the number: "))  
  
if guess == answer:  
    print("Correct! The answer was ", str(answer))  
else:  
    print("That's not what I wanted!! The answer was ", str(
```

numguess advanced!!

how to make it with more fun??

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

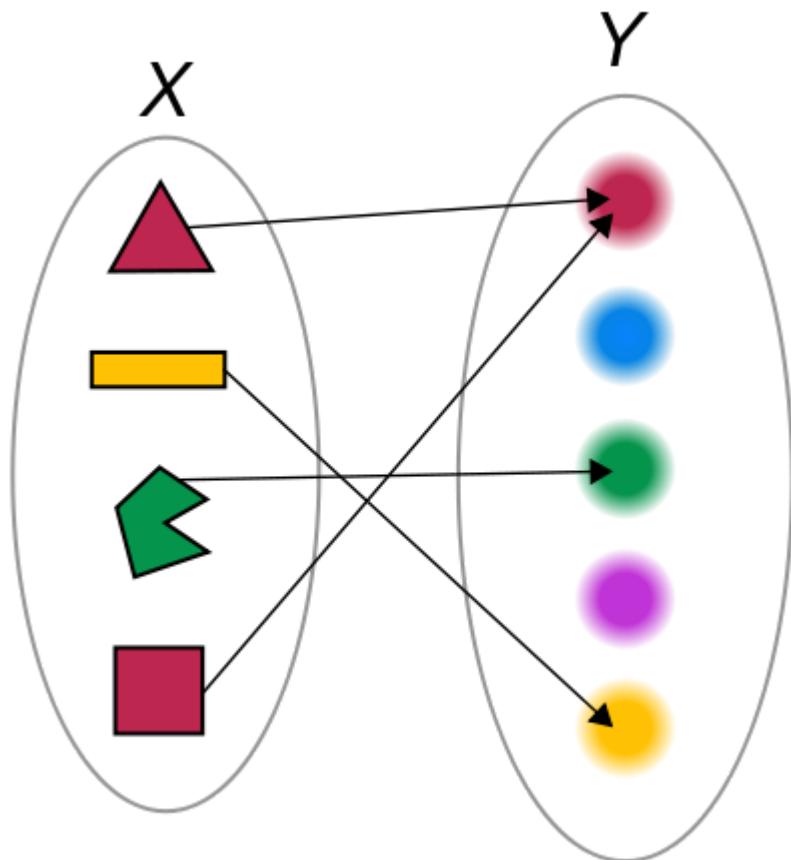
```
import random

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))

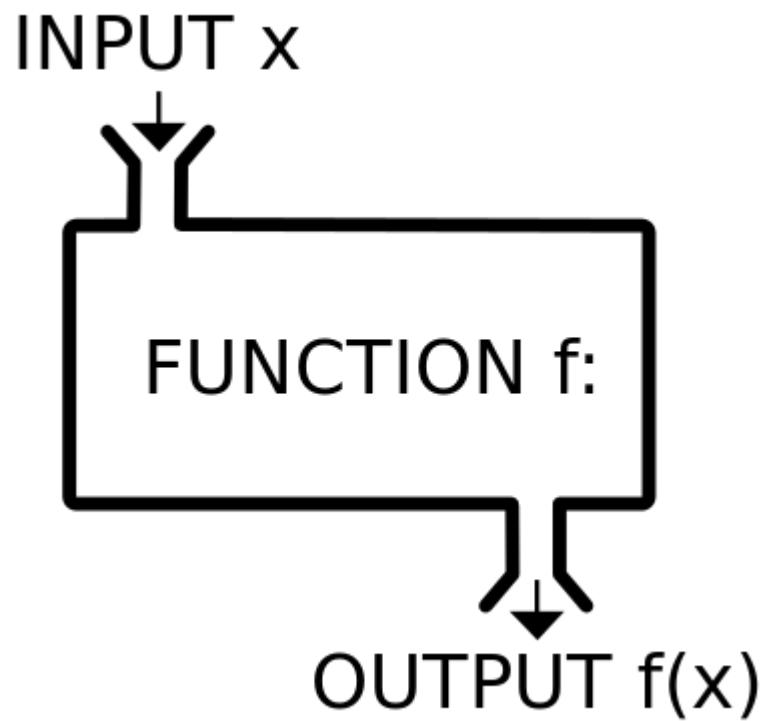
if trial == 0:
    print("You are Wrong! The answer was ", str(answer))
```

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
    ...
```

