# Python Beginner's Workshop In Collaboration with the Pikes Peak Library District 21st Century Library

Ryan E. Freckleton

PySprings

2016-09-10



Libraries

Control Flow

Introduction

Outline

Introduction First Steps First Steps

#### Ground Rules

Introduction

#### http:

//pysprings.org/pages/code-of-conduct.html

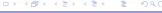
- ▶ We do not tolerate harassment of participants in any form.
- All communication should be appropriate for a professional audience including people of many different backgrounds.
- ▶ Be kind to others. Do not insult or put down other attendees. Behave professionally.
- ▶ Thank you for helping make this a welcoming, friendly event for all.
- Contact the organizers at pysprings@pysprings.org or http://sayat.me/pysprings (anonymous)

### Greetings

- 1. "How did I get here? What is my prior experience in Python and Programming (if any)?"
- 2. "What do I want to have happen?" (write on mindmap)
- 3. "What expertise or experience do I have that others here might find useful?"



**PySprings** 



Beginning Python

Libraries

### Learning Goals

Introduction

- ► Virtual environments and installing/using third-party packages
- ► Python Basics
  - Lists
  - Dictionarie

First Steps

► Control Flow





Libraries

### Learning Goals

Introduction

- ► Virtual environments and installing/using third-party packages
  - Python Basics
    - Licto
    - Dictionarie

First Steps

► Control Flow





Libraries

### Learning Goals

Introduction

- ► Virtual environments and installing/using third-party packages
- Python Basics
  - Lists
  - Dictionaries

First Steps

Control Flow





Libraries

Work in groups and collaborate if you prefer!
the material in a hands-on manner
Invention What have we learned through our exploration

surprises did we encounter? What mysteries did we uncovered?

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



Introduction

First Steps

Libraries

Control Flow

### Learning Cycle

First Steps

Introduction

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

Introduction Short lecture introducing a new concept from Python

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



Libraries

### What is Programming?

First Steps

Programming is simply the act of entering instructions for the



Control Flow

Introduction

Libraries

What is Programming?

Introduction

It doesn't involve much math

First Steps

▶ Programming is simply the act of entering instructions for the



Libraries

First Steps

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



Control Flow



Introduction

### An Example

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

10

Beginning Python PySprings

Running Python
Expressions
Data Types
Strings
Lists
Dictionaries
Libraries
Environments

Data Types

Libraries

Control Flow

**PySprings** 

Introduction

Running Python

Outline

Beginning Python

First Steps

First Steps

•0000

enter the following into the interactive prompt:

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

Data Types

Libraries

>>> **import** this

and

Introduction

Running Python

Running Python

python

First Steps

00000

Libraries

print("Hello, World!")

Introduction

Running Python

and run it with

\$ python script.py

Let's create script.py now

First Steps

00000



Libraries

▶ What problems, if any, did you encounter?

First Steps

00000

- 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?



Control Flow

Introduction

Running Python

Invention

Libraries

### Invention

Running Python

Introduction

▶ What problems, if any, did you encounter?

First Steps

00000

- ▶ What mysteries, if any, did you encounter?
- ▶ What other take-aways are there from this session, what could



### 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!")
```

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

**Pusprings** 

Expressions

Data Types
Strings
Lists
Dictionaries

Libraries
Environments
Third-Party Packages

Control Flow

Data Types

Libraries

Control Flow

**PySprings** 

Introduction

Expressions

Outline

Beginning Python

First Steps

First Steps

•000000

Libraries

Control Flow

```
Beginning Python
```

- 163

163

Introduction

Expressions

First Steps

000000

>>> 1 - 2\*100 + 3\*12

>>> **abs**(-163)

Python as a Calculator

Libraries

### Does python obey order of operations?

Introduction

Expressions

- ▶ abs bin hex oct ord round
- ► divmod min max pow

First Steps

0000000

Python Math Operations

► What's the difference between these two lists of functions?



Libraries

First Steps

0000000

- **▶** % \*\* //
- ▶ Does python obey order of operations?

#### Functions:

Introduction

Expressions

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



Libraries

Python Math Operations

Introduction

Expressions

- **>** % \*\* //

#### Functions:

- ▶ abs bin hex oct ord round
- divmod min max pow

First Steps

0000000

▶ What's the difference between these two lists of functions?



Libraries

```
Operators:
```

Introduction

Expressions

- **▶** + \*
- **>** % \*\* //
- Does python obey order of operations?

#### Functions:

- ▶ abs bin hex oct ord round
- ► divmod min max pow

First Steps

0000000

▶ What's the difference between these two lists of functions?



Libraries

