Python Beginner's Workshop

In Collaboration with the Pikes Peak Library District 21st Century Library

Ryan E. Freckleton

PySprings

2017-08-19

Outline

- Introduction
- First Steps
  - Running Python
  - Expressions
- Data Types
  - Strings
  - Lists
  - Dictionaries
- Libraries
  - Environments
  - Third-Party Packages
- Control Flow
  - Booleans
  - Looping and Branching

Conduct

- Treat everyone with the respect due their inherent dignity.
- All communication should be appropriate for a professional audience including people of many different backgrounds.
- Be kind to others. Make an environment conducive to learning. Behave professionally.
- Thank you for helping make this a welcoming, friendly event for all.
- Contact the organizers at`pysprings@pysprings.org` or`http://sayat.me/pysprings` (anonymous)

PySprings

Greetings

1. Your name
2. How did you get here?

Learning Goals

1-2-4-All

- What's one thing you know about programming in Python?
- What's one thing that you'd like to learn about programming in Python?

Learning Cycle

Introduction Short lecture introducing a new concept from Python

Exploration Hands-on application of the concept introduced. Work in groups and collaborate if you prefer! Explore the material in a hands-on manner

Invention What have we learned through our exploration? What surprises did we encounter? What mysteries did we uncovered?

Application With our newly"invented" knowledge, what can we do? Thisleads into a new exploration phase

Learning Cycle

Introduction Short lecture introducing a new concept from Python

Exploration Hands-on application of the concept introduced. Work in groups and collaborate if you prefer! Explore the material in a hands-on manner

Invention What have we learned through our exploration? What surprises did we encounter? What mysteries did we uncovered?

Application With our newly"invented" knowledge, what can we do? Thisleads into a new exploration phase

PySprings

Learning Cycle

Introduction Short lecture introducing a new concept from Python

Exploration Hands-on application of the concept introduced. Work in groups and collaborate if you prefer! Explore the material in a hands-on manner

Invention What have we learned through our exploration? What surprises did we encounter? What mysteries did we uncovered?

Application With our newly"invented" knowledge, what can we do? Thisleads into a new exploration phase

PySprings

Learning Cycle

Introduction Short lecture introducing a new concept from Python

Exploration Hands-on application of the concept introduced. Work in groups and collaborate if you prefer! Explore the material in a hands-on manner

Invention What have we learned through our exploration? What surprises did we encounter? What mysteries did we uncovered?

Application With our newly"invented" knowledge, what can we do? Thisleads into a new exploration phase

What is Programming?

- Programming is a creative activity
- It doesn't involve much math
- Programming is simply the act of entering instructions for the computer to perform

What is Programming?

- Programming is a creative activity
- It doesn't involve much math
- Programming is simply the act of entering instructions for the computer to perform

What is Programming?

- Programming is a creative activity
- It doesn't involve much math
- Programming is simply the act of entering instructions for the computer to perform

An Example

```
passwordFile = open('SecretPasswordFile.txt')
secretPassword = passwordFile.read()
print('Enter your password.')
typedPassword = input()
if typedPassword == secretPassword:
    print('Access granted')
    if typedPassword == '12345':
        print('That one is used on luggage.')
else:
    print('Access denied')
```

Outline

Introduction
First Steps
Running Python
Expressions
Data Types
Strings
Lists
Dictionaries
Libraries
Environments
Third-Party Packages
Control Flow
Booleans
Looping and Branching

Running Python

Example

run python with:

```
$ python3
```

enter the following into the interactive prompt:

```
>>> print("Hello, World!")
```

and

```
>>> import this
```

Installing Python

Let's create script.py now

```
print("Hello, World!")
```

and run it with

```
$ python3 script.py
```

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Notation

When you see an example like:

```
>>> print("Hello, World!")
```

it means to type that out in the interactive prompt. When you see an example like:

example.py

```
print("Hello, World!)
```

1

it means to type that out into a file, in this case, named *example.py*.

Outline

Introduction
First Steps
Running Python
Expressions
Data Types
Strings
Lists
Dictionaries
Libraries
Environments
Third-Party Packages
Control Flow
Booleans
Looping and Branching

Python as a Calculator

```
>>> 100 * 2
200
>>> (1 + 2 + 3 + 4 + 5 + 6) / 6
3.5
>>> 1 - 2*100 + 3*12
-163
>>> abs(-163)
163
```

Python Math Operations

Operators:

- `+ - * /`
- `% ** //`
- Does python obey order of operations?

Functions:

- **`abs bin hex oct ord round`**
- **`divmod min max pow`**
- What's the difference between these two lists of functions?

Python Math Operations

Operators:

- `+ - * /`
- `% ** //`
- Does python obey order of operations?