function

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

a = 2
b = 3
print(awe_sum(a,b))
```

function without input

```
def print_hello():
    return "hello"

result_hello = print_hello()
print(result_hello)
```

function without return

```
def func_wo_return(a):
    print("This is function without return for " + str(a) + " ti
func_wo_return()
```

function with multiple return

```
def mul_return(a):
    b = a + 1
    return a,b
```

return skill

```
def id_check(id):
    if id == "admin":
        print("invalid id: admin")
        return
    print("valid id: ", id)
```

parameter with initialize

```
def say_hello(name="Fool", nick=True):
    print("Hi, ", name)
    if nick == True:
        print("But, you are Fool")
    else:
        print("Oh, you are not Fool")
```

초기값을 설정할땐 항상 그 인자를 마지막에 두어야 합니다.

arguments

```
def mul_sum(*args):
    sum = 0
    for i in args:
        sum += i
    return sum
```

keyword arguments

```
def show_kwargs(**kwargs):
    print(str(kwargs))

show_kwargs(a=10, b="google")
```

keyword arguments

```
def kwargs_url(server, port, **query):
    url = "https://" + server + ":" + port + "?"
    for key in query.keys():
        url += key + "=" + query[key] + "&"
    return url

kwargs_url("localhost", "8080", utm_source="google", keyword="nav")
```

variable outside function

```
a = "hello"
def glob_test(a):
    a += "world"
    return a

glob_test(a)
print(a)
```

```
a = "hello"
def glob_test(x):
    x += "world"
    return x

glob_test(a)
print(a)
```

variable outside function

```
def glob_test2(x):
    x += "world"
    return x

glob_test2("hello")
glob_test2(x)
```

So, how to globalize

(1) using return

```
a = "hello"
def glob_test(a):
    a += "world"
    return a

a = glob_test(a)
print(a)
```

So, how to globalize

(2) use global

```
a = "hello"
def glob_test(a):
    global a
    a += "world"
    return a

glob_test(a)
print(a)
```

`global`이라는 명령을 사용하여 전역변수로 사용하게 되면 함수는 독립성을 잃게 되어 함수가 외부변수에 의존적이게 됩니다.

Leap year

4로 나뉘어 떨어지면 윤년,
100으로 나뉘어 떨어지면 평년,
400으로 나뉘어 떨어질땐 윤년

Leap year(answer)

```
leap = False
def is_leap(y):
    if y % 4 == 0 and (y % 100 != 0 or y % 400 == 0)
        leap = True
    return leap

y = int(input("Is leap?? "))
print(is_leap(y))
```

numguess with function

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

Recursive

```
times = int(input("How many times want to curse the beast??: "))
def recurse_beast(a):
    if a == 0:
        print("curse complete!")
    else:
        print("Fusion!!!(%d times left)" % a - 1)
        recurse_beast(a-1)

