

**Fastcampus**

**Computer Science Extension School**

**Python Basic\_Day3**

# Review

- PEP
- data type
- syntax
- input
- string

**Before we start**



# HackerRank

[Hackerrank 30 days of code](#)

[Hackerrank python challenges](#)

# Strings

# Strings

```
some_string = "python"  
len(some_string)
```

- index

p	y	t	h	o	n
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

```
some_string[3:5] = "ho"  
some_string[1:5:2] = "yh"  
some_string[::] = some_string[0:len(some_string):1]  
some_string[::-1] = some_string[-1:-len(some_string):-1]  
some_string[::-1] = "nohtyp"
```

## but, strings are immutable

```
>>> some_string = "python"

>>> some_string[0] = "c"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment

>>> some_string = "c" + some_string[1:]
```

# String Functions

```
func = "python is easy programming language"  
func.count('p')
```

```
func.find('p')
```

```
comma = ","  
func = comma.join('python')
```

```
func.split(',')
```

```
python_is_easy = "python is easy"  
python_is_easy.split()
```

```
python_is_easy.replace("python", "golang")
```

# String Functions

```
some_string = "    computer    "  
some_string.strip()
```

```
some_string = ",,,Fastcampus..."  
some_string.strip(",")  
some_string.strip(".")
```



## String Formatting - old way

```
print("I have a %s, I have an %s." % ("pen", "apple"))
```

- %s - string
- %c - character
- %d - Integer(decimal)
- %f - floating-point
- %o - 8진수(Octal)
- %x - 16진수(hexadecimal)
- %% - %

## String Formatting - New way

```
print("I have a {}, I have an {}".format("pen", "apple"))
```

```
print("I have a {0}, I have an {1}".format("pen", "apple"))
```

```
print("I have a {0}, I have an {0}".format("pen", "apple"))
```

# Toggl

<https://blog.toggl.com/wp-content/uploads/2016/12/toggl-it-jobs-explained-with-changing-lightbulb.jpg>

<https://assets.toggl.com/images/toggl-how-to-save-the-princess-in-8-programming-languages.jpg>

# List, Tuple

## List

```
animals = [' ', ' ', ' ']
```

## Tuple

```
animals = (' ', ' ', ' ')
```

## List

**빈 list를 선언합니다. 선언과 동시에 값을 채워넣을 수 있습니다.**

```
lang = ["python", "c", "java", "golang"]  
lang = []
```

**list에 요소를 추가합니다.**

```
lang.append("python")  
lang.append("java")  
lang.append("golang")  
print(lang)
```

**혹은 특정한 위치에 원하는 값을 추가할 수 있습니다.**

```
lang.insert(1, "c")  
print(lang)
```

**특정 요소를 삭제할 수도 있습니다.**

```
lang.remove("golang")  
print(lang)
```

**혹은 리스트에 있던 값을 빼낼 수도 있습니다.**

```
java = lang.pop(2)  
print(lang)  
print(java)
```

**리스트를 정렬하는 법을 알아보니다.**

```
numbers = [2, 1, 4, 3]
```

```
print(numbers)
```

```
numbers.sort()
```

```
print(numbers)
```

**리스트를 역순으로 출력하고 싶을땐 이렇게 한답니다.**

```
numbers = [2, 1, 4, 3]
```

```
numbers.reverse()
```

```
print(numbers)
```

**리스트를 내림차순으로 정렬하려면??**



## 1. sort -> reverse

```
numbers.sort()  
numbers.reverse()
```

## 2. sort(reverse=True)

```
numbers.sort(reverse=True)
```

**특정 값의 위치를 출력할때 이렇게 합니다.**

```
index_of_two = numbers.index(2)  
print(index_of_two)
```

**리스트끼리 더할 때 extend를 활용합니다.**

```
numbers += [5, 6]  
print(numbers)  
numbers.extend([7, 8])  
print(numbers)
```

## Tuple

**Tuple은 괄호를 이용해 선언할 수 있습니다.**

```
tuple1 = (1, 2, 3, 4)
```

**tuple은 삭제나 추가가 불가능합니다.**

```
del tuple[1]  
tuple1[1] = 'c'
```

**tuple끼리 더하거나 반복하는 것은 가능합니다.**

```
tuple2 = (5, 6)
```

```
print(tuple1 + tuple2)
```

```
print(tuple1 * 3)
```

**tuple은 값을 편하게 바꿀 수 있습니다.**

```
x = y
y = x (x)

temp = x
x = y
y = temp

(x,y) = (y,x)
```

**혹은 함수에서 하나 이상의 값을 반환할 때 사용합니다.**

```
def quot_and_rem(x,y):
    quot = x // y
    rem = x % y
    return (quot, rem)

(quot, rem) = quot_and_rem(3,10)
```

## List <-> Tuple

```
list((1,2))  
tuple([1,2])
```

# 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

$x == n$

$x != n$

$x < n$

$x > n$

$x \leq n$

$x \geq n$



# 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")
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")
```

# 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??