CSE 106

Lecture 2 – Python

Variables/Types, Lists, Basic operators, String formatting, String operations

Acknowledgement: many code examples from learnpython.org

Variables and Types

- Python is completely object oriented
- Every variable in Python is an object
- Not statically typed
  - No need to declare variables before using them
  - No need to declare variable types

Variables and Types - Numbers

- Integers

```
myint = 7
print(myint)
```

- Floating point numbers

```
myfloat = 7.0
print(myfloat)
```

- Operators (+, -, /, *, etc.)

```
one = 1
two = 2
three = one + two
print(three)
```

Variables and Types - Strings

- Double or single quote (double is easier for apostrophes)

```
mystring = 'hello'
print(mystring)
mystring = "hello"
print(mystring)
mystring = "Don't worry about apostrophes"
print(mystring)
```

- New line (\n)

```
print('First line.\nSecond line.')
```

- Operators

```
hello = "hello "
world = "world"
print(hello + " " + world)
```

Variables and Types

- Assign values to multiple variables at once

```
a, b = 3, 4
print(a,b)
```

- Ask the type using the `type` function

```
name = "Sam"
print(type(name))
```

- No mixing operators between numbers and strings

```
one = 1
two = 2
hello = "hello"
print(one + two + hello)  # results in error message
```

Variables and Types - Casting

- Need to cast to make this work

```
one = 1
two = 2
hello = "hello"
print( str(one + two) + hello )
```

- Cast examples:

```
x = float(1)       # x will be 1.0
y = int(2.8)       # y will be 2
z = float("4.2")   # z will be 4.2
w = str(2)         # w will be ‘2’
v = str(3.0)       # v will be ‘3.0’
```

Lists

- Ordered set of items, accessed by an index
- Enclosed in square brackets

```
list = [3, 6, 9, 12]
print(list[2])
print(list)
```

- Can include different data types (e.g. strings, integers, objects, etc.)
- Mutable – You can change the contents after its creation
- Items in list do not need to be unique
- Stored as dynamic arrays in memory (not linked list)

Lists

```
mylist = []
mylist.append(10)
mylist.append(20)
mylist.append(30)
print(mylist[0]) # prints 10
print(mylist[1]) # prints 20
print(mylist[2]) # prints 30

# prints out 10 20 30
for item in mylist:
    print(item)
```

Lists

- Common methods
  - append() Adds an element at the end of the list
  - clear() Removes all the elements from the list
  - count() Returns the number of elements with the specified value
  - extend() Add the elements of a list to the end of the current list
  - insert() Adds an element at the specified position
  - reverse() Reverses the order of the list
  - sort() Sorts the list
- Accessing an index which does not exist generates an exception

```
mylist = [1,2,3]

print(mylist.count())
```

Wake up!

https://youtu.be/nMJdsQL_Bco

Basic Operators - Arithmetic

- Addition, subtraction, multiplication, and division operators

```
number = 1 + 2 * 3 / 4.0

print(number)
```

- Modulo (%)

```
remainder = 11 % 3

print(remainder)
```

- Power

```
cubed = 2 ** 3

print(cubed)
```

Basic Operators - Strings

- Adding strings

```
helloworld = "hello" + " " + "world"

print(helloworld)
```

- String with a repeating sequence

```
lotsofhellos = "hello" * 10

print(lotsofhellos)
```

Basic Operators - Lists

- Lists joined with addition operator

```
even_numbers = [2,4,6,8]

odd_numbers = [1,3,5,7]

all_numbers = odd_numbers + even_numbers

print(all_numbers)
```

- Form new lists with a repeating sequence

```
print([1,2,3] * 3)
```

String Formatting

- Python uses C-style string formatting to create new, formatted strings
- The "%" operator is used to format a set of variables
- Uses special symbols like "%s" and "%d” as place holders
- Single formatted variable:

```
name = "John"

print("Hello, %s!" % name)
```

String Formatting

- Multiple formatted variables:

```
name = "John"

age = 23

print("%s is %d years old." % (name, age))
```

- Any object which is not a string can be formatted using %s

```
# This prints out: A list: [1, 2, 3]

mylist = [1,2,3]

print("A list: %s" % mylist)
```

String Operations

- Length of string

```
astring = "Hello world!"

print(len(astring))
```

- Finding the index of a letter (only finds the first one)

```
astring = "Hello world!"

print(astring.index("o"))
```

- Count the number of times a letter occurs in a string

```
astring = "Hello world!"

print(astring.count("l"))
```

String Operations

- Get a portion of the string

```
astring = "Hello world!"

print(astring[6:11]) # just print world!
```

- All upper or lower case

```
astring = "Hello world!"

print(astring.upper())

print(astring.lower())
```

String Operations

- Ask if it starts with a specific word

```
astring = "Hello world!"

print(astring.startswith("Hello"))

print(astring.endswith("asdfasdfasdf"))
```

- List of words in string

```
astring = "Hello world!"

stringList = astring.split(" ")

print(stringList)
```

For next time

- Python
  - Conditions
  - Loops
  - Functions
  - Input/output from files
  - Classes and Objects
  - Dictionaries
  - Modules and Packages