# Input and Output and calculations


Lillebaelt Academy of
Professional Higher Education

Author
Martin Grønholdt
mart80c7@edu.eal.dk

**Sunday 30 October 2016**

# Table of Contents

Introduction.....3
1 Personal Information.....4
2 Sales Prediction.....5
3 Land Calculation.....6
4 Total Purchase.....7
5 Distance Traveled.....9
Conclusion.....10

# Introduction

These programs shows how to use basic python input and output functions.

The basic program structucture start with a function, that does the main processing of the program

```
def main():
      …
```

Then comes some python code to execute the main() function when called as a script.

```
if __name__ == '__main__':
      main()
```

# 1 Personal Information

This program uses print, to print my personal information.

## prog1.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
'''
Name: prog1
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (30/10-2016)

Program that prints some information about me.
'''

def main():
	# Print my name.
	print('Martin Bo Kristensen Grønholdt')
	# Print my address.
	print('Jacob Hansensvej 153, 1')
	# Print the zip code and city.
	print('5260 Odense S')
	# Tell that I was a "sproglig student" back in the days
	print('Language student')

#Run this when invoked directly
if __name__ == '__main__':
	main()
```

## Result

```
oblivion@server-martin:~/Dokumenter/Skole/It-et/2016/programming/Assignent A1$
python prog1.py

Martin Bo Kristensen Grønholdt
Jacob Hansensvej 153, 1
5260 Odense S
Language student
```

*Output of the program when run from the command line.*

# 2 Sales Prediction

This program uses input to wait for user input. No error checking is performed, the program will throw an exception on malformed input.

## prog2.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
'''
Name: prog2
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (30/10-2016)

Program to print the projected profit of a sale.
'''
def main():
      # Get the total sales from via user input.
      total_sale = input('What is the projected total sale? ')
      # Calculate the total profit, by first converting the input to
      # an floating point value, then calculate the 23%, convert this to a
      # string and add it to the initial message string.
      print('Projected profit (23%): ' + str(0.23 * float(total_sale)))

#Run this when invoked directly
if __name__ == '__main__':
      main()
```

## Result

```
oblivion@server-martin:~/Dokumenter/Skole/It-et/2016/programming/Assignent A1$
python prog2.py

What is the projected total sale? 123
Projected profit (23%): 28.29
```

*Output of the program when run from the command line.*

```
oblivion@server-martin:~/Dokumenter/Skole/It-et/2016/programming/Assignent A1$
python prog2.py

What is the projected total sale? dfr
Traceback (most recent call last):
 File "prog2.py", line 23, in <module>
   main()
 File "prog2.py", line 15, in main
   total_sale = input('What is the projected total sale? ')
 File "<string>", line 1, in <module>
NameError: name 'dfr' is not defined
```

*Output of the program when given malformed input.*

# 3 Land Calculation

This program is basicly the same as the last, except for a different formula. No error checking is performed, the program will throw an exception on malformed input.

## prog3.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
'''
Name: prog3
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (30/10-2016)

Program to convert square feet to acres.
'''
def main():
      # Get the area in square feet
      square_feet = input('Input the total square feet? ')
      # Calculate the acres, by first converting the input to
      # a floating point value, divide by 43560, convert this to a
      # string and add it to the initial message string.
      print('Acres: ' + str(float(square_feet) / 43560))

#Run this when invoked directly
if __name__ == '__main__':
      main()
```

## Result

```
oblivion@server-martin:~/Dokumenter/Skole/It-et/2016/programming/Assignent A1$
python prog3.py

Input the total square feet? 123
Acres: 0.00282369146006
```

*Output of the program when run from the command line.*

```
oblivion@server-martin:~/Dokumenter/Skole/It-et/2016/programming/Assignent A1$
python prog3.py

