



## Background of this document + what it is

I put this information together in order to have a supplement to the school program in computer science for my son.

This document contains a guideline for teachers.

The Python examples should help non programmers to understand some Python programming language basics. This document is a supplement to the [Programming introduction](#) document, it describes not the whole language. The tutorials and references you can find on the Internet will give you all information about Python.

## Prerequisites

- You have to know the "Programming" section mentioned in the [Computer Crash Course](#)
- You have to understand the [Programming language basic elements](#)
- It's helpful to understand the [Programming introduction](#) document
- You can install and use a programming environment on a PC

## Computer Crash Course (CCC)

The Computer Crash Course contains a guideline for teachers in basic computer knowledge. It can also be used as an introduction for beginners. You have to follow the mentioned links and read additional information concerning the keywords in the CCC document, otherwise you won't understand this course.

## Programming language basic elements

This course gives an general introduction to the basic elements of a high level programming language. Warning : I use ONLY pseudo code in this document !

## Programming introduction

The Programming introduction is a very short "jump" into programming at a basic level. This document will give you an idea of writing a shell-script and programming in a high level programming language.

## Table of Contents

<b>Input, output</b> .....	4
<b>While</b> (control structure).....	5
<b>If</b> (control structure).....	7
<b>Functions</b> .....	10
<b>Lists</b> (data structure).....	11
<b>For</b> (control structure).....	18
<b>Dictionaries</b> (data structure).....	21
<b>Importing</b> (modules).....	24
<b>File I/O</b> .....	26
<b>Error handling</b> .....	27

## Input, output

### Code

The code will calculate the square root and the sum of a given number.

```
x = input ("Please enter a number: ")
print "The square of that number is", x*x, "the sum of that number is", x+x
```

---

### Code

The code will print the entered string and multiplying the two integers.

```
string1 = raw_input('String 1: ')
string2 = raw_input('String 2: ')
int1 = input('Integer 1: ')
int2 = input('Integer 2: ')
print "string you typed in : ", string1, " ", string2
print "multiplying the integers :", int1,"x",int2,"=",int1 * int2
```

### Explanations

Anything can be entered by the function "raw\_input".

---

## While (control structure)

### Code

The code will output and count down from a entered number.

```
x = input("enter a number from 1 to 10 : ")
while x >= 0:
    print "count down x is:  ",x
    x = x-1 # decrement
# end while
```

---

### Code

The code will verify if you typed in the correct user name (Marc) and password (unicorn).

```
frei = "c"
while frei != "exit":
    user = "a"
    password = "b"
    while user != "Marc":
        user = raw_input("User Name: ")
    # end while
    print "the user name is ok"
    while password != "unicorn":
        password = raw_input("Password: ")
    # end while
    print "the password is ok"
    print
    print "welcome ", user
    print
    frei = raw_input("type exit to exit OR CR to continue : ")
# end while
```

---

## Code

You can set a user name and password, then the code will verify if you typed in the correct user name and password.

```
key = " "  
while key != 'exit':  
    name = raw_input('Set name: ')  
    password = raw_input('Set password: ')  
    nameguess=passwordguess=key=""  
    while (nameguess != name) or (passwordguess != password):  
        nameguess = raw_input('Name? ')  
        passwordguess = raw_input('Password? ')  
    # end while  
    print "Welcome, ", name,  
    print  
    key = raw_input("Type exit to quit, or CR to continue: ")  
    while key == 'exit':  
        print "***** terminated *****"  
        break  
    # end while  
# end while
```

---

## If (control structure)

### Code

The code will output the sum of two numbers, if the sum is higher than 100, the output text will be different.

```
a = input ("type the first number: ")
b = input ("type the second number: ")
c = a + b
if c > 100:
    print ("the sum is higher than 100, it is : "), c
else:
    print ("the sum is"), c
# end if
```