Functions:

- `abs bin hex oct ord round`
- `divmod min max pow`
- What's the difference between these two lists of functions?

Python Math Operations

Operators:

- `+ - * /`
- `% ** //`
- Does python obey order of operations?

Functions:

- `abs bin hex oct ord round`
- `divmod min max pow`
- What's the difference between these two lists of functions?

Python Math Operations

Operators:

- `+ - * /`
- `% ** //`
- Does python obey order of operations?

Functions:

- `abs bin hex oct ord round`
- `divmod min max pow`
- What's the difference between these two lists of functions?

Python Math Operations

Operators:

- `+ - * /`
- `% ** //`
- Does python obey order of operations?

Functions:

- `abs bin hex oct ord round`
- `divmod min max pow`
- What's the difference between these two lists of functions?

Python Math Operations

Operators:

- `+ - * /`
- `% ** //`
- Does python obey order of operations?

Functions:

- **`abs bin hex oct ord round`**
- **`divmod min max pow`**
- What's the difference between these two lists of functions?

Python Math Operations

Operators:

- `+ - * /`
- `% ** //`
- Does python obey order of operations?

Functions:

- **`abs bin hex oct ord round`**
- **`divmod min max pow`**
- What's the difference between these two lists of functions?

Python Math Operations

Operators:

- `+ - * /`
- `% ** //`
- Does python obey order of operations?

Functions:

- **`abs bin hex oct ord round`**
- **`divmod min max pow`**
- What's the difference between these two lists of functions?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Functions

```
def hello():
    print('Howdy!')
    print('Howdy!!!')
    print('Hello there.')

hello()
hello()
hello()
```

Functions

```
def hello(name):
    print('Hello ' + name)

hello('Alice')
hello('Bob')
```

Functions

```
def add(a, b):
      return a + b

print(add(1,2))
print(add(1,2) + add(3,4))
```

Outline

Introduction
First Steps
Running Python
Expressions
Data Types
Strings
Lists
Dictionaries
Libraries
Environments
Third-Party Packages
Control Flow
Booleans
Looping and Branching

Strings

Examples

```
"This is a string."
'This is also a string.'
"This is 'a' string"
'This is "a" string'
```

We can also get more information from python:

```
>>> help(str)
```

Strings

Examples

```
"This is a string."
'This is also a string.'
"This is 'a' string"
'This is "a" string'
```

We can also get more information from python:

```
>>> help(str)
```

Strings

Examples

```
"This is a string."
'This is also a string.'
"This is 'a' string"
'This is "a" string'
```

We can also get more information from python:

```
>>> help(str)
```

Strings

More Examples

```
>>> 'this is a string'.title()
'This Is A String'
>>> 'this is a string'.upper()
'THIS IS A STRING'
>>> 'what ARE you doing!?'.lower()
'what are you doing!?'
>>> "   there's whitespace in this   ".strip()
"there's whitespace in this string."
```

Hello again

hello.py

```
name = input('What is your name? ')
print('Hello, ' + name + '!')
```

let's try it!

```
$ python3 hello.py
```

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Indexing

```
>>> s = 'We are the Knights who say ni!'
>>> s[0]
'W'
>>> s[-1]
'!'
>>> s[7:10]
'the'
>>> s[-7:-4]
'say'
```

Indexing

```
+---+---+---+---+---+---+
| P | y | t | h | o | n |
+---+---+---+---+---+---+
0   1   2   3   4   5   6
-6  -5  -4  -3  -2  -1
```

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Outline

- Introduction
- First Steps
  - Running Python
  - Expressions
- Data Types
  - Strings
  - Lists
  - Dictionaries
- Libraries
  - Environments
  - Third-Party Packages
- Control Flow
  - Booleans
  - Looping and Branching

Lists

```
>>> mylist = [1, 2, 'three', "4", 5.3]
>>> s = "What are the words in this string?"
>>> s.split()
['What', 'are', 'the', 'words', 'in', 'this',
   'string?']
>>> words = s.split()
>>> words.sort()
>>> words
['What', 'are', 'in', 'string?', 'the', 'this'
   , 'words']
```

Lists

What are the methods of list?

Remember:

```
>>> help(list)
```

and try out:

```
>>> dir(list)
```

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Outline

Introduction
First Steps
Running Python
Expressions
Data Types
Strings
Lists
Dictionaries
Libraries
Environments
Third-Party Packages
Control Flow
Booleans
Looping and Branching

Dictionaries

```
>>> myCat = {'size': 'fat', 'color': 'gray',
...          'disposition': 'loud'}
>>> myCat['size']
'fat'
>>> 'My cat has ' + myCat['color'] + ' fur.'
'My cat has gray fur.'
```

Dictionaries

What are the methods of list?

