

```
# -Exercises1.py *- coding: utf-8 -*-
```

```
"""
```

```
Student needs:
Exercises1.py
```

We will mainly use the following window panes: IPython Console, Editor, File Explorer, and Object Inspector.  
 ### breaks up the Editor document into cells. The green triangle in the tool bar executes the entire file (after saving it), Ctrl-Enter (Command-Return on a Mac) executes only the cell that the cursor is in (but does not save).

Instructions on changing working directory in Spyder: At the top on the right you will see a path, the working directory. To its right is a yellow file folder. Click it and you can change the working directory. When you do, you can click the icon to the right of that and set that path as the IPython console's new working directory. Then all the panes: Editor, IPython Console, and File Explorer are pointed to this current working directory. In the more recent versions of Spyder, this button has been eliminated and the editor and IPython console are automatically set to the current working directory.

```
"""
```

```
"""
```

```
Quick look at arithmetic operations
```

```
+, -, *, **, /, //, %
```

These add, subtraction, multiple, exponentiate, divide, integer divide (drops fractional part), computes remainder on division for integers.

Try some examples interactively in IPython window on lower right.

```
"""
```

```
###
```

```
def hello():
    """ prints hello, world """
    print("Hello, world!")
```

```
###
```

```
def areacircle(radius):
    """ Computes the area of a circle of the given radius """
    area = 3.14*radius**2
    print("The area of a circle of radius",radius,"is", area)
```

```
###
```

```
"""
```

```
Exercise:
```

Write a function 'def areatriangle(b,h)' to compute the area of a triangle: formula is  $area = .5*b*h$ .

Output should look like:

The area of a triangle of base 3 and height 5 is 7.5

You can test your function by executing the following code:

```
"""
```

```
###
```

```
# The following will test areatriangle()
areatriangle(3,5)
areatriangle(2,20)
```

```
###
```

```
"""
```

```
Solution:
```

```
"""
```

```
###
```

```
def areatriangle(b,h):
```

```
###
"""
```

```
End solution
```

```
###
"""
```

To make a string we may use ' or ". Either works equally well. But if the string contains one, we need to use the other:

```
"""
```

```
###
```

```
name = "His name is Conan O'Brien"
```

```
cat = 'My cat is named "Butters"'
```

```
print(name)
```

```
print(cat)
```

```
###
```

```
"""
```

If you need both a ' and a " in your string, you can use the escape character \ which tells Python that the following character is to be taken as the literal character and is not a quote to delimit the string. See it in action escaping the " below:

```
"""
```

```
###
```

```
both = "My cat's name is \"Butters\""
```

```
print(both)
```

```
###
```

```
def fahrenheit_to_celsius(temp):
```

```
    """ Converts Fahrenheit temperature to Celsius.
```

```
    Formula is 5/9 of temp minus 32 """
```

```
    # Note that this line is not executed
```

```
    # end='' keeps print from starting a new line.
```

```
    newTemp = 5*(temp-32)/9
```

```
    print("The Fahrenheit temperature",temp,"is equivalent to",newTemp,end='')
    print(" degrees Celsius")
```

```
###
```

```
"""
```

```
Exercise:
```

Write a function 'def celsius\_to\_fahrenheit(temp)' to convert Celsius to Fahrenheit temperature. The formula is (9/5) times temp plus 32.

Print the output in the form:

The Celsius temperature 50.0 is equivalent to 122.0 degrees Fahrenheit.

```
"""
```

```
###
```

```
# The following will test the above function
```

```
celsius_to_fahrenheit(100)
```

```
celsius_to_fahrenheit(0)
```

```
celsius_to_fahrenheit(50.)
```

```
###
```

```
"""
```

```
Solution:
```

```
"""
```

```
###
```

```
def celsius_to_fahrenheit(temp):
```

```
###
```

```
"""
```

```
End solution
```

```
"""
```

```
###
```

```
def name():
```

```
    """ Input first and last name, combine to one string and print """
```

```
    fname = input("Enter your first name: ")
```

```
    lname = input("Enter your last name: ")
```

```
    fullname = fname + " " + lname
```

```
    print("Your name is:", fullname)
```

```
###
```

```
"""
```

```
Exercise:
```

```
Extend the name function written in class to include the city and state.
```

```
That is, ask two more questions to get the city and the state you live in.
```

```
Print where you are from on a new line. Put the customary comma between
```

```
city and state. to save time, here is the starting function.
```

```
Your run should look like the following (even if this is not the customary
```

```
way in your country):
```

```
Enter your first name: Bill
```

```
Enter your last name: Boyd
```

```
Enter the city you live in: Middletown
```

```
Enter the state you live in: CT
```

```
Your name is: Bill Boyd
```

```
You live in: Middletown, CT
```

```
"""
```

```
"""
```

```
Solution:
```

```
"""
```

```
###
```

```
def name():
```

```
    """ Input first and last name, combine to one string and print
```

```
        Also, input the city and state and print."""
```

```
    fname = input("Enter your first name: ")
```

```
    lname = input("Enter your last name: ")
```

```
    fullname = fname + " " + lname
```

```
    print("Your name is:", fullname)
```

```
###
```

```
"""
```

```
End solution
```

```
"""
```

```
###
```

```
def if_statement():
```

```
    """ Three slightly difference versions of if: if, if-else, if-elif-else"""
```

```
    x = 5
```

```
    y = 0
```

```
    z = 0
```

```
    if x > 0:
```

```
        print("x is positive")
```

```
    if y > 0:
```

```
        print("y is positive")
```

```
    else:
```

```
print("y is not positive")
```

```
# elif can be repeated as often as necessary
if z > 0:
    print("z is positive")
elif z < 0:
    print("z is negative")
else:
    print("z must be 0")
```

```
###
"""
```

Python uses '=' for assignment and '==' for testing equality. Also '!=' is used to test for non-equality. Try these examples:

```
"""
```

```
###
```

```
x = 5
y = 5
z = 6
```

```
###
"""
```

Now we try to following:

```
"""
```

```
print("x is equal to y: ", x == y)
print("x is not equal to y: ", x != y)
print("x is equal to z: ", x == z)
print("x is not equal to z: ", x != z)
```

```
###
"""
```

The following function uses an 'if' statement. Note that the indention marks the scope of the 'if', 'elif', 'else' actions.

```
"""
```

```
def area(type_, x):
    """ Computes the area of a square or circle.
        type_ must be the string "circle" or the string "square"
        We use type_ here, because type is a Python keyword. """
    if type_ == "circle":
        area = 3.14*x**2
        print(area)
    elif type_ == "square":
        area = x**2
        print(area)
    else:
        print("I don't know that one.")
```

```
###
"""
```

Exercise:

Write a function `absolutevalue(num)` that computes the absolute value of a number. You will need to use an 'if' statement. Remember if a number is less than zero then you must multiply by -1 to make it greater than zero. Give output in the form:

The absolute value of -5 is 5

```
"""
```

```
###
```

```
# Test runs
absolutevalue(5)
absolutevalue(-5)
absolutevalue(4-4)
```

```
###
"""
```

Solution:

```
"""
```

```
###
```

```
def absolutevalue(num):
```

```
###
"""
```

```
End solution
```

```
"""
```

Example: The next three examples work with the 'input' statement and point out some of the things that you might need to be aware of in using one. It also shows how to use the 'print' statement without having a new line started at the end of that statement by using an 'end' argument in it.

```
"""
```

```
###
```

```
def fahrenheit_to_celsius1():
```

```
    """ BAD. Does not check input before using it.
```

```
    Input from keyboard, which is always a string and must often be
    converted to an int or float.
```

```
    Converts Fahrenheit temp to Celsius.
```

```
    """
```

```
    temp_str = input("Enter a Fahrenheit temperature: ")
```

```
    temp = int(temp_str)
```

```
    newTemp = 5*(temp-32)/9
```

```
    print("The Fahrenheit temperature",temp,"is equivalent to ",end='')
```

```
    print(newTemp,"degrees Celsius")
```

```
###
```

```
"""
```

Test the program above by entering a temperature such as 212. Also check what happens if you simply press enter.

```
"""
```

```
###
```

```
def fahrenheit_to_celsius2():
```

```
    """ IMPROVED. Does some checking of input before using it.
```

```
    Input from keyboard, which is always a string and must often be
    converted to an int or float.
```

```
    Converts Fahrenheit temp to Celsius.
```

```
    Uses 'if' to make sure an entry was made.
```

```
    """
```

```
    temp_str = input("Enter a Fahrenheit temperature: ")
```

```
    if temp_str:
```

```
        temp = int(temp_str)
```

```
        newTemp = 5*(temp-32)/9
```

```
        print("The Fahrenheit temperature",temp,"is equivalent to ",end='')
```

```
        print(newTemp,"degrees Celsius")
```

```
###
```

```
"""
```