---

### Code

The code will verify if you typed in a number < than 10.

```
a = input ("type a number < 10 : ")
if a < 10:
    print a, "is a valid number"
elif a > 20:
    print "the number you typed in :",a, "is definitely out of range"
elif a >= 10:
    print a, "is higher or equal than 10"
# end if
```

---

## Code

The code will output the float format of an integer or float input.

```
x = 1 # initialize
while x == 1:
    print
    y = float(raw_input("type in a number "))
    print
    print "your entry in float format was : ", y
    print
    x = input ("type 1 to continue, 0 for exit : ")
    if x == 0:
        print
        print 'program terminated'
        print
    # end if
# end while
```

---

## Code

The code will help you guess a number in indicating if your guess is below or above the "secret number". The program will terminate after 3 false guesses.

```
number = 78
guess = 0
count = 0
exit = 1

while exit == 1:
    guess = input ("Guess a number: ")
    if guess > number:
        print "Too high"
    elif guess < number:
        print "Too low"
    elif guess == 78:
        print "Just right"
        count = 0 # in case if the third guess was right !
        exit = 0
    # end if
    count = count + 1
    if count == 3:
        print ("try again later")
        exit = 0
    # end if
# end while
```

---

## Code

The code will verify if you typed in the correct user name (Phil) and password (McChicken).

```
login = "john"
password = "tucker"
logged=2
while logged != 0:
    while login != "Phil":
        login = raw_input("Login : ")
    # end while
    while password != "McChicken":
        password = raw_input("Password: ")
    # end while
    logged = 1

print "Welcome!", login
print "To leave type exit "

while logged == 1:
    leave = raw_input(">> ")
# end while
    if leave == "exit":
        logged = 0
    # end if
# end while
print "Goodbye!!"
```

---

## Functions

### Introduction

Functions have a special type of variable called local variables. These variables only exist while the function is running. When a local variable has the same name as another variable (such as a global variable), the local variable hides the other.

### Code

This code will calculate the square root of your input.

```
def square(x):  
    return x*x
```

```
x = input ("Please enter a number: ")  
print "the square root is: ",square(x)
```

---

## Code

This code will print a menu, convert from Celsius to Fahrenheit et vice versa.

```
def print_options():
    print "Menu:"
    print " 'p' print menu"
    print " 'c' convert from Celsius"
    print " 'f' convert from Fahrenheit"
    print " 'q' quit the program"

def celsius_to_fahrenheit(c_temp):
    return 9.0 / 5.0 * c_temp + 32

def fahrenheit_to_celsius(f_temp):
    return (f_temp - 32.0) * 5.0 / 9.0

choice = "p"
while choice != "q":
    if choice == "c":
        temp = input("Celsius temperature: ")
        print "Fahrenheit:", celsius_to_fahrenheit(temp)
    elif choice == "f":
        temp = input("Fahrenheit temperature: ")
        print "Celsius:", fahrenheit_to_celsius(temp)
    elif choice != "q":
        print_options()
    # end if
    choice = raw_input("choose: ")
# end while
print ("program terminated .....")
```

---

## Lists (data structure)

### Introduction

Lists are also called one-dimensional arrays.

Lists are written with brackets and can be nested.

Lists can hold dictionaries.

You can access the elements of a list separately or in groups, through **indexing** and **slicing**.

### - Indexing

### Code

This code will output the month as text corresponding to the number of the month as input.

```
which = input("What month (1-12)? ")

months = ['January', 'February', 'March', 'April', 'May', 'June', 'July',
          'August', 'September', 'October', 'November', 'December']

if 1 <= which <= 12:
    print "The month is", months[which - 1]
else:
    print ("use a valid month (1 - 12)")
# end if
```

---

## Codes

The following codes show list manipulations :

```
test_list = ["Cleese", "John",100, "test", 22, 46, "and", "wood"]
print test_list[1], test_list[0]          # shows element 1 and 2
```

---

```
test_list[0] = "Smith"                   # overwrite "Cleese" by "Smith"
print test_list[1], test_list[0]
```

