

Solutions to additional exercises

These exercises were set at the end of the second afternoon of the “Python: Introduction for Absolute Beginners” course. Here are some model solutions.

1. You have the following collection of value pairs you want to store in a variable of some sort. They are the names and boiling points of some elements. Store them in a variable. Then work out how, given a temperature, you would display only those elements whose boiling point is within 10 degrees of that temperature.

Element	Boiling Point (°C)
helium	-268.6
oxygen	-183
argon	-186
fluorine	-188.14
bromine	58.78
mercury	356.58
aluminium	2467

Solution:

```
# We have a collection of value *pairs*, so we probably want a  
# to put them in a dictionary.
```

```
boiling_points = { "helium": -268.6,  
                  "oxygen": -183.0,  
                  "argon": -186.0,  
                  "fluorine": -188.14,  
                  "bromine": 58.78,  
                  "mercury": 356.58,  
                  "aluminium": 2467.0  
                }
```

```
# Temperature of which we want boiling points within 10 degrees.  
temperature = -185.0
```

```
# Loop through dictionary and see whether boiling point of  
# of substance is within 10 degrees of the temperature. If so,  
# print the name of the substance.
```

```
for substance in boiling_points:  
    if abs(temperature - boiling_points[substance]) <= 10.0:  
        print substance  
del substance
```

2. For x in the range 1 to 25, print out $x, x^2, \frac{5}{x}$. You should display the values in descending order of x , i.e. from 25 to 1, and for each value of x , the three values you have to display should be on a single line separated by tabs.

Solution:

```
# Print out the results of simple mathematical operations.
# Note that we have to make sure we are doing floating point
# division for the last operation (5/x).
for x in range(25, 0, -1):
    print x, '\t', x * x, '\t', 5.0 / x
del x
```

3. Turn your answers for exercises 1 and 2 into functions.

Solution:

```
def within_10_of_x(dictionary, x):

    # Start with an empty list
    results = []

    # Loop through the dictionary and see whether, for each
    # key, the corresponding value is within 10 of x (the
    # argument we were passed). If so, add that key to
    # our list.
    for item in dictionary:
        if abs(x - dictionary[item]) <= 10.0:
            results.append(item)

    # Return list of keys whose values are within 10 of x.
    return results

def display_some_calculations():
    # Print out the results of simple mathematical operations.
    # Note that we have to make sure we are doing floating
    # point division for the last operation (5/x).
    for x in range(25, 0, -1):
        print x, '\t', x * x, '\t', 5.0 / x
```

4. Write a function to determine whether two floating-point numbers are within the machine epsilon of each other.

Solution:

```
def close_as_epsilon(number1, number2):
    # Calculate estimate of machine epsilon.
    epsilon = 1.0
    while (1.0 + epsilon) > 1.0:
        epsilon = epsilon / 2.0
    epsilon = epsilon * 2.0

    # If numbers are within machine epsilon of each other,
    # return True, otherwise return False.
    if abs(number1 - number2) <= epsilon:
        return True
    else:
        return False
```