About the Tutorial

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985 – 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is named after a TV Show called ‘Monty Python’s Flying Circus’ and not after Python-the-snake.

Python 3.0 was released in 2008. Although this version is supposed to be backward incompatibles, later on many of its important features have been backported to be compatible with the version 2.7. This tutorial gives enough understanding on Python 3 version programming language. Please refer to this link for our Python 2 tutorial.

Audience

This tutorial is designed for software programmers who want to upgrade their Python skills to Python 3. This tutorial can also be used to learn Python programming language from scratch.

Prerequisites

You should have a basic understanding of Computer Programming terminologies. A basic understanding of any of the programming languages is a plus.

Execute Python Programs

For most of the examples given in this tutorial you will find Try it option, so just make use of it and enjoy your learning.

Try the following example using Try it option available at the top right corner of the below sample code box –

```python
#!/usr/bin/python3

print ("Hello, Python!"
```

Copyright & Disclaimer

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com
# Table of Contents

About the Tutorial ........................................................................................................... i
Audience .......................................................................................................................... i
Prerequisites .................................................................................................................... i
Execute Python Programs ............................................................................................... i
Copyright & Disclaimer ................................................................................................. i
Table of Contents ............................................................................................................ ii

**PYTHON 3 – BASIC TUTORIAL** ................................................................. 1

1. **Python 3 – What is New?** ................................................................................. 2
   - The __future__ module .......................................................................................... 2
   - The print Function ............................................................................................... 2
   - Reading Input from Keyboard ............................................................................ 2
   - Integer Division .................................................................................................. 3
   - Unicode Representation ..................................................................................... 3
   - xrange() Function Removed ............................................................................ 4
   - raise exceprion ................................................................................................. 4
   - Arguments in Exceptions ................................................................................... 4
   - next() Function and .next() Method ................................................................. 4
   - 2to3 Utility ....................................................................................................... 4

2. **Python 3 – Overview** ...................................................................................... 6
   - History of Python ............................................................................................... 6
   - Python Features ................................................................................................. 7

3. **Python 3 – Environment Setup** ..................................................................... 8
   - Local Environment Setup ................................................................................... 8
   - Getting Python ................................................................................................... 8
   - Setting up PATH ................................................................................................ 9
   - Setting Path at Unix/Linux ............................................................................... 10
   - Setting Path at Windows ................................................................................. 10
   - Python Environment Variables ......................................................................... 10
   - Running Python ............................................................................................... 11

4. **Python 3 – Basic Syntax** ................................................................................ 13
   - First Python Program ........................................................................................ 13
   - Python Identifiers .............................................................................................. 14
   - Reserved Words ................................................................................................ 15
   - Lines and Indentation ........................................................................................ 15
   - Multi-Line Statements ....................................................................................... 17
   - Quotation in Python ......................................................................................... 17
   - Comments in Python ........................................................................................ 17
   - Using Blank Lines ............................................................................................ 18
   - Waiting for the User ......................................................................................... 18
   - Multiple Statements on a Single Line ............................................................... 18
   - Multiple Statement Groups as Suites ............................................................... 19
   - Command Line Arguments ............................................................................ 19
5. **Python 3 – Variable Types** ........................................................................................................... 23
   Assigning Values to Variables ................................................................................................. 23
   Multiple Assignment .............................................................................................................. 23
   Standard Data Types .............................................................................................................. 24
   Python Numbers .................................................................................................................... 24
   Python Strings ........................................................................................................................ 25
   Python Lists ............................................................................................................................ 26
   Python Tuples .......................................................................................................................... 27
   Python Dictionary .................................................................................................................... 27
   Data Type Conversion .............................................................................................................. 28

6. **Python 3 – Basic Operators** .................................................................................................. 30
   Types of Operator ................................................................................................................... 30
   Python Arithmetic Operators ................................................................................................. 30
   Python Comparison Operators ............................................................................................... 32
   Python Assignment Operators ............................................................................................... 33
   Python Bitwise Operators ....................................................................................................... 35
   Python Logical Operators ....................................................................................................... 37
   Python Membership Operators ............................................................................................... 38
   Python Identity Operators ...................................................................................................... 39
   Python Operators Precedence ................................................................................................. 40