---

```
print test_list
test_list.append("everything")           # adds "everything"
print test_list
```

---

```
print "number of elements in the list ", len(test_list)      # shows the length of the list
```

---

```
print "the position of 100 is ", test_list.index(100)        # shows the position of a list content
```

---

The following code will output every element of the list

```
c = 0
while c < len(test_list):
    print "List element ", c, ": ", test_list[c]
    c = c + 1
# end while
```

---

```
del test_list[1]          # delete the element "John"
print test_list
```

---

```
if "wood" in test_list:  # check if "wood" is found in the list
    print "'wood' was found in list"
# end if
```

---

**Python examples** by [Marc Oscar Schwager](#)

```
if "Kingkong" not in test_list:           # check if "Kingkong" is (not) found in the list
    print "'Kingkong' was not found in list"
# end if
```

---

```
test_list.sort()                         # shows the sorted list
print "The sorted list is", test_list
```

---

## Code

This code allows to add, remove and change list elements.

```
menu_item = 0
namelist = []          # defines an empty list

while menu_item != 9:
    print "-----"
    print "1. Print the list"
    print "2. Add a name to the list"
    print "3. Remove a name from the list"
    print "4. Change an item in the list"
    print "9. Quit"
    menu_item = input("Choose an item from the menu: ")

    if menu_item == 1:
        current = 0
        if len(namelist) > 0:
            while current < len(namelist):
                print "element ", current, ".", namelist[current]
                current = current + 1
            # end while
        else:
            print "List is empty"
        # end if "len"

    elif menu_item == 2:
        name = raw_input("Type in a name to add: ")
        namelist.append(name)

    elif menu_item == 3:
        del_name = raw_input("What name would you like to remove: ")
        if del_name in namelist:                # remark 1
            item_number = namelist.index(del_name)
            del namelist[item_number]          # remark 2
        else:
            print del_name, "was not found"
        # end if "del_name"

    elif menu_item == 4:
        old_name = raw_input("What name would you like to change: ")
        if old_name in namelist:
```

**Python examples** by [Marc Oscar Schwager](#)

```
    item_number = namelist.index(old_name)
    new_name = raw_input("What is the new name: ")
    namelist[item_number] = new_name
else:
    print old_name, "was not found"
# end if "old_name"
```

```
    # end if "menu item"
# end while
print "Goodbye"
```

**# remark 1**

```
    # the actual code removes only the first occurrence of the name
```

```
    # the following code will delete all occurrences of the name
    # while del_name in namelist:
    # item_number = namelist.index(del_name)
    # del namelist[item_number]
```

**# remark 2**

```
    # alternative code for the same task
    # namelist.remove(del_name)
```

---

## Code

This code is a simple question answer "game".

```
def get_questions():
    return ["What color is the daytime sky on a clear day? ", "blue"],
           ["What is the answer to life, the universe and everything? ", "42"],
           ["What is a three letter word for mouse trap? ", "cat"],
           ["What noise does a truly advanced machine make? ", "ping"]]

def check_question(question_and_answer):
    question = question_and_answer[0]
    answer = question_and_answer[1]
    given_answer = raw_input(question)
    if answer == given_answer:
        print "Correct"
        return True
    else:
        print "Incorrect, correct was:", answer
        return False
    # end if

def run_test(questions):
    if len(questions) == 0:
        print "No questions were given."
        return
    # end if
    index = 0
    right = 0
    while index < len(questions):
        if check_question(questions[index]):
            right = right + 1
        # end if
        index = index + 1
    # end while
    print "You got", right, "answer right of", len(questions)

def t_list (g_list):
    index = 0
    while index < len(g_list):
        print g_list [index]
        index = index + 1
    # end while
```

