Introduction to Programming in Python

Python is basically a scripting language - it needs an interpreter to run. However, since it has gained a lot of attention lately (plus, my daughter has to learn it and I obviously HAVE to help her), I will start some notes on that here :-p

Since I am from a C-programming background, I will most of the time compare the two languages.

Development Environment

For this, we only need a code editor to write our code and python interpreter installed. For those using Linux, most distributions have this installed by default. For Windows users, you will have to get the installer from here.

I prefer simple code editors, so I recommend Geany. If you want an IDE, pycharm (a commercial software - but has a community edition version available) is not bad. Its interface is VERY similar to Android Studio :-p

There are currently two python versions that is being used python2.7 and python3.x - somehow, python3 brings a lot of changes that render scripts/codes written for python2.7 no longer usable. So, until that is fixed, we will have to choose. Slackware Linux (MY Linux distribution of choice) installs python2.7 by defaut - so, that is what I will be using.

Writing Programs in Python

A few things to note:

1. python is like assembly, a line holds a single statement/command (so, no need for ';' like in C)
2. a block of codes is presented using the same indentation (so, no need for '{' and '}' pairs like in C)
3. it supports OOP - classes (inheritance,polymorphism,abstraction), exception handling, etc.

to be continued...

Coding: Basic User Interface

learn1.py

```
# howto: print text
# howto: get input from user
# howto: display variable
```

```
from datetime import date
import sys

def get_input(ask):
  if sys.version_info[0] < 3 or sys.version_info[1] < 4:
    temp = raw_input(ask)
  else:
    temp = input(ask)
  return temp

def get_year():
  year = date.today().strftime("%Y")
  return int(year)

print("This is "+"awesome!")
name = get_input("Please enter your name: ")
print("Hello, " + name + "!")
year = get_input("Please enter the year you were born: ")
print("You were born in year " + year + "?" )
age1 = get_year()-int(year)
print("That means you are " + str(age1) + " years old!")
```

Exception Handling

learn1_exception.py

```
# howto: detect input error (using exception)

from datetime import date
import sys

def get_input(ask):
  if sys.version_info[0] < 3 or sys.version_info[1] < 4:
    temp = raw_input(ask)
  else:
    temp = input(ask)
  return temp

def get_year():
  year = date.today().strftime("%Y")
  return int(year)

print("This is "+"awesome!")
name = get_input("Please enter your name: ")
print("Hello, " + name + "!")
year = get_input("Please enter the year you were born: ")
try :
```

```
  test = int(year)
  print("You were born in year " + year + "?" )
  age1 = get_year() - test
  print("That means you are " + str(age1) + " years old!")
except ValueError :
  print("Is '" + year + "' a valid year?")
```

Output Control

learn1_no_newline.py

```
# howto: print text with no newline
# howto: delay execution

# need this to enable python3 print feature in python2
from __future__ import print_function
import sys
import time

print("This... ",end='')
sys.stdout.flush()
time.sleep(1)
print("is... ",end='')
sys.stdout.flush()
time.sleep(1)
print("legend (wait for it)... ",end='')
sys.stdout.flush()
time.sleep(1)
print("ary!")
```

Coding: Code Selection

Branch statement (if-else)

learn2.py

```
# howto: branch statement (selection)

import sys

def get_input(ask):
  if sys.version_info[0] < 3 or sys.version_info[1] < 4:
    temp = raw_input(ask)
```

```
  else:
    temp = input(ask)
  return temp

mark = get_input("Enter your mark: ")
mark = int(mark)

if mark > 100 or mark < 0 :
  print("That is out of valid range!")
elif mark >= 80 :
  print("Grade : A")
elif mark >= 75 :
  print("Grade : A-")
elif mark >= 70 :
  print("Grade : B+")
elif mark >= 65 :
  print("Grade : B")
elif mark >= 60 :
  print("Grade : B-")
elif mark >= 55 :
  print("Grade : C+")
elif mark >= 50 :
  print("Grade : C")
elif mark >= 45 :
  print("Grade : C-")
elif mark >= 40 :
  print("Grade : D+")
elif mark >= 35 :
  print("Grade : D")
elif mark >= 30 :
  print("Grade : D-")
else:
  print("Grade : F")
```

Coding: Code Iteration

Iterative code (loops)

learn3.py

```
loop = 0
while loop < 100:
  loop = loop + 1
  if loop==9:
    break
  elif loop==4:
    continue
```

```
  print(loop)

print("-- Range:")
for loop in range(3):
  print(loop)

print("-- Spell:")
for that in "pisang goreng":
  if that==' ':
    continue
  print(that)
```

Coding: Command-line parameters

learnX_read_args.py

```
# howto: get command-line text

import sys

print('Number of arguments:'+str(len(sys.argv))+'arguments.')
print('Argument String:'+str(sys.argv));

step=0
for that in sys.argv:
  step = step + 1
  print("#"+str(step)+": "+that)
```

From:
https://azman.my1matrix.cc/ - **Azman @MY1**

Permanent link:
https://azman.my1matrix.cc/doku.php?id=learn:programming_python

Last update: **2026/04/28 02:17**