recurse_beast(times)
```

숙제

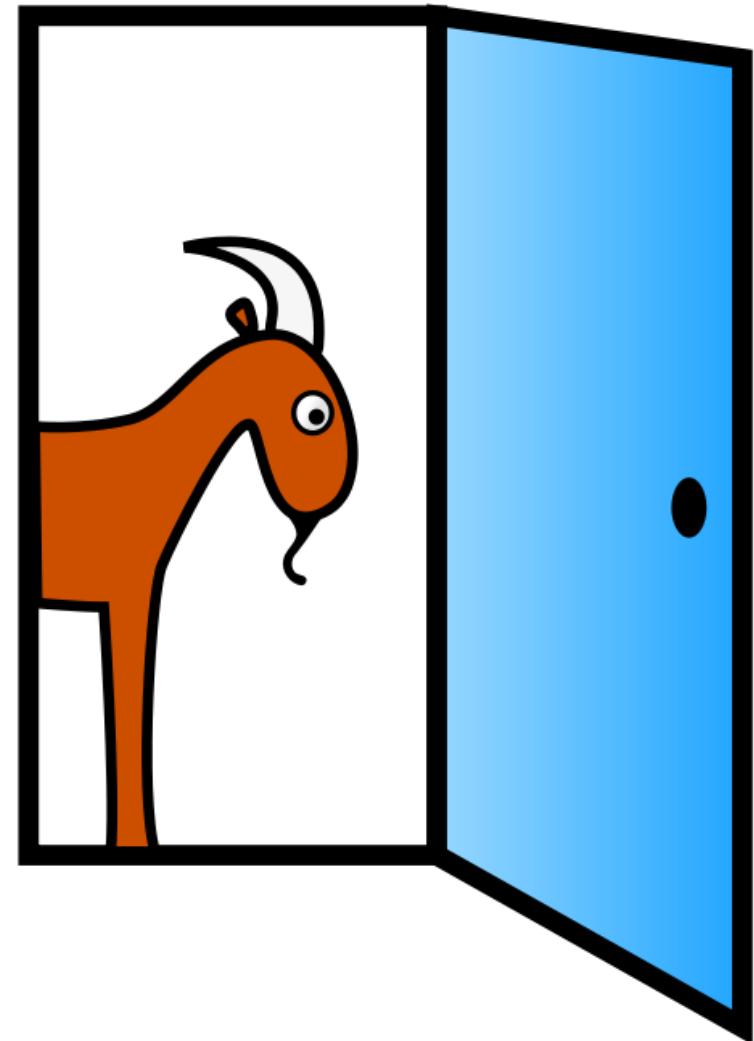
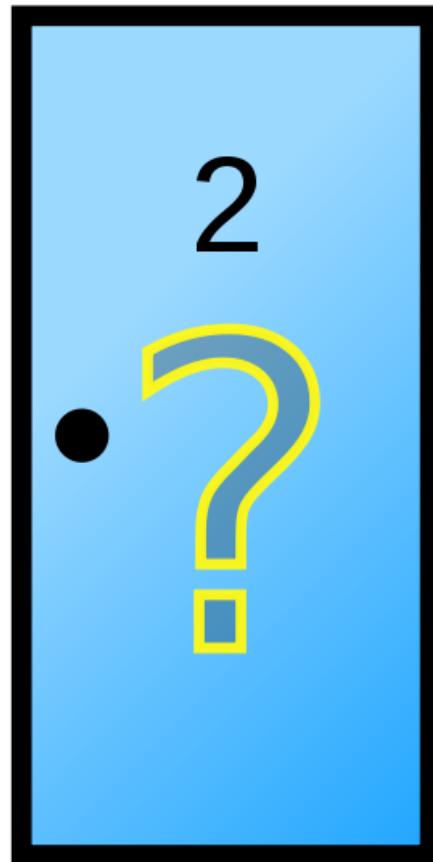
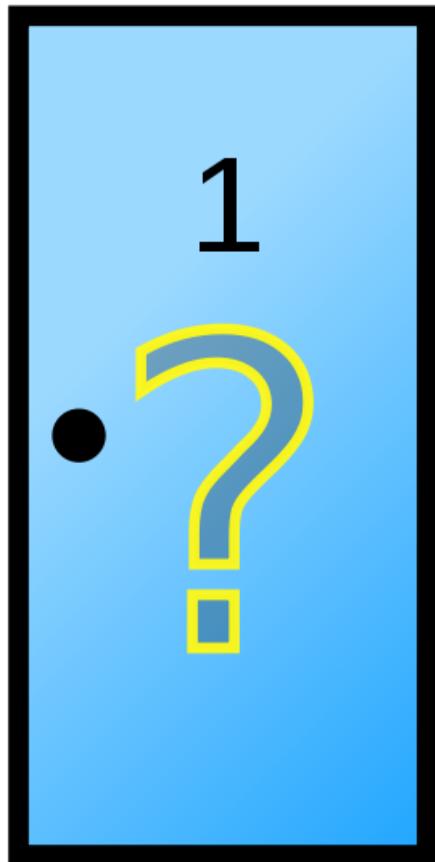
