

# Python Programming Language Notes

Timothy J. Helton  
720.641.8370  
timothy.j.helton@gmail.com

June 14, 2016

# Contents

<b>1</b>	<b>Built-In Functions</b>	<b>1</b>
1.1	abs()	1
1.2	all()	1
1.3	any()	1
1.4	ascii()	1
1.5	bin()	1
1.6	bool()	2
1.7	bytearray()	2
1.8	bytes()	2
1.9	callable()	2
1.10	chr()	2
1.11	classmethod()	2
1.12	compile()	2
1.13	complex()	2
1.14	delattr()	2
1.15	dict()	2
1.16	dir()	2
1.17	divmod()	2
1.18	enumerate()	2
1.19	eval()	2
1.20	exec()	2
1.21	filter()	2
1.22	float()	2
1.23	format()	2
1.24	frozenset()	2
1.25	getattr()	2
1.26	globals()	2
1.27	hasattr()	2
1.28	hash()	2
1.29	help()	2
1.30	hex()	2
1.31	id()	2
1.32	input()	2
1.33	int()	2
1.34	isinstance()	2
1.35	issubclass()	2
1.36	iter()	2
1.37	len()	2
1.38	list()	2
1.39	locals()	2
1.40	map()	2
1.41	max()	2
1.42	memoryview()	2
1.43	min()	2
1.44	next()	2
1.45	object()	2
1.46	oct()	2
1.47	open()	2

1.48	ord()	2
1.49	pow()	2
1.50	print()	2
1.51	property()	2
1.52	range()	2
1.53	repr()	2
1.54	reversed()	2
1.55	round()	2
1.56	set()	2
1.57	setattr()	2
1.58	slice()	2
1.59	sorted()	2
1.60	staticmethod()	2
1.61	str()	2
1.62	sum()	2
1.63	super()	2
1.64	tuple()	2
1.65	vars()	2
1.66	zip()	2
1.67	__import__()	2
<b>2</b>	<b>pip - Python Package Index</b>	<b>3</b>
2.1	Install pip	3
2.1.1	Without Internet Connection	3
2.2	Find the Site Packages Installation Directory	3
2.3	Install Packages	3
2.3.1	Install a Single Package	3
2.3.2	Install Packages From requirements.txt File	3
2.3.3	Install a Package From Wheel	3
2.3.4	Install Package in Developer Mode	3
2.4	List Outdated Modules	4
2.5	Change the Version of an Installed Package	4
2.5.1	Upgrade to the Latest Version	4
2.5.2	Install a Previous Version	4
2.6	Create a Wheels	4
2.7	Package Configuration	4
<b>3</b>	<b>numpy</b>	<b>5</b>
<b>4</b>	<b>pandas</b>	<b>6</b>
<b>5</b>	<b>scipy</b>	<b>7</b>

## List of Figures

## List of Tables

## Nomenclature

pip            Python Package Index

# 1 Built-In Functions

## 1.1 abs()

Input must be an **int** , **float** or **complex number** .

- Returns the absolute value if the argument is a float or int.
- Returns the magnitude if the argument is a complex number.

## 1.2 all()

Input must be an **iterable** .

- Returns True if:
  - all elements of the iterable are true
  - the iterable is **empty**

## 1.3 any()

Input must be an **iterable** .

- Returns True if:
  - any of the elements of the iterable are true
- Returns False if:
  - the iterable is **empty**

## 1.4 ascii()

Input is an **object** .

Use this function to display a printable representation of an object, similar to repr(), but use escapes for non-ascii characters.

## 1.5 bin()

Input must be an **int** or an **object with a \_\_index\_\_() method** that returns an int.

Convert an integer to a binary string.

1.6 `bool()`  
1.7 `bytearray()`  
1.8 `bytes()`  
1.9 `callable()`  
1.10 `chr()`  
1.11 `classmethod()`  
1.12 `compile()`  
1.13 `complex()`  
1.14 `delattr()`  
1.15 `dict()`  
1.16 `dir()`  
1.17 `divmod()`  
1.18 `enumerate()`  
1.19 `eval()`  
1.20 `exec()`  
1.21 `filter()`  
1.22 `float()`  
1.23 `format()`  
1.24 `frozenset()`  
1.25 `getattr()`  
1.26 `globals()`  
1.27 `hasattr()`  
1.28 `hash()`  
1.29 `help()`  
1.30 `hex()`  
1.31 `id()`  
1.32 `input()`  
1.33 `int()`  
1.34 `isinstance()`  
1.35 `issubclass()`  
1.36 `iter()`  
1.37 `len()`  
1.38 `list()`  
1.39 `locals()`  
1.40 `map()`  
1.41 `max()`



## 2 pip - Python Package Index

### 2.1 Install pip

#### 2.1.1 Without Internet Connection

1. Download `get-pip.py`
2. On a computer with an internet connection create the following wheels.
  - pip
  - setuptools
3. Move the wheels to the computer without an internet connection.
4. Call the following command

---

```
python get-pip.py --no-index --find-links=WheelHouseDirectory
```

---

### 2.2 Find the Site Packages Installation Directory

---

```
python -c "from distutils.sysconfig
import get_python_lib;
print get_python_lib()"
```

---

### 2.3 Install Packages

#### 2.3.1 Install a Single Package

---

```
pip install PackageName
```

---

#### 2.3.2 Install Packages From requirements.txt File

---

```
pip install -r requirements.txt
```

---

#### 2.3.3 Install a Package From Wheel

- If you do not want to use the cached wheel then add the following argument.

---

```
--no-cache-dir
```

---

---

```
pip install --use-wheel --no-index --find-links=WheelHouseDirectory PackageName
```

---

#### 2.3.4 Install Package in Developer Mode

This option allows a package to be actively developed while being installed in a Python interpreter.

---

```
pip install -e .
```

---

## 2.4 List Outdated Modules

---

```
pip list -o
```

---

## 2.5 Change the Version of an Installed Package

### 2.5.1 Upgrade to the Latest Version

---

```
pip install --upgrade PackageName
```

---

### 2.5.2 Install a Previous Version

---

```
pip install --upgrade PackageName==Version
```

---

## 2.6 Create a Wheels

---

```
pip wheel --wheel-dir=WheelHouseDirectory
```

---

## 2.7 Package Configuration

The package configuration may be maintained by a single text file, which is commonly called requirements.txt. Boolean operations may also be used in these configuration files. To create a snapshot of the current interpreter use the following command.

---

```
pip freeze > requirements.txt
```

---

### 3    `numpy`

## 4 pandas

## 5    `scipy`

## References