```
Operators:
```

Introduction

Expressions

- **▶** + \*
- **>** % \*\* //
- Does python obey order of operations?

#### Functions:

- ▶ abs bin hex oct ord round
- ► divmod min max pow

First Steps

0000000

▶ What's the difference between these two lists of functions?



Libraries

### Operators:

Introduction

Expressions

- **L** L +
- **>** % \*\* //
- ► Does python obey order of operations?

#### Functions:

- ▶ abs bin hex oct ord round
- ► divmod min max pow

First Steps

0000000

▶ What's the difference between these two lists of functions?



Libraries

### Operators:

- **>** % \*\* //
- ► Does python obey order of operations?

#### Functions:

Introduction

Expressions

- ▶ abs bin hex oct ord round
- ▶ divmod min max pow

First Steps

0000000

▶ What's the difference between these two lists of functions?



Libraries

### Python Math Operations

First Steps

#### Operators:

Introduction

- **L** \_ J
- **»** \*\* //
- ▶ 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?



Libraries

Invention

Introduction

Expressions

▶ What problems, if any, did you encounter?

First Steps

0000000

- ▶ What other take-aways are there from this session, what could

Libraries

## Invention

Introduction

Expressions

What problems, if any, did you encounter?

First Steps

0000000

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

Libraries

Introduction

### 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?



Libraries

**Functions** 

Introduction

Expressions

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

First Steps

0000000



Control Flow

5

hello()

Libraries

```
print 'Hello ' + name
hello('Alice')
hello('Bob')
```



Control Flow

Introduction

Expressions

**Functions** 

First Steps

0000000

def hello(name):

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

print(add(1,2))

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

First Steps

000000

Libraries

Control Flow





Introduction

Expressions

**Functions** 

Data Types Strings

Data Types

•0000000

Libraries

Control Flow

**PySprings** 

Introduction

Outline

Beginning Python

Strings

First Steps

0000000

Libraries

```
'This is "a" strin
We can also get more inf
```

>>> help(str)

First Steps

Introduction

Strings

Strings

0000000

Libraries

```
PuŠpring:
```

>>> help(str)

Introduction

Strings

Strings

First Steps

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

0000000

Libraries

```
"This is 'a' string"
'This is "a" string'
```

First Steps

'This is also a string.'

We can also get more information from python:

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



Control Flow

Introduction

Strings

Strings

Libraries

```
'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."
```

Control Flow

Introduction

First Steps

>>> 'this is a string'.title()

00000000

Libraries

```
hello.py
```

Introduction

Strings

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

let's try it!

\$ python hello.py

First Steps



00000000

Libraries

▶ What problems, if any, did you encounter?

First Steps

- ▶ 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?



Control Flow

Introduction

Invention

Strings

00000000

Libraries

Introduction

Invention

Strings

First Steps

- ▶ What mysteries, if any, did you encounter?
- ▶ What other take-aways are there from this session, what could



## 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?



>>> s = 'We are the Knights who say ni!'

Data Types

00000000

Libraries

Control Flow

**PySprings** 

Introduction

Indexing

>>> s[7:10]

>>> s[-7:-4]

'the'

'say'

Beginning Python

Strings

First Steps

00000000

Libraries

Control Flow

**PySprings** 

Introduction

Indexing

Beginning Python

Strings

First Steps

0000000

Libraries

# Invention

Introduction

Strings

▶ What problems, if any, did you encounter?

First Steps

- 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?



0000000

Libraries

▶ What problems, if any, did you encounter?

First Steps

- ▶ What mysteries, if any, did you encounter?
- ▶ What other take-aways are there from this session, what could



Control Flow

Introduction

Invention

Strings

# 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?



Data Types Lists

Data Types

0000

Libraries

Control Flow

**PySprings** 

Introduction

Outline

Beginning Python

Lists

First Steps

Libraries

#### Lists

Introduction

