

Crash course in Python 3.x



Quick Reference!

Data Type Summary		Arithmetic Operations	Identity Operators
Numeric (Integer)	Store whole number values	+ Addition (or string concatenation) - Subtraction or Negation * Multiplication (or string repetition) / Division % Modulus ** Exponential // Floor Division (integer)	x is True if x and y are variables that reference the same data underneath! x is Only true if x and y are independent. not They may have the same value! y
Numeric (Float)	Store decimal number values		Identity operators verify that the variables are located on the same part of the memory.
Numeric (Complex)	Stores complex number values		
Boolean	Stores true (or false) values (in an integer)		
String	Stores an ordered sequence of alphanumeric character values		
Tuple	Stores an ordered sequence of immutable data values. ()		
Lists	Stores an ordered sequence of mutable data values. []		
Dictionaries	Stores key value pairs (mappings) {}		
Set	Stores an unordered collection of unique and immutable objects		
Slices		Logical Operators	Assignment Operators
[x:y]	Slices from index x up to index y (up to!)	and Logical And or Logical Or not Logical Not	x = y Simple assignment x=y x += y x=x+y x -= y x=x-y x *= y x=x*y x /= y x=x/y x %= y x=x%y x //= y x=x//y x **= y x=x**y x &= y x=x & y x = y x=x y x ^= y x=x^y x >>= y x=x >> y x <<= y x=x << y
[x:]	Slices from index x to last index in the set		
[::y]	Slices from the first index up to index y		
[::]	Slices the entire set		
[:-1]	Slices the last item from the set		
[:-x]	Slices everything except the last x items from the set. (generalized)		
[x:y:z]	Slices from index x up to index y (up to!), by step z	def MaxFunc(x,y): if x>y: return x else: return y	
Bitwise Operators		Simple Function	Sample If Statement
		<< Shift left >> Shift Right & Binary AND Binary OR ~ Binary NOT ^ Binary XOR	if crew_age < 10: rank = junior elif age < 18: rank = ensign else: rank = commander
		Don't confuse binary not with logical not!	
Membership Operators		Membership Operators	
		x in y True if the value x can be found in y. x not in y True if the value x can not be found in y.	

For additional code samples and resources be certain to visit:
<http://www.tbdatascientist.com/live1.html>

Crash course in Python 3.x

Quick Reference!



Copyright (C) David Valentine
2018

List Operations

Creating an list

```
crew = ['Spock','McCoy']
```

Getting first item from the list

```
first = crew[0]
```

Get the last item from the list

```
last = crew[-1]
```

Looping through the list

```
for person in crew:  
    print(person)
```

Adding items to a list

```
crew.append('Kirk')
```

Slicing a list

```
human=crew[1:2]
```

Pointer to a list (same data)

```
crew2=crew
```

Duplicate of the list (copy of data)

```
duplicate=crew[:]
```

Simple while loop

```
i=0
```

```
while i<10:
```

```
    i=i+1
```

```
    print("loop - iteration #",i)
```

```
print("Done")
```

Loop control

```
break immediate exit of loop
```

```
continue resumes at test condition of the loop
```

Dictionaries

Creating a dictionary

```
example_dictionary =  
{1:"Orange",2:"Apple",3:"Banana",4:"Peach",  
5:"Pear"}
```

Accessing a value

```
print (example_dictionary[2])
```

Adding a new key-value pair

```
example_dictionary[15]="Plum"
```

Looping through all key-value pairs

```
for key in example_dictionary:  
    print(key,example_dictionary[key])
```

Looping through all keys

```
for key in example_dictionary.keys():  
    print("Key:",key)
```

Looping through all values

```
for value in example_dictionary.values():  
    print("Value:",value)
```

For additional code samples and resources be certain to visit:
<http://www.tbdatascientist.com/live1.html>