7. **Python 3 – Decision Making** ............................................................................................. 43
   IF Statement ........................................................................................................................... 44
   IF...ELIF...ELSE Statements ................................................................................................ 45
   Nested IF Statements ............................................................................................................ 48
   Single Statement Suites ........................................................................................................ 49

8. **Python 3 – Loops** ................................................................................................................. 51
   while Loop Statements .......................................................................................................... 52
   for Loop Statements ............................................................................................................. 56
   Nested loops .......................................................................................................................... 59
   Loop Control Statements ....................................................................................................... 60
   break statement .................................................................................................................... 61
   continue Statement ............................................................................................................... 63
   pass Statement ...................................................................................................................... 65
   Iterator and Generator ......................................................................................................... 66

9. **Python 3 – Numbers** ............................................................................................................. 68
   Mathematical Functions ......................................................................................................... 70
   Number abs() Method ........................................................................................................... 71
   Number ceil() Method .......................................................................................................... 71
   Number exp() Method .......................................................................................................... 72
   Number fabs() Method ......................................................................................................... 73
   Number floor() Method ....................................................................................................... 74
   Number log() Method .......................................................................................................... 75
   Number log10() Method ....................................................................................................... 76
   Number max() Method .......................................................................................................... 77
   Number min() Method .......................................................................................................... 78
   Number modf() Method ....................................................................................................... 79
10. Python 3 – Strings .......................................................... 99
    Accessing Values in Strings .............................................. 99
    Updating Strings .......................................................... 99
    Escape Characters ....................................................... 100
    String Special Operators ............................................. 101
    String Formatting Operator .......................................... 102
    Triple Quotes ................................................................ 104
    Unicode String ............................................................ 105
    String capitalize() Method ............................................ 109
    String center() Method ................................................ 110
    String count() Method .................................................. 111
    String decode() Method ................................................ 112
    String encode() Method ................................................ 112
    String endswith() Method ............................................ 113
    String expandtabs() Method ......................................... 114
    String find() Method .................................................... 115
    String index() Method .................................................. 116
    String isalnum() Method ............................................. 117
    String isalpha() Method .............................................. 118
    String isdigit() Method ................................................. 118
    String islower() Method .............................................. 119
    String isnumeric() Method ........................................... 120
    String isspace() Method .............................................. 121
    String istitle() Method ............................................... 122
    String isupper() Method .............................................. 122
    String join() Method .................................................... 123
    String len() Method ..................................................... 124
    String ljust() Method .................................................. 125
    String lower() Method .................................................. 125
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>String lstrip() Method</td>
<td>126</td>
</tr>
<tr>
<td>String maketrans() Method</td>
<td>127</td>
</tr>
<tr>
<td>String max() Method</td>
<td>128</td>
</tr>
<tr>
<td>String min() Method</td>
<td>129</td>
</tr>
<tr>
<td>String replace() Method</td>
<td>129</td>
</tr>
<tr>
<td>String rfind() Method</td>
<td>130</td>
</tr>
<tr>
<td>String rindex() Method</td>
<td>131</td>
</tr>
<tr>
<td>String rjust() Method</td>
<td>132</td>
</tr>
<tr>
<td>String rstrip() Method</td>
<td>133</td>
</tr>
<tr>
<td>String split() Method</td>
<td>134</td>
</tr>
<tr>
<td>String splitlines() Method</td>
<td>135</td>
</tr>
<tr>
<td>String startswith() Method</td>
<td>135</td>
</tr>
<tr>
<td>String strip() Method</td>
<td>136</td>
</tr>
<tr>
<td>String swapcase() Method</td>
<td>137</td>
</tr>
<tr>
<td>String title() Method</td>
<td>138</td>
</tr>
<tr>
<td>String translate() Method</td>
<td>138</td>
</tr>
<tr>
<td>String upper() Method</td>
<td>140</td>
</tr>
<tr>
<td>String zfill() Method</td>
<td>140</td>
</tr>
<tr>
<td>String isdecimal() Method</td>
<td>140</td>
</tr>
</tbody>
</table>