```
>>> 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'l
```

0000

Libraries

Introduction

Lists

Lists

What are the methods of list?

>>> dir(list)

First Steps

000

Libraries

Introduction

Invention

Lists

- ▶ What problems, if any, did you encounter?

First Steps

▶ What other take-aways are there from this session, what could



000

Libraries

## Invention

Introduction

Lists

First Steps

- ▶ What mysteries, if any, did you encounter?
- ▶ What other take-aways are there from this session, what could



Libraries

## Invention

Introduction

- ▶ 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?



Expressions

Data Types

Strings
Lists

Dictionaries

Libraries

Environments

Data Types

0000

Libraries

Control Flow

**PySprings** 

Introduction

Dictionaries

Outline

Beginning Python

First Steps

Libraries

# Dictionaries

First Steps

Introduction

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

0000

Libraries

Remember:

Introduction

Dictionaries

**Dictionaries** 

>>> help(dict)

First Steps

000

Libraries

# Invention

Introduction

Dictionaries

▶ What problems, if any, did you encounter?

First Steps

- 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?



000

Libraries

## Invention

Introduction

Dictionaries

▶ What problems, if any, did you encounter?

First Steps

- ▶ What mysteries, if any, did you encounter?
- ▶ What other take-aways are there from this session, what could



Libraries

### Invention

Introduction

- ▶ 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?



Running Python
Expressions

Data Types
Strings
Lists
Dictionaries

Libraries
Environments

Data Types

Libraries

•0

Control Flow

**PySprings** 

Introduction

Environments

Outline

Beginning Python

First Steps

Libraries

0

## virtualenv

Introduction

Environments

virtualenv raindrop

First Steps

- . raindrop/source/bin/activate # Linux and OSX
- randrop\Scripts\activate # Windows



Libraries

•000

Control Flow

Libraries

Introduction

Outline

Third-Party Packages

First Steps

Third-Party Packages

Libraries

0000

\$ pip install requests

First Steps

Installing Third-Party Packages



Control Flow

Introduction

Third-Party Packages

Libraries

0000

Finding Third-Party Packages

First Steps

http://pypi.python.org



Control Flow

Introduction

Third-Party Packages

# requests script.py

Requests Example

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



Expressions

Data Types
Strings
Lists
Dictionaries
Libraries

Data Types

Libraries

Control Flow

**PySprings** 

Introduction

**Booleans** 

Outline

Control Flow Booleans

Beginning Python

First Steps

Libraries

Introduction

**Booleans** 

Booleans

True

False

False

>>> bool("")

>>> bool([])

>>> **bool**([42])

First Steps

Running Python
Expressions
Data Types
Strings
Lists
Dictionaries
Libraries
Environments

Data Types

Libraries

Control Flow

**PySprings** 

Introduction

Outline

Looping and Branching

Control Flow

Beginning Python

Looping and Branching

First Steps

Libraries

```
Looping and Branching
```

First Steps

```
print(word.title())
```

Introduction

Looping and Branching



password = input("Enter the secret word: ")

Data Types

Libraries

```
if password == "sesame":
    print("Access granted.")
else:
    print("Access denied!")
```

Boolean operators

```
▶ == != <= >= > < in
```

First Steps



Control Flow

00000000000

Introduction

Looping and Branching

password = input("Enter the secret word: ")

Data Types

Libraries

```
else:
    print("Access denied!")
Boolean operators:
```

if password == "sesame":

print("Access granted.")

First Steps

```
▶ == != <= >= > < in
```



Control Flow

00000000000

Introduction

Looping and Branching

Libraries

# Invention

Looping and Branching

Introduction

▶ What problems, if any, did you encounter?

First Steps

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

Libraries

What problems, if any, did you encounter?

First Steps

- ► 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?



Control Flow

Introduction

Looping and Branching

Invention

Libraries

Data Types

### Invention

Introduction

► What problems, if any, did you encounter?

First Steps

- ▶ 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?



Libraries

# Looping and Branching While Loop

First Steps

Introduction

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



Libraries

First Steps

- ▶ What problems, if any, did you encounter?
- ▶ What other take-aways are there from this session, what could



Control Flow 000000000000

Introduction

Looping and Branching

Invention

Libraries

First Steps

- ▶ 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



Control Flow 000000000000

Introduction

Looping and Branching

Invention

Libraries

Data Types

# Invention

Introduction

► What problems, if any, did you encounter?

First Steps

- ► 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?



Beginning Python

Introduction

#### 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.



PvSprings

Looping and Branching

#### 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.



Beginning Python PySprings

Looping and Branching

Introduction

#### 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.



Beginning Python PySprings

# 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

Looping and Branching

- ► Final Takeaways
- Survey https://goo.gl/forms/ZpNl0z8pw5J8J8Rv1
- Feedback http://sayat.me/pysprings
- ► Based on https://automatetheboringstuff.com/released under



Looping and Branching

## Projects!

Beginning Python

Introduction

- ► Colorwall https://github.com/jesstess/ColorWall
- WordPlay https://github.com/jesstess/Wordplay
- ► Twitter https://github.com/jesstess/TwitterAPI
- General Projects Page http://wiki.openhatch.org/ Boston\_Python\_Workshop\_6/Saturday\_projects
- Daily Programmer https://www.reddit.com/r/dailyprogrammer/



PvSprings