Test the program above by entering the temperature 212 and also by simply pressing 'Enter' or 'Return' key. Note the improvement. Now try entering 'a'.

```
"""
```

```
###
```

```
def fahrenheit_to_celsius3():
```

```
    """ MORE IMPROVED. Does even more checking of input before using it.
```

```
    Input from keyboard, which is always a string and must often be
    converted to an int or float.
```

```
    Converts Fahrenheit temp to Celsius.
```

```
    Uses if to check whether input is a number and then uses .isdigit() method
    of strings to check whether input is made of of digits.
```

```
    """
```

```
temp_str = input("Enter a Fahrenheit temperature: ")
if temp_str:
    if temp_str.isdigit():
        temp = int(temp_str)
        newTemp = 5*(temp-32)/9
        print("The Fahrenheit temperature",temp,"is equivalent to ",end='')
        print(newTemp,"degrees Celsius")
    else:
        print("You must enter a number. Bye")
```

```
###
"""
```

Test the program above by entering the temperature 212, by simply pressing 'Enter' or 'Return' key, and by entering 'a'. Note the improvement. We will leave the function at this point though further improvements could be made.

```
"""
"""
```

The following function uses integer division.

```
"""
```

```
###
```

```
def inches_to_feet1(inches):
    """ converts inches to feet and inches """
    feet = inches//12 # division by integer with fraction thrown away
    extra_inches = inches - 12*feet
    print(inches,"inches is",feet,"feet and",extra_inches,"inches")
```

```
###
```

```
"""
```

Exercise: Rewrite inches\_to\_feet1(inches) calling it inches\_to\_feet2(inches) using % to compute the inches. Recall that 19 % 5 will give 4 (the remainder). Copy and paste the original into the solution area and modify to same typing time.

```
"""
"""
```

Solution:

```
"""
```

```
###
```

```
def inches_to_feet2(inches):
```

```
###
```

```
"""
```

End solution

```
"""
```

```
"""
```

The 'while' loop. Loops are used to repeat actions and the scope of this repetition is indicated by the indentation after the 'while' statement.

```
"""
```

```
###
```

```
def cheer():
    """ Prints 2 4 6 8, who do we appreciate .... Note that everything in
    the while loop is indented. The first line not indented is the first
    line following the while loop. """
    ct = 2
    while ct <= 8:
        print(ct,end=" ") # end = " " keeps from starting a new line
        ct = ct + 2
    print() # now we'll start a new line
    print("Who do we appreciate?")
    print("COURSERA!")
```

```
###
```

```
"""
```

```
Exercise:
```

Write a function `count_down()` that starts at 10 and counts down to rocket launch. It's output should be 10 9 8 7 6 5 4 3 2 1 BLASTOFF! You can make all the numbers on the same line or different lines. Use a while loop.

```
"""
```

```
"""
```

```
Solution:
```

```
"""
```

```
###
```

```
###
```

```
"""
```

```
End solution
```

```
"""
```

```
"""
```

The 'for' loop. This loop uses an iterator to determine how many times to go through the loop. The iterator we use below is 'range(start, stop, step)'.

```
"""
```

```
###
```

```
def cheer2():
```

```
    """ Same as cheer, but uses a for loop and range()
    range uses a start number, a stop number and a step size. """
    for ct in range(2,9,2):
        print(ct,end=' ')
    print()
    print("Who do we appreciate?")
    print("COURSERA!")
```

```
###
```

```
"""
```

```
Exercise:
```

Write a function `countdown1()` that starts at 10 and counts down to rocket launch. It's output should be 10 9 8 7 6 5 4 3 2 1 BLASTOFF! You can make all the numbers on the same line or different lines. Use a 'for' loop and `range()`. `range` has a start and a stop and a step that MAY BE NEGATIVE.

```
"""
```

```
"""
```

```
Solution:
```

```
"""
```

```
###
```

```
def countdown1():
```

```
###
```

```
"""
```

```
End solution
```

```
"""
```

```
###
```

```
"""
```

Some of our exercises involve finding and fixing errors in code.

Here is an example. Can you see the errors (there are two)? Note that the editor is pointing out one line with troubles.

You can find the error by reading the example carefully, or trying to make it work by using Shift-Enter to insert the function into IPython and reading

what error it gives or trying to run the function.

"""

###

def favorite():

my\_toy = input("What is my favorite toy? ")

print("Your favorite toy is", my\_toy)

###

"""

My solution:

"""

###

"""

end solution

"""