11. Python 3 – Lists ................................................................. 143

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python Lists</td>
<td>143</td>
</tr>
<tr>
<td>Accessing Values in Lists</td>
<td>143</td>
</tr>
<tr>
<td>Updating Lists</td>
<td>144</td>
</tr>
<tr>
<td>Delete List Elements</td>
<td>144</td>
</tr>
<tr>
<td>Basic List Operations</td>
<td>145</td>
</tr>
<tr>
<td>Indexing, Slicing and Matrixes</td>
<td>145</td>
</tr>
<tr>
<td>Built-in List Functions &amp; Methods</td>
<td>146</td>
</tr>
<tr>
<td>List len() Method</td>
<td>146</td>
</tr>
<tr>
<td>List max() Method</td>
<td>147</td>
</tr>
<tr>
<td>List min() Method</td>
<td>147</td>
</tr>
<tr>
<td>List list() Method</td>
<td>148</td>
</tr>
<tr>
<td>List append() Method</td>
<td>150</td>
</tr>
<tr>
<td>List count() Method</td>
<td>151</td>
</tr>
<tr>
<td>List extend() Method</td>
<td>151</td>
</tr>
<tr>
<td>List index() Method</td>
<td>151</td>
</tr>
<tr>
<td>List insert() Method</td>
<td>153</td>
</tr>
<tr>
<td>List pop() Method</td>
<td>154</td>
</tr>
<tr>
<td>List remove() Method</td>
<td>154</td>
</tr>
<tr>
<td>List reverse() Method</td>
<td>155</td>
</tr>
<tr>
<td>List sort() Method</td>
<td>156</td>
</tr>
</tbody>
</table>