Input the total square feet? dfg
Traceback (most recent call last):
 File "prog3.py", line 23, in <module>
   main()
 File "prog3.py", line 15, in main
   square_feet = input('Input the total square feet? ')
 File "<string>", line 1, in <module>
NameError: name 'dfg' is not defined
```

*Output of the program when given malformed input.*

# 4 Total Purchase

This program expands on user input, by using a while loop to keep asking for input, until a certain condition is met. In this instance, that the input is larger than zero. No error checking is performed, the program will throw an exception on malformed input.

## prog4.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
'''
Name: prog4
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (30/10-2016)

Calculate the total price of a number of items, as well as the taxes
paid.
'''
def main():
      #Start with and emtpy total
      total_price = 0
      total_items = 0

      #Get the price
      price = input('Input item price (stop adding items by ' +
                        'inputting less than 1):')
      #Keep asking for the price of an item until the user inputs 0
      while price > 0:
            #Add the price of the new item
            total_price += float(price)
            #Keep count of the number of items added
            total_items += 1
            #Ask for the price of antoher item
            price = input('Input item price (stop adding items by ' +
                          'inputting less than 1):')

      #Print the subtotal
      print('\nThe subtotal price of your ' + str(total_items) +
            ' items are: ' + str(float(total_price)))
      #Print the tax
      print('Sales tax (6%) amounts to: ' +
            str(float(total_price) * 0.06))
      #Print the total price
      print('The total price for you items including tax is: ' +
            str((total_price + float(total_price) * 0.06)))


#Run this when invoked directly
if __name__ == '__main__':
      main()
```

## Result

```
oblivion@server-martin:~/Dokumenter/Skole/It-et/2016/programming/Assignent A1$
python prog4.py
```

```
Input item price (stop adding items by inputting less than 1):12.5
Input item price (stop adding items by inputting less than 1):.5
Input item price (stop adding items by inputting less than 1):2
Input item price (stop adding items by inputting less than 1):6
Input item price (stop adding items by inputting less than 1):0

The subtotal price of your 4 items are: 21.0
Sales tax (6%) amounts to: 1.26
The total price for you items including tax is: 22.26
```

*Output of the program when run from the command line.*

```
oblivion@server-martin:~/Dokumenter/Skole/It-et/2016/programming/Assignent A1$
python prog4.py

Input item price (stop adding items by inputting less than 1):1
Input item price (stop adding items by inputting less than 1):3
Input item price (stop adding items by inputting less than 1):f
Traceback (most recent call last):
 File "prog4.py", line 45, in <module>
   main()
 File "prog4.py", line 30, in main
   'inputting less than 1):')
 File "<string>", line 1, in <module>
NameError: name 'f' is not defined
```

*Output of the program when given malformed input.*

# 5 Distance Traveled

This program uses another loop construct, the *for* loop, to run through a list of values and execute the lines in the loop. In this case the for loop runs through the hours which the car has travelled, and outputs the distance at each loop.

## prog5.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
'''
Name: prog5
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (30/10-2016)

Program to print the distance traveled at a speed of 60 mph in 5, 8,
and 12 hours.
'''
def main():
      #Speed in mph
      mph = 60

      #Print a message to tell what is calulated
      print('When traveling at ' + str(mph) + ' miles per hour:\n');

      #Loop through the values 5, 8, 12 which is the hours traveled.
      for hours in [5, 8, 12]:
            #Calculate and print the distance traveled by the formula
            #distance = hours * speed.
            print('The distance traveled after ' + str(hours) + ' is ' +
                  str(hours * mph) + ' miles.')

#Run this when invoked directly
if __name__ == '__main__':
      main()
```

## Result

```
oblivion@server-martin:~/Dokumenter/Skole/It-et/2016/programming/Assignent A1$
python prog5.py

When traveling at 60 miles per hour:

The distance traveled after 5 is 300 miles.
The distance traveled after 8 is 480 miles.
The distance traveled after 12 is 720 miles.
```

*Output of the program when run from the command line.*

# Conclusion

These examples hopefully shows what wonderful language python is. It is a language that makes it easy to do a fast prototype of an idea, just to see if the concept works. Python is very versatile, and is used for both small script and professional open and closed source products.