Introduction to Python

Conditionals

Topics

1) Conditionals
    a) if, if-if, if-elif, if-elif-else
    b) Ternary operator

Conditionals

The reserved word **if** begins an conditional block.

```
if condition:
    block
```

The condition determines if the block is to be executed.

A block contains one or more statements.

The statements inside of a block must be indented the same number of spaces from the left. The standard is 4 spaces.

If block

What’s the output?

```
In[1]: x = -5
       if x > 0:
          print(x)
          print(“x is positive”)
       print(“outside of block”)
```

```
  outside of block
```

If block

What’s the output?

```
In[2]: x = 5
       if x > 0:
           print(x)
           print(“x is positive”)
       print(“outside of block”)
```

```
5
x is positive
outside of block
```

Sequence of Ifs

A sequence of consecutive if statements are independent. None, some or all of them can be executed.

```
In[1]: x = 4
       if x % 2 == 0:
          print(“x is even”)
       if x > 0:
          print(“x is positive”)
```

```
x is even
x is positive
```

Sequence of Ifs

```
In[2]: x = -8
       if x % 2 == 0:
          print(“x is even”)
       if x > 0:
          print(“x is positive”)
```

```
x is even
```

if-elif

An if block followed by a sequence of elif blocks will execute the **first** block whose condition evaluates to True. No block is executed if all conditions evaluate to False.

```
In[1]: x = 25
       if x < 5:
           print(“x is less than 5”)
       elif x < 10:
           print(“x is less than 10”)
       elif x < 15:
           print(“x is less than 15”)
```

Note that all of the above conditions are false and thus no block is executed.

if-elif

```
In[2]: x = 7
       if x < 5:
           print(“x is less than 5”)
       elif x < 10:
           print(“x is less than 10”)
       elif x < 15:
           print(“x is less than 15”)
```

```
x is less than 10
```

Note that only the middle elif block is executed!

if-elif

```
In[3]: x = 1
       if x < 5:
           print(“x is less than 5”)
       elif x < 10:
           print(“x is less than 10”)
       elif x < 15:
           print(“x is less than 15”)
```

```
x is less than 5
```

Note that only the first if block is executed, even though all three conditions are true.

if-elif-else

An `if` statement followed by a sequence of `elif` statements and ending in an `else` statement will execute the first block whose condition evaluates to `True`. If all conditions evaluate to `False`, it will execute the default `else` block.

```
In[1]:  x = 0
       if x < 0:
           print(“x is negative”)
       elif x > 0:
           print(“x is positive”)
       else:
           print(“x is zero”)
```

```
  x is zero
```

if-elif-else

```
In[2]:  x = 10
       if x < 0:
           print(“x is negative”)
       elif x > 0:
           print(“x is positive”)
       else:
           print(“x is zero”)
```

```
  x is positive
```

and, or, not

Use *and*, *or*, and *not* Boolean operators to simplify conditionals.

The following

```
if x > 0:
    if x < 10:
        print(x)
```

is equivalent to

```
if x > 0 and x < 10:
    print(x)
```

and, or, not

The following code prints the quadrant of an ordered (x,y) on the Cartesian plane.

```
x = 4
y = 7
if (x > 0) and (y > 0):
   print(“first quadrant.”)
elif (x < 0) and (y > 0):
   print(“second quadrant.”)
elif (x < 0) and (y < 0):
   print(“third quadrant.”)
elif (x > 0) and (y < 0):
   print(“fourth quadrant.”)
else:
   print(“on x or y axis.”)
```

Output:
first quadrant

and, or, not

The following code prints the quadrant of an ordered (x,y) on the Cartesian plane.

```
x = -26
y = -31
if (x > 0) and (y > 0):
   print(“first quadrant.”)
elif (x < 0) and (y > 0):
   print(“second quadrant.”)
elif (x < 0) and (y < 0):
   print(“third quadrant.”)
elif (x > 0) and (y < 0):
   print(“fourth quadrant.”)
else:
   print(“on x or y axis.”)
```

Output:
third quadrant

Ternary Operators

A *ternary operator* evaluates an expression based on the value of a boolean condition. This is sometimes called *conditional expression* or an *inline if-else* statement.

```
x = 50
grade = “pass” if x >= 60 else “fail”
print(grade)
```

```
fail
```

Ternary Operators

```
x = 60
grade = “pass” if x >= 60 else “fail”
print(grade)
```

```
pass
```

References

1) Vanderplas, Jake, A Whirlwind Tour of Python, O'reilly Media.
2) Richard Halterman, Fundamental of Python Programming.