12. Python 3 – Tuples ............................................................. 157

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessing Values in Tuples</td>
<td>157</td>
</tr>
<tr>
<td>Updating Tuples</td>
<td>158</td>
</tr>
<tr>
<td>Delete Tuple Elements</td>
<td>158</td>
</tr>
<tr>
<td>Basic Tuples Operations</td>
<td>159</td>
</tr>
<tr>
<td>Indexing, Slicing, and Matrixes</td>
<td>159</td>
</tr>
<tr>
<td>No Enclosing Delimiters</td>
<td>160</td>
</tr>
<tr>
<td>Built-in Tuple Functions</td>
<td>160</td>
</tr>
<tr>
<td>Tuple len() Method</td>
<td>160</td>
</tr>
<tr>
<td>Tuple max() Method</td>
<td>161</td>
</tr>
<tr>
<td>Method</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------</td>
</tr>
<tr>
<td>os.rmdir() Method</td>
<td>238</td>
</tr>
<tr>
<td>os.renames() Method</td>
<td>240</td>
</tr>
<tr>
<td>os.rename() Method</td>
<td>241</td>
</tr>
<tr>
<td>os.remove() Method</td>
<td>248</td>
</tr>
<tr>
<td>os.pipe() Method</td>
<td>250</td>
</tr>
<tr>
<td>os.open() Method</td>
<td>251</td>
</tr>
<tr>
<td>os.minor() Method</td>
<td>252</td>
</tr>
<tr>
<td>os.makedirs() Method</td>
<td>254</td>
</tr>
<tr>
<td>os.makedev() Method</td>
<td>255</td>
</tr>
<tr>
<td>Python os.close() Method</td>
<td>255</td>
</tr>
<tr>
<td>os.closerange() Method</td>
<td>256</td>
</tr>
<tr>
<td>os.dup() Method</td>
<td>258</td>
</tr>
<tr>
<td>os.dup2() Method</td>
<td>259</td>
</tr>
<tr>
<td>os.fchdir() Method</td>
<td>260</td>
</tr>
<tr>
<td>os.fchmod() Method</td>
<td>261</td>
</tr>
<tr>
<td>os.fchown() Method</td>
<td>263</td>
</tr>
<tr>
<td>os.fdatasync() Method</td>
<td>264</td>
</tr>
<tr>
<td>os.fdopen() Method</td>
<td>266</td>
</tr>
<tr>
<td>os.fpathconf() Method</td>
<td>267</td>
</tr>
<tr>
<td>os.fstat() Method</td>
<td>269</td>
</tr>
<tr>
<td>os.fstatvfs() Method</td>
<td>270</td>
</tr>
<tr>
<td>os.fsync() Method</td>
<td>272</td>
</tr>
<tr>
<td>os.ftruncate() Method</td>
<td>273</td>
</tr>
<tr>
<td>os.getcwd() Method</td>
<td>274</td>
</tr>
<tr>
<td>os.getcwdu() Method</td>
<td>275</td>
</tr>
<tr>
<td>os.isatty() Method</td>
<td>276</td>
</tr>
<tr>
<td>os.lchflags() Method</td>
<td>278</td>
</tr>
<tr>
<td>os.lchown() Method</td>
<td>279</td>
</tr>
<tr>
<td>os.link() Method</td>
<td>280</td>
</tr>
<tr>
<td>os.listdir() Method</td>
<td>281</td>
</tr>
<tr>
<td>os.lseek() Method</td>
<td>282</td>
</tr>
<tr>
<td>os.lstat() Method</td>
<td>284</td>
</tr>
<tr>
<td>os.major() Method</td>
<td>286</td>
</tr>
<tr>
<td>os.makedev() Method</td>
<td>286</td>
</tr>
<tr>
<td>os.makedirs() Method</td>
<td>288</td>
</tr>
<tr>
<td>os.minor() Method</td>
<td>288</td>
</tr>
<tr>
<td>os.mkdir() Method</td>
<td>289</td>
</tr>
<tr>
<td>os.mkfifo() Method</td>
<td>290</td>
</tr>
<tr>
<td>os.mknod() Method</td>
<td>291</td>
</tr>
<tr>
<td>os.open() Method</td>
<td>292</td>
</tr>
<tr>
<td>os.openpty() Method</td>
<td>293</td>
</tr>
<tr>
<td>os.pathconf() Method</td>
<td>294</td>
</tr>
<tr>
<td>os.pipe() Method</td>
<td>296</td>
</tr>
<tr>
<td>os.popen() Method</td>
<td>297</td>
</tr>
<tr>
<td>os.read() Method</td>
<td>298</td>
</tr>
<tr>
<td>os.readlink() Method</td>
<td>299</td>
</tr>
<tr>
<td>os.remove() Method</td>
<td>300</td>
</tr>
<tr>
<td>os.removedirs() Method</td>
<td>301</td>
</tr>
<tr>
<td>os.rename() Method</td>
<td>302</td>
</tr>
<tr>
<td>os.renames() Method</td>
<td>303</td>
</tr>
<tr>
<td>os.rename() Method</td>
<td>304</td>
</tr>
<tr>
<td>os.rmdir() Method</td>
<td>305</td>
</tr>
<tr>
<td>18. Python 3 – Exceptions Handling</td>
<td>323</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Standard Exceptions</td>
<td>323</td>
</tr>
<tr>
<td>Assertions in Python</td>
<td>325</td>
</tr>
<tr>
<td>What is Exception?</td>
<td>326</td>
</tr>
<tr>
<td>Handling an Exception</td>
<td>326</td>
</tr>
<tr>
<td>The except Clause with No Exceptions</td>
