Python

Session 6

By Declan Fox
With thanks to Al Sweigart

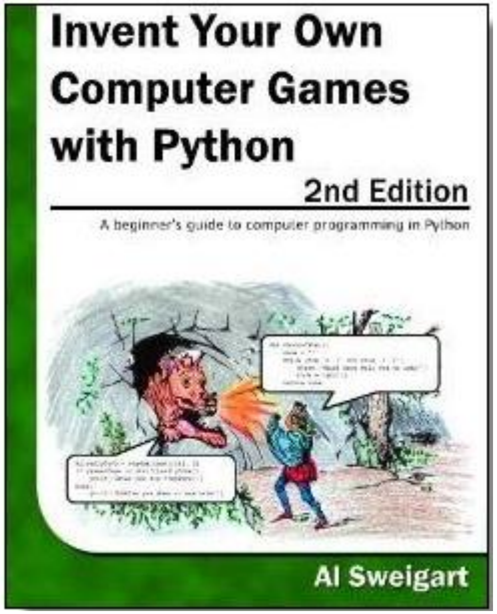
“Above all, be cool.”

Wi-Fi Name: CoderDojo

Password: coderdojowireless

Website: http://cdathenry.wordpress.com/

Remember to check out “Invent your own computer games with Python” by Al Sweigart.



Invent Your Own Computer Games with Python

2nd Edition

A beginner's guide to computer programming in Python

Al Sweigart

The book is free online at

http://inventwithpython.com

New Topics

- Multi-line strings
- Lists
- for loops
- Methods
    - list methods
    - string methods
- elif statements
- The dictionary data type.
- Multiple variable assignment, such as a, b, c = [1, 2, 3]

Multi-line strings

Ordinarily when you write strings in your source code, the string has to be on one line.
However, if you use three single-quotes instead of one single-quote to begin and end the string, the string can be on several lines.

Lists

Lists are a new data type.
A list value can contain several other values in it.
Lists are a good way to store several different values into one variable.
The individual values inside of a list are also called **items**.

Lists

Creating a list

```
Judges = [‘Simon’, ‘Sharon’, ‘Danni’, ‘Louis’]
```

Printing a list

```
Print (judges)
```

To print individual items from the list

```
Print (judges[1])
```

This would print out Sharon

The number in the square brackets is called the index

In Python, the first index is the number 0 instead of the number 1

So to get Simon we would need to do the following

```
Print (judges[0])
```

List Concatenation

We can use the + sign to join lists together.

```
judges = ['Simon', 'Louie', 'Sharon', 'Danni']
print(judges)

presenters = ['Dermot', 'Holly',]
print(presenters)

people = judges + presenters
print(people)
```

List in Operator

The in Operator

The in operator makes it easy to see if a value is inside a list or not.
Expressions that use the in operator return a Boolean value:
True if the value is in the list and False if the value is not in the list.

Removing Items from Lists with del Statements

You can remove items from a list with a del statement. ("del" is short for "delete.")

```
del judges[0]
```

Lists of Lists

Lists are a data type that can contain other values as items in the list.
But these items can also be other lists.
For example we could make up a list of judges, a list of presenters and a list of winners and make a list to contain these lists called xFactor.
xFactor = [judges, presenters, winners]

To get an item inside the list of lists, you would use *two* sets of square brackets like this: listOfLists[1][2]

Methods

Methods are just like functions, but they are always attached to a value.
For example, all string values have a lower() method, which returns a copy of the string value in lowercase.
You cannot just call lower()by itself and you do not pass a string argument to lower() by itself (as in lower('Hello')).
You must attach the method call to a specific string value using a dot (full stop).

List Methods

The list data type also has methods.
The reverse() method will reverse the order of the items in the list.

The most common list method you will use is append(). This method will add the value you pass as an argument to the end of the list.

The Difference Between Methods and Functions

- You may be wondering why Python has methods, since they seem to act just like functions. Some data types have methods. Methods are functions associated with values of that data type. For example, string methods are functions that can be called on any string.

- You cannot call string methods on values of other data types. For example, [1, 2, 'apple'].upper() would cause an error because [1, 2, 'apple'] is a list and upper() is a string method.

- The values of data types that have methods are also called objects. Object-oriented programming is a bit advanced for this group, and you don't need to understand it to make games. Just understand that objects are another name for a values of data types that have methods. For example, all strings and lists are objects.

The split() method

The split method splits a string of words into a list of words.

```
Judges = ‘ Simon Sharon Danni Louis’
Judges = ‘ Simon Sharon Danni Louis’.split()
```

Will give us

```
Judges = [‘Simon’, ’Sharon’, ’Danni’, ’Louis’]
```