Remember:

```
>>> help(dict)
>>> dir(dict)
```

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Outline

- Introduction
- First Steps
  - Running Python
  - Expressions
- Data Types
  - Strings
  - Lists
  - Dictionaries
- Libraries
  - Environments
  - Third-Party Packages
- Control Flow
  - Booleans
  - Looping and Branching

Environments

virtualenv

```
$ virtualenv raindrop
$ . raindrop/source/bin/activate # Linux and OSX
$ randrop\Scripts\activate # Windows
```

Outline

- Introduction
- First Steps
  - Running Python
  - Expressions
- Data Types
  - Strings
  - Lists
  - Dictionaries
- Libraries
  - Environments
  - Third-Party Packages
- Control Flow
  - Booleans
  - Looping and Branching

Installing Third-Party Packages

```
$ pip3 install requests
```

Finding Third-Party Packages

`http://pypi.python.org`

Requests Example

requests_script.py

```
import requests                                          1
resp = requests.get('http://httpbin.org/ip')             2
print(resp.json())                                       3
```

Outline

- Introduction
- First Steps
  - Running Python
  - Expressions
- Data Types
  - Strings
  - Lists
  - Dictionaries
- Libraries
  - Environments
  - Third-Party Packages
- Control Flow
  - Booleans
  - Looping and Branching

Booleans

```
>>> bool(1)
True
>>> bool(0)
False
>>> bool("no")
True
>>> bool("")
False
>>> bool([])
False
>>> bool([42])
True
```

Outline

- Introduction
- First Steps
  - Running Python
  - Expressions
- Data Types
  - Strings
  - Lists
  - Dictionaries
- Libraries
  - Environments
  - Third-Party Packages
- Control Flow
  - Booleans
  - Looping and Branching

Looping and Branching

```
>>> words = 'this is a list of words'.split()
>>> for word in words:
...     print(word.title())
...
This
Is
A
List
Of
Words
```

Looping and Branching

```
password = input(
    "Enter the secret word: "
)
if password == "sesame":
    print("Access granted.")
else:
    print("Access denied!")
```

Boolean operators:

- == != <= >= > < in

Looping and Branching

```
password = input(
    "Enter the secret word: "
)
if password == "sesame":
    print("Access granted.")
else:
    print("Access denied!")
```

Boolean operators:

- `==` `!=` `<=` `>=` `>` `<` `in`

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Looping and Branching

While Loop

```
while True:
    password = input("Enter the secret word: ")
    if password == "sesame":
        print("Access granted.")
        break
    else:
        print("Access denied!")
```

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Invention

- What problems, if any, did you encounter?
- What mysteries, if any, did you encounter?
- What other take-aways are there from this session, what could you use from it in the future?

Looping and Branching

elif

```
age = int(input("How old are you? "))
if age < 18:
    print("You're not old enough dance.")
elif age == 18:
    print("Welcome, is it your first time here?")
else:
    print("You can dance if you want to, you can le
```

Practice Problems

- Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.
- Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.
- Write a function named collatz() that has one parameter named number. If number is even, then collatz() should print number // 2 and return this value. If number is odd, then collatz() should print and return 3 * number + 1.

PySprings

Practice Problems

- Say you have a list value like this:

  ```
  spam = ['apples', 'bananas', 'tofu', 'cats']
  ```

  Write a function that takes a list value as an argument and returns a string with all the items separated by a comma and a space, with and inserted before the last item. For example, passing the previous spam list to the function would return 'apples, bananas, tofu, and cats'. But your function should be able to work with any list value passed to it.

Practice Problems

- You are creating a fantasy video game. The data structure to model the player's inventory will be a dictionary where the keys are string values describing the item in the inventory and the value is an integer value detailing how many of that item the player has. For example, the dictionary value

```
{'rope': 1, 'torch': 6, 'gold coin': 42,
 'dagger': 1, 'arrow': 12}
```

  means the player has 1 rope, 6 torches, 42 gold coins, and so on.

Practice Problems

Write a function named displayInventory() that would take any possible "inventory" and display it like the following:

```
Inventory:
12 arrow
42 gold coin
1 rope
6 torch
1 dagger
Total number of items: 62
```

Conclusion

- Final Takeaways (1-2-4-all)
- Survey`https://goo.gl/forms/ZpNl0z8pw5J8J8Rv1`
- Feedback –`http://sayat.me/pysprings`
- Based on`https://automatetheboringstuff.com/` released under 

PySprings

Projects!

- Daily Programmer`https://www.reddit.com/r/dailyprogrammer/`
  - Game of Threes
- WordPlay`https://github.com/jesstess/Wordplay`
- Colorwall`https://github.com/jesstess/ColorWall`

PySprings