<td>328</td>
</tr>
<tr>
<td>The except Clause with Multiple Exceptions</td>
<td>328</td>
</tr>
<tr>
<td>The try-finally Clause</td>
<td>329</td>
</tr>
<tr>
<td>Argument of an Exception</td>
<td>330</td>
</tr>
<tr>
<td>Raising an Exception</td>
<td>331</td>
</tr>
<tr>
<td>User-Defined Exceptions</td>
<td>332</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19. Python 3 – Object Oriented</th>
<th>334</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of OOP Terminology</td>
<td>334</td>
</tr>
<tr>
<td>Creating Classes</td>
<td>335</td>
</tr>
<tr>
<td>Creating Instance Objects</td>
<td>336</td>
</tr>
<tr>
<td>Accessing Attributes</td>
<td>336</td>
</tr>
<tr>
<td>Built-In Class Attributes</td>
<td>337</td>
</tr>
<tr>
<td>Destroying Objects (Garbage Collection)</td>
<td>339</td>
</tr>
<tr>
<td>Class Inheritance</td>
<td>340</td>
</tr>
<tr>
<td>Overriding Methods</td>
<td>342</td>
</tr>
<tr>
<td>Base Overloading Methods</td>
<td>342</td>
</tr>
<tr>
<td>Overloading Operators</td>
<td>343</td>
</tr>
<tr>
<td>Data Hiding</td>
<td>344</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>20. Python 3 – Regular Expressions</th>
<th>346</th>
</tr>
</thead>
<tbody>
<tr>
<td>The match Function</td>
<td>347</td>
</tr>
<tr>
<td>The search Function</td>
<td>348</td>
</tr>
<tr>
<td>Matching Versus Searching</td>
<td>349</td>
</tr>
<tr>
<td>Search and Replace</td>
<td>350</td>
</tr>
<tr>
<td>Regular Expression Modifiers: Option Flags</td>
<td>350</td>
</tr>
<tr>
<td>Regular Expression Patterns</td>
<td>351</td>
</tr>
<tr>
<td>Regular Expression Examples</td>
<td>353</td>
</tr>
<tr>
<td>Character classes</td>
<td>354</td>
</tr>
</tbody>
</table>
21. **Python 3 – CGI Programming** ................................................................. 358
   - What is CGI? ......................................................................................... 358
   - Web Browsing ..................................................................................... 358
   - CGI Architecture Diagram .................................................................. 359
   - Web Server Support and Configuration .................................................. 359
   - First CGI Program ................................................................................. 360
   - HTTP Header ......................................................................................... 361
   - CGI Environment Variables ................................................................. 361
   - GET and POST Methods ...................................................................... 363
   - Passing Information using GET method ................................................. 363
   - Simple URL Example – Get Method ....................................................... 363
   - Simple FORM Example – GET Method .................................................. 364
   - Passing Radio Button Data to CGI Program ......................................... 367
   - Passing Text Area Data to CGI Program ............................................... 368
   - Passing Drop Down Box Data to CGI Program ...................................... 369
   - Using Cookies in CGI ........................................................................... 370
   - How It Works? ....................................................................................... 370
   - Setting up Cookies ............................................................................... 371
   - Retrieving Cookies ................................................................................ 371
   - File Upload Example ............................................................................ 372
   - How To Raise a "File Download" Dialog Box ? ...................................... 374