## Python examples by [Marc Oscar Schwager](#)

```
menu_item = 0
while menu_item != 9:
    print "-----"
    print "1. Run the test"
    print "2. View the list of questions + answers"
    print "9. Quit"
    print "-----"
    print " "
    menu_item = input("Choose an item from the menu: ")
    if menu_item == 1:
        run_test(get_questions())
    elif menu_item == 2:
        print ("questions + answers are : ")
        t_list(get_questions())
    # end if
# end while
print "Goodbye"
```

---

### - Slicing

#### Introduction

Slicing is almost like indexing, except that you indicate both the start and stop index of the result, with a colon (":") separating them. "list[:3]" means every element from the beginning of the list up to element 3, non-inclusive. "list[-3]" is the third element from the end of the list.

#### Code

This code shows the output of parts of a list using slicing.

```
x = ["alpha", "beta", "gamma", "delta", "spam", "eggs", "and", "end"]
print x[5:8] # prints "eggs", "and", "end"
print x[-8] # prints "alpha"
```

---

## For (control structure)

### Introduction

Everything that can be done with For loops can also be done with While loops!

But For loops gives an easy way to go through all the elements in a List or to do something a certain number of times.

### Code

This code prints out the values from 1 to 10 inclusive.

```
for count in range(1, 11):  
    print count  
# end for
```

---

### Code

This code prints out the values from 0 to 99 inclusive.

```
for x in range(100):  
    print x  
# end for
```

---

## Code

This code asks for the name until you type in the right one, but it stops after 3 wrong guesses.

```
name = "Marc"
i = 0
for i in range (3):
    guess_name = raw_input ("Guess my name: ")
    if guess_name == name:
        print ("very good!!")
        break # breaks the if and for
    # end if
# end for
if guess_name != name:
    print ("the guess was to heavy for you")
# end if
```

---

## Code

This code prints out 1 - 2 - 4 - 8 - 10.

```
for i in [1,2,4,8,10]:
    print "This is the iteration number", i
# end for
```

---

## Code

This code calculates and prints out the sum of 2+4+6+8.

```
list = [2, 4, 6, 8]
sum = 0
for num in list:
    sum = sum + num
# end for
print "The sum is:", sum
```

---

## Code

This code loops through the list and prints out all elements.

```
demolist = ['life', 42, 'the universe', 6, 'and', 7, 'everything']
for item in demolist:
    print "The current item is:",
    print item
# end for
```

---

## Code

This code loops through a list and outputs the data user friendly (human readable format).

```
model_list = [('BMW Z4', 8.7), ('Audi TT', 8.6), ('Mercedes CLK350', 8.8)]
for model in model_list:
    print 'The', model[0], 'acceleration is', model[1]
# end for
```

---

## Code

This code detects duplicate content in the list.

```
list = [2, 5, 7, 8, 9, 7, 10, 2]
list.sort()          # sort in order to prepare the comparison
prev = list[0]       # stores the first element in "prev"
del list[0]          # delete the first element to avoid false detection of duplicate content
for item in list:
    if prev == item:
        print "Duplicate content found on position : ", list.index(prev)+1 , "value : ", prev
    # end if
    prev = item
# end for
```

---

## Code

This code calculates the gallon price depending on the mileage.

```
while True: # as the condition for this while loop is always true....
    print # makes an empty line
    mpg = float(raw_input("What is the mileage? Enter 0 to end: "))
    print
    if mpg == 0:
        print 'program terminated'
        print
        break # the break instruction breaks the loop (a while loop or a for loop) , program exit
    # end if
    gallon_price_list = range(220, 250, 5)
    for gallon_price in gallon_price_list:
        cost = gallon_price / mpg
        print 'gallon price in US$', gallon_price / 100.0,
        print "cost in US$", cost, "per hundred miles"
    # end for
# end while
```

---

## Dictionaries (data structure)

### Introduction

Dictionaries are also called associative arrays.

The dictionary contents are not ordered.

Dictionaries have keys and values. Keys can be strings or numbers. Keys point to values.

