

Branch: **master** [hackerU\\_python](#) / [lessonPlan](#) / [lesson1\\_variablesAndDataTypes.md](#) Find file Copy path

 **t-0-m-1-3** added folders 190da8f on Jul 30, 2018

1 contributor

265 lines (191 sloc) 6.2 KB Raw Blame History  

# Intro To Python For Security

---

## What is Python and What can I do with it?

---

Python is the 2nd best language for anything...

It was created by Guido Van Rossum in 1989 and he was dubbed Benevolent Dictator for Life of the language.

## Why Python

---

- Open Source, Multi-Platform
- Libraries
- Large number of open source projects to work on

## Implementations

---

- **Cpython**
- **Jython** - Python in Java
- **IronPython** - Python in C#

## Python has PEPs

---

- **PEP 8** is the style guide for Python

**Python Enhancement Proposal:** design documents providing information to the community, or describing a feature, its processes, or environment.

## Zen of Python (PEP 20)

*Tim Peters*

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality beats purity.  
Errors should never pass silently.  
Unless explicitly silenced.  
In the face of ambiguity, refuse the temptation to guess.  
There should be one-- and preferably only one --obvious way to do it.  
Although that way may not be obvious at first unless you're Dutch.  
Now is better than never.  
Although never is often better than *right* now.  
If the implementation is hard to explain, it's a bad idea.  
If the implementation is easy to explain, it may be a good idea.  
Namespaces are one honking great idea -- let's do more of those!
```

## Python in InfoSec

---

- Find a chart
- Find another Chart
- Rapid deployments
- Extensive library
- Large amounts of tools already built

## Python in OS's

---

- **Linux/Mac** have it pre-installed
- **Windows** needs to have it installed

## Python 2.7 vs 3.x?

- Most libraries are not compatible with each other.
- 3 is the standard; but the course is in 2.7.

## Run it in Kali Linux cause ( Debian )

---

## Interacting with Python

---

- **Terminal:** in the command prompt type `python` to get to the **Interpreter**

`python -v` will let you know which version of python you are running.

## Obligatory Hello World

---

```
python3.x print('Hello World')
```

```
echo print("Hello World") > firstPython.py  
python firstPython.py
```

should print out **Hello World**

```
python2.7 print 'hello world'
```

## Solving Compatibility Issues

---

- **VirtualEnv** allows for differences in dependencies in projects and python versioning.
- Install it using `apt-get install python-pip` and then running `pip install virtualenv`

You then need to *activate the environment* by running `source bin/activate`

### Exercise

1. Install Python 2.7 on Windows and Python 3 on Linux
2. write the hello world script on both machines
3. create the isolated env and run the script

## Variables:

---

Python reserves memory for variables using an assignment operator =

Based on the value type assigned to the variable, the interpreter allocates space

```
name = "David"; print(name)
```

the variable `name` have the value `david`

### Exercise: Create 10 variables; change their values.

---

#### **\*\* Multiple Assignment\*\***

capacity to assign a single value to several variables.

```
var1 = var2 = value3 = rand()
```

## Data Types:

---

Python's interpreter needs data types to know how much space to allocate

There are 5 Main Data Types:

1. String
2. Numbers
3. List
4. Tuples
5. Dictionary

## Strings (str): characters, contained in quotation marks.

---

Strings are immutable; they cannot be changed, only reinstantiated.

Strings can be operated on just like any variable in Python.

Strings can be sliced `string[1:3:5]`

Strings can be type-cast `str(42)` or `int("42")`

Strings have methods attached to them like `string.find('string')` or `string.split()`

Strings can also take in **Verbs** that allow variable content to be dynamically passed through. "hakin9 this IP: %s"%ip or "hakin9 this %s: %s"(domain, ip)

## Numbers

---

- Integers
- Floats
- Operators
  - Math
  - logical
  - bitwise
  - comparison

## Lists

---

A list contains values separated by commas and enclosed with square brackets `[]`

A list allows us to assign multiple values into one variable and access them via slicing and indexing.

### Access Items in a list:

To get more than one item from a list, you need to use a series of `[index:number:of:my:slice]` starting from left to right with ◦

Can use iteration ( `for` loops ) to access in lists as well

## Tuples

---

A tuple is similar to a list, with the exception that: 1. Lists are enclosed with brackets [ ] and tuples are enclosed with parenthesis ( ). 2. List items can be updated; tuples are immutable.

You can convert from 'list' to 'tuple' `tuple(list)` or `list(tuple)`

## Sets

---

Unordered collection of unique objects

you can take a *union* of a set with `a|b` you can find the *intersection* of a set with `a&b`

## Dictionaries:

---

Unordered pairs of key-values

The keys are unique and immutable objects the values can change.

```
dict = {}
dict['name']='dave'
dict(name='dave', age='30')
dict = {'name':'dave', 'age','30'}
dict.has_key('name') # this will check for a given key name in a dictionary
dict.keys()
dict.items()
dict.values()
dict.get('age')
del dict['name']
dict.clear()
dir(dict)

help(dict.has_key)
```

## Dictionary Exercise:

---

1. Get from user Name and Last name using a While Loop
2. Create a code that adds numbers to the list and shows the user: a. Average b. Sum c. Length
3. Create a dictionary and save in the following: a. SSH:22 b. FTP:21 c. Telnet:23 d. RDP:3389 4. Intro Get a list from user, and show items using the for loop.
4. Using a for loop, sum all even numbers between 1 and 1000. 6. Intro Get a list from user, and print the items in a reverse way.
5. Get 5 numbers from the user and print the highest number? 8. Intro Get 10 numbers into a list a. Ask the user to insert a number and check if it's in the list. b. Remove the 5th item
6. Get 2 numbers from user and show: a. addition b. subtraction c. multiplication d. division with remainder
7. Using a while loops get 10 keys and values from the user: a. Show only keys b. show only values