22. **Python 3 – MySQL Database Access** ............................................ 375
   - What is PyMySQL? ............................................................................... 375
   - How do I Install PyMySQL? .................................................................. 376
   - Database Connection ............................................................................ 376
   - Creating Database Table ....................................................................... 377
   - INSERT Operation .................................................................................. 378
   - READ Operation .................................................................................... 380
   - Update Operation ................................................................................... 382
   - DELETE Operation ................................................................................ 383
   - Performing Transactions ....................................................................... 383
   - COMMIT Operation ................................................................................ 384
   - ROLLBACK Operation .......................................................................... 384
   - Disconnecting Database ....................................................................... 384
   - Handling Errors .................................................................................... 385

23. **Python 3 – Network Programming** ............................................. 387
   - What is Sockets? .................................................................................... 387
   - The socket Module ............................................................................... 388
   - Server Socket Methods ....................................................................... 388
   - Client Socket Methods ........................................................................ 388
   - General Socket Methods ..................................................................... 389
Python 3 – Basic Tutorial
1. Python 3 – What is New?

The __future__ module

Python 3.x introduced some Python 2-incompatible keywords and features that can be imported via the in-built __future__ module in Python 2. It is recommended to use __future__ imports, if you are planning Python 3.x support for your code.

For example, if we want Python 3.x's integer division behavior in Python 2, add the following import statement.

```python
from __future__ import division
```

The print Function

Most notable and most widely known change in Python 3 is how the print function is used. Use of parenthesis () with print function is now mandatory. It was optional in Python 2.

```python
print "Hello World" # is acceptable in Python 2
print ("Hello World") # in Python 3, print must be followed by ()
```

The print() function inserts a new line at the end, by default. In Python 2, it can be suppressed by putting ', ' at the end. In Python 3, "end=' '" appends space instead of newline.

```python
print x,       # Trailing comma suppresses newline in Python 2
print(x, end=" ") # Appends a space instead of a newline in Python 3
```

Reading Input from Keyboard

Python 2 has two versions of input functions, input() and raw_input(). The input() function treats the received data as string if it is included in quotes '' or "", otherwise the data is treated as number.

In Python 3, raw_input() function is deprecated. Further, the received data is always treated as string.

In Python 2

```python
>>> x=input('something:')
something:10 #entered data is treated as number
>>> x
10
>>> x=input('something:')
```

In Python 3
Python 3

```python
something: '10'  # entered data is treated as string
>>> x
'10'
>>> x = raw_input("something:"")
something: 10  # entered data is treated as string even without '
'10'
>>> x = raw_input("something:"")
something: '10'  # entered data treated as string including ''
>>> x
"'10'"

In Python 3
>>> x = input("something:"")
something: 10
>>> x
'10'
>>> x = input("something:"")
something: '10'  # entered data treated as string with or without ''
>>> x
"'10'"
>>> x = raw_input("something:")  # will result NameError
Traceback (most recent call last):
  File "", line 1, in
      x = raw_input("something:"")
NameError: name 'raw_input' is not defined
```

**Integer Division**

In Python 2, the result of division of two integers is rounded to the nearest integer. As a result, 3/2 will show 1. In order to obtain a floating-point division, numerator or denominator must be explicitly used as float. Hence, either 3.0/2 or 3/2.0 or 3.0/2.0 will result in 1.5

Python 3 evaluates 3 / 2 as 1.5 by default, which is more intuitive for new programmers.

**Unicode Representation**

Python 2 requires you to mark a string with a `u` if you want to store it as Unicode.

Python 3 stores strings as Unicode, by default. We have Unicode (utf-8) strings, and 2 byte classes: byte and byte arrays.
**xrange() Function Removed**

In Python 2 range() returns a list, and xrange() returns an object that will only generate the items in the range when needed, saving memory.

In Python 3, the range() function is removed, and xrange() has been renamed as range(). In addition, the range() object supports slicing in Python 3.2 and later.

**raise exception**

Python 2 accepts both notations, the 'old' and the 'new' syntax; Python 3 raises a SyntaxError if we do not enclose the exception argument in parenthesis.

```
raise IOError, "file error"  #This is accepted in Python 2
raise IOError("file error") #This is also accepted in Python 2
raise IOError, "file error"  #syntax error is raised in Python 3
raise IOError("file error") #this is the recommended syntax in Python 3
```

**Arguments in Exceptions**

In Python 3, arguments to exception should be declared with 'as' keyword.

```
except Myerror, err:  # In Python2
except Myerror as err: #In Python 3
```

**next() Function and .next() Method**

In Python 2, next() as a method of generator object, is allowed. In Python 2, the next() function, to iterate over generator object, is also accepted. In Python 3, however, next(0 as a generator method is discontinued and raises AttributeError.

```
gen = (letter for letter in 'Hello World')  # creates generator object
next(my_generator)  #allowed in Python 2 and Python 3
my_generator.next() #allowed in Python 2. raises AttributeError in Python 3
```

**2to3 Utility**

Along with Python 3 interpreter, 2to3.py script is usually installed in tools/scripts folder. It reads Python 2.x source code and applies a series of fixers to transform it into a valid Python 3.x code.

```
def area(x,y=3.14):
    a=y*x*x
    print a
    return a
```
a=area(10)
print "area", a
To convert into Python 3 version:
$2to3 -w area.py
Converted code:
def area(x,y=3.14): # formal parameters
    a=y*x*x
    print (a)
    return a
a=area(10)
print("area",a)
Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas the other languages use punctuations. It has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner’s Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

- Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

- Python 1.0 was released in November 1994. In 2000, Python 2.0 was released. Python 2.7.11 is the latest edition of Python 2.

