

PYTHON GUIDE 3.7	
{ DICTIONARY (Hashtable)	
Unordered, Mutable	
a = {'A':100, 'B':200, 'C':300}	
a['lettuce'] = 900	
del a['two']	
for k in a: pass	
for k,v in a: pass	
if 'apples' in a: print("a as in apple")	
[] LIST (Dynamic Array)	
Ordered, Mutable	
a = [100, 'asdf', 2.3]	
a.append(400)	
a.insert(0, 500)	
a.remove(0)	
a.sort(key=str.lower,reverse=False)	
for i in range(len(a)): pass	
for v in a: pass	
() TUPLE (Array)	
Ordered, Immutable	
a = (100, 'asdf', 2.3)	
for i in range(len(a)): pass	
for v in a: pass	
f{} Set (Hashtable)	
Unordered, Mutable	
a = {1,2,3}	
b = {3,4,5}	
print(a b) # a.union(b)	
print(a&b) # a.intersection(b)	
print(a-b) # a.difference(b)	
print(a^b) # a.difference_symmetric(b)	
f(x) FUNCTIONS	
def listArgs(*args):	
for v in args: pass	
listArgs(1, 2, 3)	
def keywordArgs(**kwargs):	
for k in kwargs: pass	
keywordArgs(red=1, green=2, blue=3)	
def f(op, x, y): return op(x, y)	
def add(x, y): return x + y	
print(f(add, 10, 7))	
e = lambda x,y: x**2 + y + 7	
[X] COMPREHENSIONS	
[char for char in 'Hello, World!']	
[x for x in range(50) if x % 3 == 0 if x % 9]	
{v: foo(v) for v in sequence if bar(v)}	
[await fun() for fun in funcs]	
[i async for i in agen()]	
C[] CLASSES	
class A:	
a = 1.1 # class scope	
def __init__(self, a): pass	
def f(self): pass	
test = A('Apples')	
test.f()	
[:.] SLICING	
a[start:end] # from start to end-1	
a[start:] # all from start	
a[:end] # beginning to end-1	
W: WITH	
with open('out.txt', 'w') as f:	
f.write('Hello, World!') # File closed after	
=> IMPORT	
import X	
from X import a, b, c	
from X import *	
S STRINGS	
# Comment	
""" Multiline String """	
{0} {1} {2}".format(100,'abc',2.0)	

and	await	def	except
as	break	del	False
assert	class	elif	finally
async	continue	else	for
Truth Testing			
__bool__(), __len__()			
Equal False			
False, None, 0, 0.0, 0j,			
Decimal(0), Fraction(0,1)			
Boolean Ops			
a and b, x or y, not x			
Comparisons			
< <= > >= == !=			
is, is not			
Arithmetic			
+ - * ** / // %			
Int(v)			
bit_length(), to_bytes()			
from_bytes()			
float(v)			
as_integer_ratio()			
is_integer()			
hex(), from_hex()			
complex(v,i)			
c.real, c.imaginary			
Bitwise			
& ^ >> << ~			
Container			
.iterator()			
Iterator			
.iter()			
. __next__()			
Sequence Types			
list, tuple, range			
str, bytes, bytearray			
Sequence Ops			
x in s	len(s)		
x not in s	min(s)		
s + t	max(s)		
s * n	s.count(x)		
s[i:j:k]	s.index(x)		
Mutable Sequence Ops			
del[i:j:k]	s*=n		
s.append(x)	s.insert(i,x)		
s.clear()	s.pop(i)		
s.copy()	s.remove(x)		
s.extend(t) s+=t	s.reverse()		
Text Sequence Ops			
capitalize()	join()		
casefold()	ljust()		
center()	lower()		
count()	lstrip()		
encode()	maketrans()		
endswith()	partition()		
expandtabs()	replace()		
find()	rfind()		
format()	rindex()		
format_map()	rjust()		
index()	rpartition()		
isalpha()	rsplit()		
isascii()	rstrip()		
isdecimal()	split()		
isdigit()	splitlines()		
isidentifier()	startswith()		
islower()	strip()		
isnumeric()	swapcase()		
isprintable()	title()		
isspace()	translate()		
istitle()	upper()		
isupper()	zfill()		
String Formatting			
3.1415	{: .2f}	3.14	
2.718	{: .0f}	3	
5	{:0>3d}	005	
5	{:x<4d}	5xxx	
0.25	{: .2%}	25.00%	
1030	{: .2e}	1.03E+03	
13	{:10d}		13
13	{:<10d}	13	

