

Python 101

Lec02

Ifs and Loops

thoum

April 9, 2019

Before we begin

```
#works on iterables
t = (1, 2, 3)
print(t+t)
print(t+t+t)

l = [1,2,3]
print(l * 3)

s = 'abc'
print(s * 2)

#unpacking iterables
x, y, z = (1, 2, 3)
x, *y = [1, 2, 3, 4, 5]
x, y, *z = 'hiiam'
```

If

```
if some_boolean:  
    pass #do something
```

If examples

```
score = int(input())

# C style Boolean
if 90 <= score and score <= 100:
    print('A')
# use THIS!
elif 80 <= score <= 89:
    print('B')
elif 70 <= score <= 79:
    print('C')
elif 60 <= score <= 69:
    print('D')
else: # when every boolean above is False
    print('F')
```

More examples

```
# Membership
pengram = "The quick brown fox jumps over the
          lazy dog"
if 'e' in pengram:
    print("YES")

# even?
if 33 % 2 == 0:
    print("EVEN")

if [1,2,3]:
    print("Non-zero is True")
if []:
    print("Empty is Null")
if '':
    print("NULL is False")
```

For Loops

~~Repeat N times~~

Iterate members of a sequence.

For Loops, C style

```
#include <stdio.h>

int main()
{
    int i = 0;
    for (i = 0; i < 10; i++) {
        printf("%d", i);
    }
    return 0;
}
```

```
for value in iterable:  
    pass
```


For Loops

```
for i in range(6):  
    print("hello!")  
  
nums = [1, 2, 3, 4, 5, 6]  
  
# avoid magic numbers  
for i in range(6):  
    print(nums[i])  
print()  
  
# this lets us change nums[i]  
for i in range(len(nums)):  
    print(nums[i])  
    # something new?  
    nums[i] += 1
```

For Loops

```
# this lets us read the values
for e in nums:
    print(e)
print()

for n in nums:
    if n == 2:
        print("We found 2!")
        break

x = 0
for n in nums:
    x += n
print(x)
```

For Loops Nested

```
lst = [[1,2,3], [4,5,6], [7,8,9]]
```

```
#double for loop
```

```
for row in lst:
    for c in row:
        print(c, end=' ')
    print()
```

```
#triple?
```

```
for i in range(1000):
    for j in range(1000):
        for k in range(1000):
            print(i, j, k)
```

While Loops

Repeat *while* condition(boolean) is satisfied.

```
while condition:  
    pass
```

```
while True:
    print("Forever")

while False:
    print("Can I print?")

# Realistic use case
while True:
    s = input("Please Enter 'hi': ")
    if s == 'hi':
        break
```

Practice

Print from 1 to 10

Using for loop and while loop. Hint:

Not So Easy Practice

Output: print 99dan from 1 to N. Hint:

Practice with stars!

Input: N

Print:

```
*  
**  
***  
  
  *  
 **  
***  
  
   *  
  ***  
*****
```

Hint:

??????????

Using for loops, create list of even numbers in $[0, 10]$

Hint:

??????????

```
lst = []  
for n in range(0, 11, 2):  
    lst.append(n)
```

or

```
for n in range(0, 11):  
    if n % 2 == 0:  
        lst.append(n)
```

Comprehension

```
lst = [e for v in iter (if condition)]
```

In plain text:

Our NEW_LIST's elements =

FOR each VAR in SOME_ITER, (that meets CONDITION) EXP

```
for var in some_iter:  
    if condition:  
        new_list.append(exp)
```

Comprehension

```
lst = [n for n in range(10)]
lst == list(range(10))

# 1
# 2
# 3
input_nums = [int(input()) for _ in range(5)]
# convention: we use _ when we don't use it.

# 1 2 3 ....
input_nums = [int(n) for n in input().split()]

lst = [n**2 for n in range(10)]

lst = [[0] * 10 for _ in range(10)]

uppers = [c for c in 'iaMaSTring' if c.isupper()]

# try to avoid this. WHY?
lst = [print("HELLO") for _ in range(10)]
```

Generators

Lazy version of comprehension `[]` \rightarrow `()`

Easy Practice

Input: 10 numbers

Output: sum of the numbers

How: Using for loops and list comprehension