Python Basics

The purpose of this document is not to make you a Python wizard, it is only to show the most basic syntax of the language in order for you to be able to make a basic program.

Variables

Variables are declared with only its name and value and are *not* declared with a type, Python will figure that out for you. There is also the special value of `None`, which is Python's null value.

Strings

Strings are variables containing letters. They have a length, they can be concatenated and they can be indexed.

Examples

```
# A variable with a value of 10
foo = 10

# A variable containing a string
word = "foobar"

# A null variable
nothing = None

# A variable with a value of -3.14
neg_pi = -3.14

# Print the length of word
print(len(word))
# >>> 4

print(word + " forever")
# >>> foobar forever

print(word[0:3])
# >>> foo
```

Conditionals

The path of the program can be changed through the use of conditionals (if statements).

Examples

```
word = "foobar"
if len(foobar) == 5:
    print(f"{word} has a length of 5.")
elif len(foobar) == 6:
    print(f"{word} has a length of 6.")
else:
    print(f"{word} is not 5 or 6 characters in length.")
# >>> foobar has a length of 6

x = True
if x:
    print("x is Truthy")
else:
    print("x is Falsy")
# >>> x is Truthy
```

Loops

Python has two kinds of loops `for` loops and `while` loops. `for` loops enumerate over iterators, `while` loops loop until the condition becomes falsy.

Examples

```
for i in range(3):
    print(i)
# >>> 0
# >>> 1
# >>> 2
l = [1, 2, 3]
for x in l:
    print(x)
# >>> 1
# >>> 2
# >>> 3
x = True
while x:
    print(x)
    x = False
print(x)
# >>> True
# >>> False
```

Lists

Lists are collections that contain other variables in a specific order. Lists are heterogeneous, meaning the elements of the list can be of any type, and of mixed type.

Examples

```
# Creating an empty list
l = []

l = ['foo', 3.14, [True, -6], False]
print(l[1])
# >>> 3.14
print(l[2:])
# >>> [[True, -6], False]
l.append(42)
print(l)
# >>> ['foo', 3.14, [True, -6], False, 42]
l.push(99)
print(l)
# >>> [99, 'foo', 3.14, [True, -6], False, 42]
print(l.pop())
print(l)
# >>> 42
# >>> [99, 'foo', 3.14, [True, -6], False]
```

Dictionaries

Classes

Files

Main and Arguments