LOOP AND SELECTION STATEMENT

LOOP STATEMENT

Definite Iteration: The `for` Loop

We begin our study of control statements with repetition statements, also known as **loops**, which repeat an action. Each repetition of the action is known as a **pass** or an **iteration**. There are two types of loops—those that repeat an action a predefined number of times (**definite iteration**) and those that perform the action until the program determines that it needs to stop (**indefinite iteration**). In this section, we examine Python's **`for` loop**, the control statement that most easily supports definite iteration.

```
for <variable> in range(<an integer expression>):
    <statement-1>
    .
    .
    <statement-n>
```

EXAMPLE

- Contoh 1

```
for eachPass in range(4):
    print("It's alive!", end = " ")
```

- Contoh 2

```
number = 2
exponent = 3
product = 1
for eachPass in range(exponent):
    product = product * number
    print(product, end = " ")
```

- Contoh 3

```
for count in range(4):
    print(count, end = " ")
```

- Contoh 4

```
product = 1
for count in range(4):
    product = product * (count + 1)
product
```

```
for <variable> in range(<lower bound>, <upper bound + 1>):
    <loop body>
```

```
>>> lower = int(input("Enter the lower bound: "))
Enter the lower bound: 1
>>> upper = int(input("Enter the upper bound: "))
Enter the upper bound: 10
>>> theSum = 0
>>> for number in range(lower, upper + 1):
        theSum = theSum + number
>>> theSum
```

```
for <variable> in <sequence>:
    <do something with variable>

for number in [6, 4, 8]:
    print(number, end = " ")


for character in "Hi there!":
    print(character, end = " ")
```

Exercises

1. Write the outputs of the following loops:

   a.
   ```
   for count in range(5):
       print(count + 1, end = " ")
   ```

   b.
   ```
   for count in range(1, 4):
       print(count, end = " ")
   ```

   c.
   ```
   for count in range(1, 6, 2):
       print(count, end = " ")
   ```

   d.
   ```
   for count in range(6, 1, -1):
       print(count, end = " ")
   ```

2. Write a loop that prints your name 100 times. Each output should begin on a new line.

```
if <condition>:
    <sequence of statements>
```

```
if <condition>:
    <sequence of statements-1>
else:
    <sequence of statements-2>
```

```
first = int(input("Enter the first number: "))
second = int(input("Enter the second number: "))
if first > second:
    maximum = first
    minimum = second
else:
    maximum = second
    minimum = first
print("Maximum:", maximum)
print("Minimum:", minimum)
```

```
if <condition-1>:
    <sequence of statements-1>
elif <condition-n>:
    <sequence of statements-n>
else:
    <default sequence of statements>
```

```
number = int(input("Enter the numeric grade: "))
if number > 89:
    letter = 'A'
elif number > 79:
    letter = 'B'
elif number > 69:
    letter = 'C'
else:
    letter = 'F'
print("The letter grade is", letter)
```

Exercises

1. Assume that **x** is 3 and **y** is 5. Write the values of the following expressions:

   a. `x == y`

   b. `x > y - 3`

   c. `x <= y - 2`

   d. `x == y or x > 2`

   e. `x != 6 and y > 10`

   f. `x > 0 and x < 100`

2. Assume that **x** refers to a number. Write a code segment that prints the number's absolute value without using Python's **abs** function.

3. Write a loop that counts the number of space characters in a string. Recall that the space character is represented as `' '`.

```
while <condition>:
    <sequence of statements>
```

```
theSum = 0.0
data = input("Enter a number or just enter to quit: ")
while data != "":
    number = float(data)
    theSum += number
    data = input("Enter a number or just enter to quit: ")
print("The sum is", theSum)
```

Exercises

1. Translate the following **for** loops to equivalent **while** loops:

   a.
   ```
   for count in range(100):
       print(count)
   ```

   b.
   ```
   for count in range(1, 101):
       print(count)
   ```

   c.
   ```
   for count in range(100, 0, -1):
       print(count)
   ```

2. The factorial of an integer N is the product of the integers between 1 and N, inclusive. Write a **while** loop that computes the factorial of a given integer N.

3. The $\log_2$ of a given number N is given by M in the equation $N = 2^M$. Using integer arithmetic, the value of M is approximately equal to the number of times N can be evenly divided by 2 until it becomes 0. Write a loop that computes this approximation of the $\log_2$ of a given number N. You can check your code by importing the **math.log** function and evaluating the expression **round(math.log(N, 2))** (note that the **math.log** function returns a floating-point value).