Monty Hall Problem



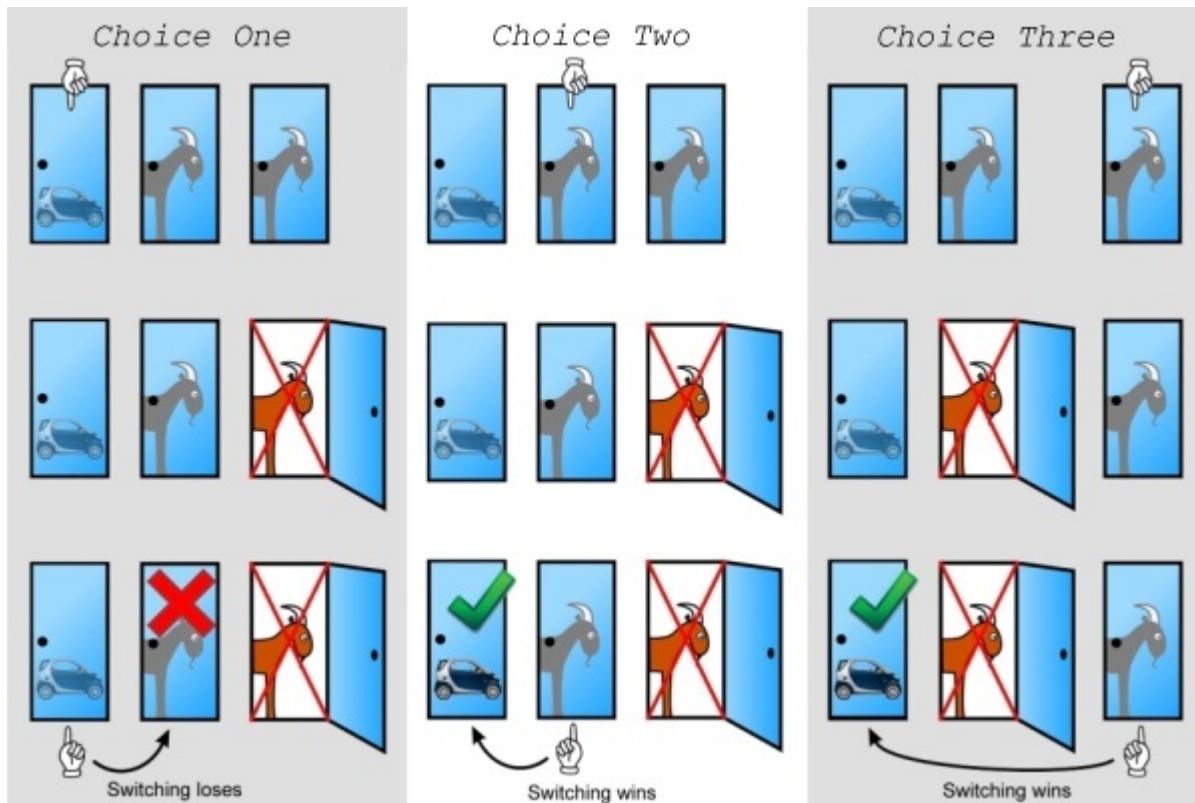
Monty Hall Problem



Monty Hall Problem



Monty Hall Problem



Let's Simulate This Problem