School Nova
Computer Science

Definite iteration: “for” loop
Conditional statement: “if”

10/27/2019
By Oleg Smirnov

Homework comments

```
number = 5
number = int(5)     # unnecessary int()

name = input("What is your name?")
name = str(input("What is your name?")     # unnecessary str()
```

Iterations: Definite loops

Definite iteration – the loop is repeated a certain number of times that you define.

```
for current_age in range(30, 65):
    print(f"You are {current_age} years old. It's too early to retire")
print(f"You reached the retirement age of {current_age + 1} years.")
```

Notice that 30 is **included** while 65 is **excluded**.

```
for i in (30, 31, 32, 33, 34, 35, 36, 37): print(i)     # possible but inefficient
```

Iterations: Definite loops

range(x, y) is a sequence of integers from *x* (included) to *y* (excluded)
You can see all elements in the sequence:

print(list(range(x, y))), or
for i in range(x, y): print(i)

range(x) is a sequence of integers from zero (!) to *x* (excluded). The last elements in the sequence is *x* – 1.

for i in range(4): print(i)

Output:
0
1
2
3

For loops: Using step

You can add a **step** to the range function (and, therefore, the for loop).

print(list(range(0, 105, 5))) # here the step is 5
for i in range(0, 105, 5): print(i)

For **reverse** loops you can you step = -1, for example:

for i in range(20, 10, -1): print(i)

You can only use **integers** with range(). You can't use float type!
However, you can go around this, for example:

I need to print all tenths between 0 and 1
for i in range(0, 11): print(i / 10)

Classroom exercise I

Task:

Calculate and print the sum and product of all odd numbers between 1 and 20.

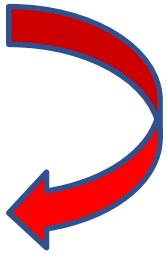
Solution:

```
sum, product = 0, 1
for i in range(1, 20, 2):
    sum = sum + i
    product = product * i
print(f"Sum is equal to {sum}. Product is equal to {product}.")
```

Definite loops, break, and continue

Break and **continue** commands work similar to how they work with the indefinite loop while.

```
for i in <condition>:
        statement 1
        statement 2
        break
        statement 3
        statement 4
statement 5
```



```
for i in <condition>:
        statement 1
        statement 2
        continue
        statement 3
        statement 4
statement 5
```

For loop: Break and Continue Example

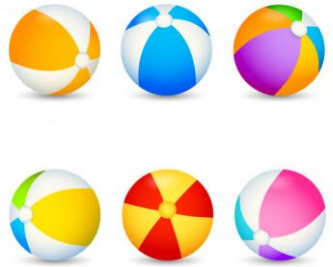

```
for i in range (5):
    print(i)
    break
    continue
```

```
>>>
0
```

```
for i in range (5):
    print(i)
    continue
    break
```

```
>>>
0
1
2
3
4
```

For loops: going over a finite collection of objects



Alternatively, a definite loop may go over a finite collection of objects.
One example of a collection of objects you have already seen: strings.
Strings consist of letters:
for i in "School Nova": print(i)

There are many other types of finite collections of objects (which we will study closely soon), for example: lists, tuples, dictionaries, sets.

animals = ["cat", "dog", "cow"] # this is a list; check type(animals)
for i in animals: print(i)

Above, you don't define the number of iterations. Instead, the number of iterations is equal to the number of elements in the list.

Classroom exercise II

Task:

Using the for loop, calculate the number of characters in your full name (first and last, ok to include spaces). Verify that your answer is correct using len() function, which counts the number of characters in a string.

Solution:

```
name = "Oleg Smirnov"
print(len(name))
letters = 0
for i in name:
    letters = letters + 1
print(letters)
```

Conditional statement IF (first look)

General structure:

```
if True:
    print("Execute this code only if True")
else:
    print("Execute this code only if False")
print("Execute this code always")
# try replacing “if True” above with various conditions
```

else statement is optional:

```
if True:
    print("Execute this code if True")
print("Execute this code always")
```

if...elif...else statement will be examined at a later date