__import__()	abs()	all()	any()	basestring()	bin()	bool()	bytearray()	callable()	chr()	classmethod()	cmp()	compile()	complex()	delattr()	dict()	dir()	divmod()	enumerate()	eval()	execfile()	file()	filter()	float()	format()	frozenset()	getattr()	globals()	hasattr()	hash()	help()	hex()	id()	input()	int()	isinstance()	issubclass()	iter()	len()	list()	locals()	long()	map()	max()	memoryview()	min()	next()	object()	oct()	open()	ord()	pow()	print()	property()	range()	raw_input()	reduce()	reload()	repr()	reversed()	round()	set()	setattr()	slice()	sorted()	staticmethod()	str()	sum()	super()	tuple()	type()	unichr()	unicode()	vars()	xrange()	zip()
--------------	-------	-------	-------	--------------	-------	--------	-------------	------------	-------	---------------	-------	-----------	-----------	-----------	--------	-------	----------	-------------	--------	------------	--------	----------	---------	----------	-------------	-----------	-----------	-----------	--------	--------	-------	------	---------	-------	--------------	--------------	--------	-------	--------	----------	--------	-------	-------	--------------	-------	--------	----------	-------	--------	-------	-------	---------	------------	---------	-------------	----------	----------	--------	------------	---------	-------	-----------	---------	----------	----------------	-------	-------	---------	---------	--------	----------	-----------	--------	----------	-------

key function	keyword argument	L	lambda	LBYL	list	list comprehension	loader	M	mapping	meta path finder	metaclass	method	method resolution order	module	module spec	MRO	mutable	N	named tuple	namespace	nested scope	new-style class	O	object	P	package	parameter	path entry	path entry finder	path entry hook	path-based finder	path-like object	portion	positional argument	provisional API	provisional package	Python 3000	PEP	S	__slots__	sequence	single dispatch	slice	special method	statement	struct sequence	T	text encoding	text file	triple-quoted string	type	type alias	type hint	U	universal newlines	V	variable annotation	virtual environment	virtual machine	Zen of Python
--------------	------------------	---	--------	------	------	--------------------	--------	---	---------	------------------	-----------	--------	-------------------------	--------	-------------	-----	---------	---	-------------	-----------	--------------	-----------------	---	--------	---	---------	-----------	------------	-------------------	-----------------	-------------------	------------------	---------	---------------------	-----------------	---------------------	-------------	-----	---	-----------	----------	-----------------	-------	----------------	-----------	-----------------	---	---------------	-----------	----------------------	------	------------	-----------	---	--------------------	---	---------------------	---------------------	-----------------	---------------

__import__()	abs()	all()	any()	basestring()	bin()	bool()	bytearray()	callable()	chr()	classmethod()	cmp()	compile()	complex()	delattr()	dict()	dir()	divmod()	enumerate()	eval()	execfile()	file()	filter()	float()	format()	frozenset()	getattr()	globals()	hasattr()	hash()	help()	hex()	id()	input()	int()	isinstance()	issubclass()	iter()	len()	list()	locals()	long()	map()	max()	memoryview()	min()	next()	object()	oct()	open()	ord()	pow()	print()	property()	range()	raw_input()	reduce()	reload()	repr()	reversed()	round()	set()	setattr()	slice()	sorted()	staticmethod()	str()	sum()	super()	tuple()	type()	unichr()	unicode()	vars()	xrange()	zip()
--------------	-------	-------	-------	--------------	-------	--------	-------------	------------	-------	---------------	-------	-----------	-----------	-----------	--------	-------	----------	-------------	--------	------------	--------	----------	---------	----------	-------------	-----------	-----------	-----------	--------	--------	-------	------	---------	-------	--------------	--------------	--------	-------	--------	----------	--------	-------	-------	--------------	-------	--------	----------	-------	--------	-------	-------	---------	------------	---------	-------------	----------	----------	--------	------------	---------	-------	-----------	---------	----------	----------------	-------	-------	---------	---------	--------	----------	-----------	--------	----------	-------