Fastcampus Data Science Extension SCHOOL

Python

Index

- Good Resources to learn skills
- List, Tuple
- Conditional Statements
- Conditional Expressions

HackerRank

https://www.hackerrank.com/

- 30 Days of Code
- Python
- Algorithms
- Data Structures

Toggl

https://toggl.wpengine.com/wp-content/uploads/2018/05/freelancer-comic-2.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 알아봅니다.

```
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 출력할땐 이렇게 합니다.

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

리스트끼리 더할 땐 extend를 활용합니다.

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

Membership Operator

`in` - `'cat' in ['fox', 'dog', 'cat']` => True

`not in` - `'wolf' not in ['fox', 'dog', 'cat']` => True

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

Conditional Statements

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")
```

Do it your self!

Numguess

- 1부터 100까지 정수 중 하나를 `answer` 라는 변수에 할당
- 사용자로 부터 임의의 값 하나를 받아 `guess` 라는 변수에 할당
- `answer` 와 `guess` 를 비교하여 정답여부를 출력

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??

삼항연산자

Ternary Operators

단항연산자

- 부호(+,-), not
  - +3
  - -0.2
  - not True

이항연산자

- 대부분의 연산자
  - 3 + 1
  - 2 * 3.14

Ternary Operators in Python

Conditional Expressions

Conditional Expressions

- `a` if `condition` else `b`
- condition == True => a
- condition == False => b

Conditional Expressions in Conditional Expressions

`a` if `condition` else `b` if `condition` else `c`

```
if condition:
        result = 'a'
else:
        if condition:
                result = 'b'
        else:
                result = 'c'
```

Expression is Expression, not Statement!!

Possible

```
if Condition:
        result = b
else:
        another_result = c
```

Impossible

```
result = b if condition else another_result = c
```

And, Conditional Expressions 는 파이썬 연산자 중 가장 낮은 순위를 가집니다.

Do it yourself!

사용자가 입력한 요일('월'~'일' or '월요일'~'일요일')을

- 평일일 경우, "평일이네요ㅜㅜ"
- 주말일 경우, "주말입니다^^"
- 잘못 입력한 경우, "요일을 입력하세요"

을 출력하도록 Conditional Statements 와 Conditional Expressions로 각각 구현하세요