Cheatography

Python Cheat Sheet

by binca via cheatography.com/44948/cs/13464/

Overview

Interpreted programming language developed in late 80s inspired by ABC language.

Extensibility is one of its major features. Libraries such as **Scapy** and **Requests** unlock Python's potential.

Basic Python scripts are **fast to write** and many libraries support easy creation of **HTTP requests, parsing of responses**

Many tools are written in Python.

It is widely available and is installed natively on macOS, most Linux distributions, annd some UNIX systems.

Python is easy to installed, and you can check version with **python -v**

Python requires consistent indentation, using 2 or 4 spaces is common. Tab should be avoided.

Python 2 versus Python 3

Python 2.x is legacy, Python 3.x is the present and future. The final major release of Python 2.7 was in 2010.

Python 2 is still the default version on macOS and Linux, though Python 3 is often included by called 'Python 3'

Python 3 Features

Major improvement is better Unicode support, all test strings being Unicode by default

Clean Unicode/byte separation

Exception chaining

Function annotations

Syntax for keyword-only arguments

Extended tuple unpacking

Non-local variable delcarations

Other changes include print and exec being statements and integers using floor division.

Data Types and Syntax

String	var="string"
Boolean	var=True
Integer	var=86
Float	var=3.14159
if/elif/else	conditional execution of functions
input( )	returns a string by default
int( )	changes a string to an integer
Boolean operators	and, or, not as well as comparison operators (<, <=, >, >=< ==)
for loops	iterates through a set
while loops	iterates until a condition met

Lists and Dictionaries

Lists are fundamental data structure they contain an ordered list of data.
**list = ['thing1', 'thing2', 'thing3']

Dictionaries are similar to lists but they are unordered key: value pairs.
**dictionary = {'key': 'value'}

In other languages, **dictionaries** are known as **associative arrays** or **hashes**.

Web Libraries

urllib

urllib2 - It can perform basic authentication, it does not handle underlying details like base-64 encoding or authorization headers. Python 3 splits functionality into urllib.request and urllib.error

urllib3

httplib - Python 3 renamed this **http.client**

httplib2

Requests developed with a number of PEP 20 idioms in mind

PEP= Python Enhancement Proposals
PEP 20 are "The Zen of Python"
Requests follows:
1. Beautiful is better than ugly
2. Explicit is better than implicit
3. Simple is better than complex
4. Complex is better than complicated
5. Readability counts.

Requests

Abstracts many lower-level details.

Supports multiple **authentication methods: Basic, Digest, Kerberos, NTLM, AWS, OAuth1**

Supports POST with options sent via a dictionary called 'data' in {'variable': 'value'}; multiple variables can be passed

Requests can also **POST** data from a file.

Handles **TLS/SSL** transparently verifying x.509 certificates by default (verify=True) and will exit if it is invalid. To connec tot a site with an invalid certificate by setting verify=False.
r=requests.get('https://'invalid.cert', verify=False)
print=(r.text)

Example of Post script:

```
#! /usr/bin/python3
import requests
r=requests.post('http://security.com/form_auth/login.php', data=
{'user': 'admin', 'pass': 'admin', 'button': 'Login'})
print(r.text)
```