- Meanwhile, Python 3.0 was released in 2008. Python 3 is not backward compatible with Python 2. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules so that "There should be one -- and preferably only one -- obvious way to do it." Python 3.5.1 is the latest version of Python 3.
Python Features

Python's features include:

- **Easy-to-learn**: Python has few keywords, simple structure, and a clearly defined syntax. This allows a student to pick up the language quickly.

- **Easy-to-read**: Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain**: Python's source code is fairly easy-to-maintain.

- **A broad standard library**: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode**: Python has support for an interactive mode, which allows interactive testing and debugging of snippets of code.

- **Portable**: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Extendable**: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases**: Python provides interfaces to all major commercial databases.

- **GUI Programming**: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable**: Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features. A few are listed below:

- It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code for building large applications.

- It provides very high-level dynamic data types and supports dynamic type checking.

- It supports automatic garbage collection.

- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.
Try it Option Online
We have set up the Python Programming environment online, so that you can compile and execute all the available examples online. It will give you the confidence in what you are reading and will enable you to verify the programs with different options. Feel free to modify any example and execute it online.

Try the following example using our online compiler available at CodingGround

```bash
#!/usr/bin/python3
print("Hello, Python!")
```

For most of the examples given in this tutorial, you will find a Try it option on our website code sections, at the top right corner that will take you to the online compiler. Just use it and enjoy your learning.

Python 3 is available for Windows, Mac OS and most of the flavors of Linux operating system. Even though Python 2 is available for many other OSs, Python 3 support either has not been made available for them or has been dropped.

Local Environment Setup
Open a terminal window and type "python" to find out if it is already installed and which version is installed.

Getting Python

Windows platform
Binaries of latest version of Python 3 (Python 3.5.1) are available on this download page

The following different installation options are available.

- Windows x86-64 embeddable zip file
- Windows x86-64 executable installer
- Windows x86-64 web-based installer
- Windows x86 embeddable zip file
- Windows x86 executable installer
- Windows x86 web-based installer

**Note:** In order to install Python 3.5.1, minimum OS requirements are Windows 7 with SP1. For versions 3.0 to 3.4.x, Windows XP is acceptable.
**Linux platform**

Different flavors of Linux use different package managers for installation of new packages. On Ubuntu Linux, Python 3 is installed using the following command from the terminal.

```
$ sudo apt-get install python3-minimal
```

**Installation from source**

Download Gzipped source tarball from Python's download URL: [https://www.python.org/ftp/python/3.5.1/Python-3.5.1.tgz](https://www.python.org/ftp/python/3.5.1/Python-3.5.1.tgz)

Extract the tarball
```
tar xvfz Python-3.5.1.tgz
```

Configure and Install:
```
cd Python-3.5.1
./configure --prefix=/opt/python3.5.1
make
sudo make install
```

**Mac OS**

Download Mac OS installers from this URL: [https://www.python.org/downloads/mac-osx/](https://www.python.org/downloads/mac-osx/)

- Mac OS X 64-bit/32-bit installer: python-3.5.1-macosx10.6.pkg
- Mac OS X 32-bit i386/PPC installer: python-3.5.1-macosx10.5.pkg

Double click this package file and follow the wizard instructions to install.

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python:


You can download Python documentation from the following site. The documentation is available in HTML, PDF and PostScript formats.

**Python Documentation Website**: [www.python.org/doc/](http://www.python.org/doc/)

**Setting up PATH**

Programs and other executable files can be in many directories. Hence, the operating systems provide a search path that lists the directories that it searches for executables.

The important features are-

- The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.
• The path variable is named as **PATH** in Unix or **Path** in Windows (Unix is case-sensitive; Windows is not).

• In Mac OS, the installer handles the path details. To invoke the Python interpreter from any particular directory, you must add the Python directory to your path.

