

02_Conditional_Execution

January 19, 2018

1 Conditional Execution

1.0.1 If Statements

```
In [1]: # The boolean expression after the if statement is called the condition.  
# We end the if statement with a colon character (:) and the line(s) after  
# the if statement are indented.
```

```
x = 3  
  
if x>0:  
    print("x is positive")
```

x is positive

1.0.2 If-Else statement

```
In [2]: # Alternative Execution
```

```
if x >0:  
    print("x is positive")  
else:  
    print("x is not positive")
```

x is positive

Write a program to prompt the user for an integer and then checks if it is odd or even.

```
In [3]: num = int(input("Enter an integer: "))
```

```
if num%2 == 0:  
    print(str(num) + " is even")  
else:  
    print(str(num) + " is odd")
```

Enter an integer: 3

3 is odd

1.0.3 Nested If Statement

In [4]: *# try different values for x and y*

```
x = 3
y = 7

if x == y:
    print("x and y are equal")
else:
    if x < y:
        print("x is less than y")
    else:
        print("x is greater than y")
```

x is less than y

1.0.4 Elif Statements

In [5]: *# Must be after an if statement. elif statement allows you to check multiple expressions for True and execute a block of code as soon as one of the conditions evaluate to True. Similar to the else, the elif statement is optional. However, unlike else, for which there can be at most one statement, there can be an arbitrary number of elif statements following it.*

```
x = -3

if x > 0:
    print("x is positive")
elif x < 0: #elif needs a condition
    print("x is negative")
else:
    print("x is zero")
```

x is negative

1.0.5 Catching Exceptions using try and except

In [6]: *# This is useful, when you want to make sure that the input is correct. Python starts by executing the sequence of statements in the try block. If all goes well, it skips the except block and proceeds. If an exception occurs in the try block, Python jumps to the except block.*

```
inp = input("Enter Fahrenheit Temperature:\n")

try:
    fahr = float(inp)
    cel = (fahr - 32) *(5/9)
```

```

    print("The temperature in cel: " + str(cel))
except:
    print("Please enter a number!")

```

Enter Fahrenheit Temperature:

72

The temperature in cel: 22.22222222222222

1. Rewrite your pay computation to give the employee 1.5 times the hourly rate for hours worked above 40 hours.

```

In [7]: hours = float(input("Enter Hours: "))
        rate = float(input("Enter Rate: "))

        # check if the empolyee had overtime hours, and compute pay
        if hours > 40:
            overtime_hours = hours - 40
            overtime_pay = overtime_hours * rate * 1.5
            final_pay = ((hours - overtime_hours) * rate) + overtime_pay
        else: # if the emplpoyee did not have overtime
            final_pay = (hours * rate)

        print("Pay: " + str(final_pay))

```

Enter Hours: 3

Enter Rate: 15

Pay: 45.0

2. Rewrite your pay program using try and except so that your program handles non-numeric input gracefully by printing a message and exiting the program.

```

In [8]: import sys # needed for the function sys.exit() to stope the execution

```

```

try:
    hours = float(input("Enter Hours: "))
    rate = float(input("Enter Rate: "))
except:
    print("please enter a number")
    sys.exit(1)

# check if the empolyee had overtime hours, and compute pay
if hours > 40:
    overtime_hours = hours - 40
    overtime_pay = overtime_hours * rate * 1.5
    final_pay = ((hours - overtime_hours) * rate) + overtime_pay
else: # if the emplpoyee did not have overtime

```

```
    final_pay = (hours * rate)
    print("Pay: " + str(final_pay))
```

Enter Hours: 48

Enter Rate: 15

Pay: 780.0