Values can be any type of variable :

- Dictionaries can hold other dictionaries
- Dictionaries can hold lists

### Code

This code shows basic dictionary manipulations.

```
person = { 'first name': "Robin", 'last name': "Hood", 'occupation': "Scoundrel" }  
print person  
person['last name'] = "of Locksley" # change the last name  
print person  
print "his occupation: ",person ['occupation'] # print only an element
```

---

## Code

This code is a simple telephone diary example using a dictionary. You can basically add, remove and look-up a phone number.

```
def print_menu():
    print '1. Print phone numbers'
    print '2. Add a phone number'
    print '3. Remove a phone number'
    print '4. Look-up a phone number'
    print '5. Quit'

numbers = {} # defines an empty dictionary
menu_choice = 0

while menu_choice != 5:
    print ("-----")
    print_menu()
    print ("-----")
    menu_choice = input("Choose from 1 to 5: ")
    print
    if menu_choice == 1:
        if not numbers:
            print "No phone number available, add one or more"
        else:
            print "Phone numbers:"
            for x in numbers.keys():
                print "Name: ", x, "\tNumber:", numbers[x]
            print

    elif menu_choice == 2:
        print "Add name and number"
        name = raw_input("Name: ")
        phone = raw_input("Number: ")
        numbers[name] = phone

    elif menu_choice == 3:
        print "Remove name and number"
        name = raw_input("Name: ")
        if numbers.has_key(name):
            del numbers[name]
        else:
            print name, "was not found"
```

## Python examples by [Marc Oscar Schwager](#)

```
elif menu_choice == 4:
    print "Look-up number"
    name = raw_input("Name: ")
    if numbers.has_key(name):
        print "The number is", numbers[name]
    else:
        print name, "was not found"
# end if

# end while
print ("Goodbye")
```

---

## Importing (modules)

### Code

This code prints a calendar of your choice.

```
import calendar
year = input("Type in the year number: ")
calendar.prcal(year)
```

### Explanations

External libraries are used with the import [libname] keyword.

---

### Code

This code prints a calendar of your choice.

```
from calendar import prcal
year = input("Type in the year number: ")
prcal(year)
```

### Explanations

You can use from [libname] import [funcname] for individual functions.

---

### Code

This code prints a random number.

```
import random
randomint = random.randint(1, 100)
print randomint
```

---

## Code

This code lets you guess a random number.

```
import random
number = random.randint(1, 100)
guess = 0
print "maybe you would try this one ? :", number # only to help that you don't have to guess too
many times !
while guess != number:
    guess = input ("Guess a number: ")
    if guess > number:
        print "Too high"
    elif guess < number:
        print "Too low"
# end while
print "Just right"
```

---

## File I/O

### Code

This code does basic file I/O.

```
out_file = open("test.txt", "w")    # Write a file
out_file.write("This Text will be in the file test.txt\nLook at it and see!")
out_file.close()
```

```
in_file = open("test.txt", "r")     # Read a file
text = in_file.read()
in_file.close()
```

```
print text                          # show file content on screen
```

---

## Code

This code prints blocks of 5 lines from a text document.

```
test_file = open('test.txt', 'rb')
i = 0 # set the line counter to zero
print

for line in test_file: # repeat the for loop until all lines are printed
    # loop (for 5 lines)
    i = i + 1 # increment i

    print line # print line from file

    if i == 5: # end loop (for 5 lines)

        i = 0 # then set the counter to zero
        print
        raw_input("Hit enter to continue to see the next 5 lines : ") # then wait for input
        print
    # end if
# end for

print
print "***** this is the end of the document *****"
print
test_file.close()
```

---

## Error handling

### Code

This code checks if you entered a number (integer) or not.

```
number = 1
while number != -1:
    try:
        number = int(raw_input("Enter a number or type -1 to exit : "))
        if number > -1:
            print "You entered:", number
        else:
            print "goodbye"
    except ValueError:
        print "This was not a number."
# end while
```

---