### Setting Path at Unix/Linux

To add the Python directory to the path for a particular session in Unix-

- **In the csh shell**: type `setenv PATH "$PATH:/usr/local/bin/python3"` and press Enter.

- **In the bash shell (Linux)**: type `export PATH="$PATH:/usr/local/bin/python3"` and press Enter.

- **In the sh or ksh shell**: type `PATH="$PATH:/usr/local/bin/python3"` and press Enter.

**Note:** `/usr/local/bin/python3` is the path of the Python directory.

### Setting Path at Windows

To add the Python directory to the path for a particular session in Windows-

**At the command prompt**: type `path %path%;C:\Python` and press Enter.

**Note:** `C:\Python` is the path of the Python directory.

### Python Environment Variables

Here are important environment variables, which are recognized by Python-

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PYTHONPATH</strong></td>
<td>It has a role similar to PATH. This variable tells the Python interpreter where to locate the module files imported into a program. It should include the Python source library directory and the directories containing Python source code. PYTHONPATH is sometimes, preset by the Python installer.</td>
</tr>
<tr>
<td><strong>PYTHONSTARTUP</strong></td>
<td>It contains the path of an initialization file containing Python source code. It is executed every time you start the interpreter. It is named as .pythonrc.py in Unix and it contains commands that load utilities or modify PYTHONPATH.</td>
</tr>
</tbody>
</table>
**PYTHONCASEOK**

It is used in Windows to instruct Python to find the first case-insensitive match in an import statement. Set this variable to any value to activate it.

**PYTHONHOME**

It is an alternative module search path. It is usually embedded in the PYTHONSTARTUP or PYTHONPATH directories to make switching module libraries easy.

---

**Running Python**

There are three different ways to start Python-

(1) **Interactive Interpreter**

You can start Python from Unix, DOS, or any other system that provides you a command-line interpreter or shell window.

Enter **python** the command line.

Start coding right away in the interactive interpreter.

```bash
$python       # Unix/Linux
or
python%     # Unix/Linux
or
C:>python    # Windows/DOS
```

Here is the list of all the available command line options-

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>provide debug output</td>
</tr>
<tr>
<td>-O</td>
<td>generate optimized bytecode (resulting in .pyo files)</td>
</tr>
<tr>
<td>-S</td>
<td>do not run import site to look for Python paths on startup</td>
</tr>
<tr>
<td>-v</td>
<td>verbose output (detailed trace on import statements)</td>
</tr>
<tr>
<td>-X</td>
<td>disable class-based built-in exceptions (just use strings); obsolete starting with version 1.6</td>
</tr>
<tr>
<td>-c cmd</td>
<td>run Python script sent in as cmd string</td>
</tr>
<tr>
<td>file</td>
<td>run Python script from given file</td>
</tr>
</tbody>
</table>

(2) **Script from the Command-line**
A Python script can be executed at the command line by invoking the interpreter on your application, as shown in the following example.

```
$python script.py  # Unix/Linux
or
python% script.py  # Unix/Linux
or
C:>python script.py # Windows/DOS
```

**Note:** Be sure the file permission mode allows execution.

(3) **Integrated Development Environment**
You can run Python from a Graphical User Interface (GUI) environment as well, if you have a GUI application on your system that supports Python.

- **Unix:** IDLE is the very first Unix IDE for Python.
- **Windows:** **PythonWin** is the first Windows interface for Python and is an IDE with a GUI.
- **Macintosh:** The Macintosh version of Python along with the IDLE IDE is available from the main website, downloadable as either MacBinary or BinHex'd files.

If you are not able to set up the environment properly, then you can take the help of your system admin. Make sure the Python environment is properly set up and working perfectly fine.

**Note:** All the examples given in subsequent chapters are executed with Python 3.4.1 version available on Windows 7 and Ubuntu Linux.

We have already set up Python Programming environment online, so that you can execute all the available examples online while you are learning theory. Feel free to modify any example and execute it online.
End of ebook preview

If you liked what you saw...

Buy it from our store @ https://store.tutorialspoint.com