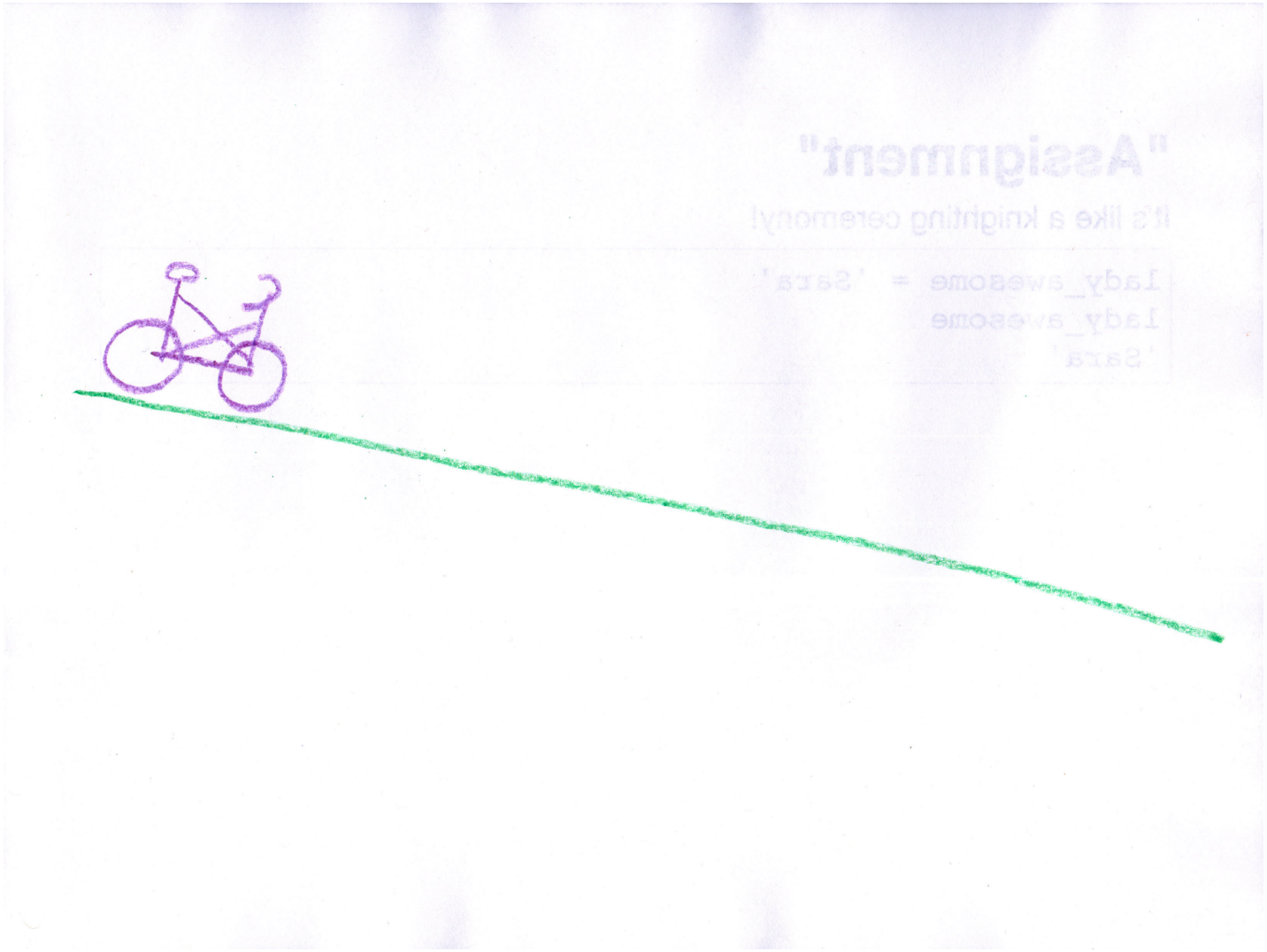
Python: Over the Bumps

@catherinedevlin

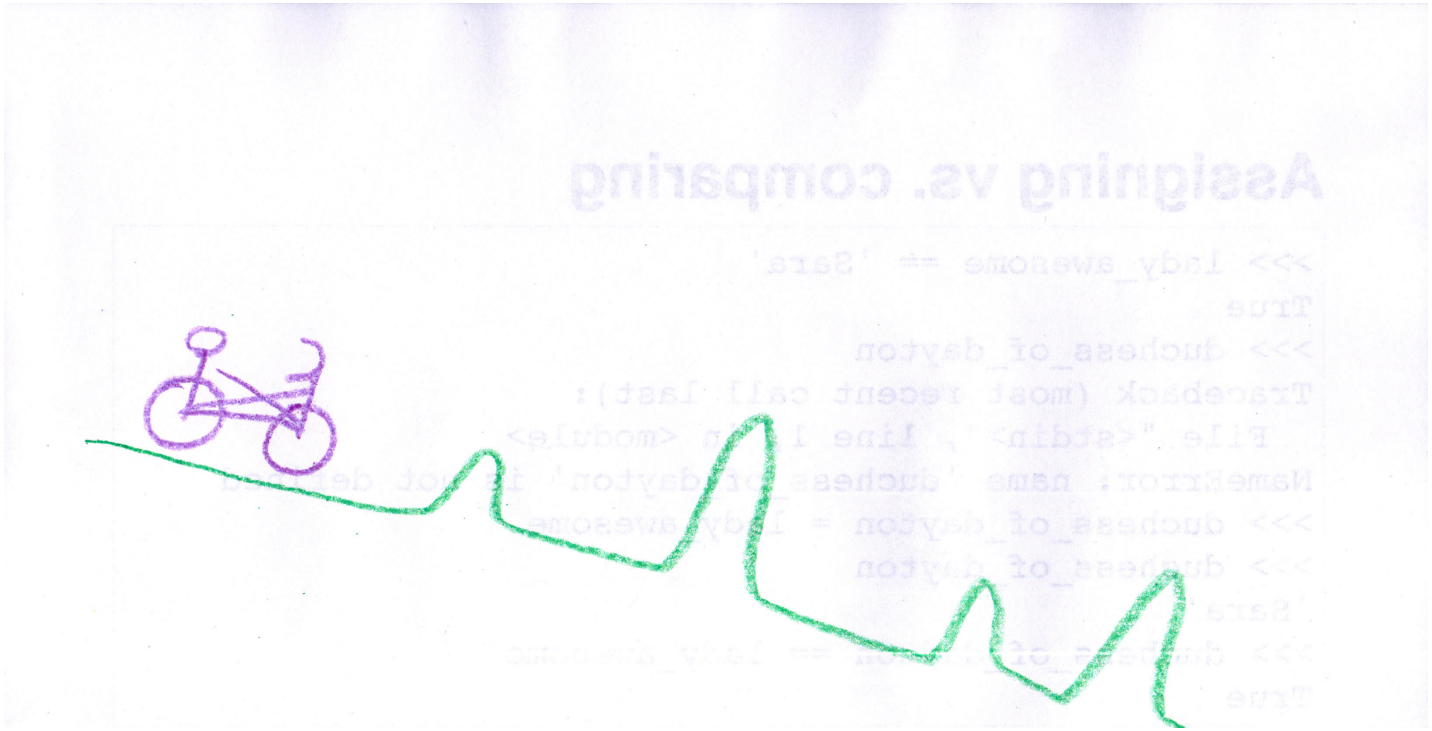
PyOhio 2016

github.com/catherinedevlin/python-over-the-bumps

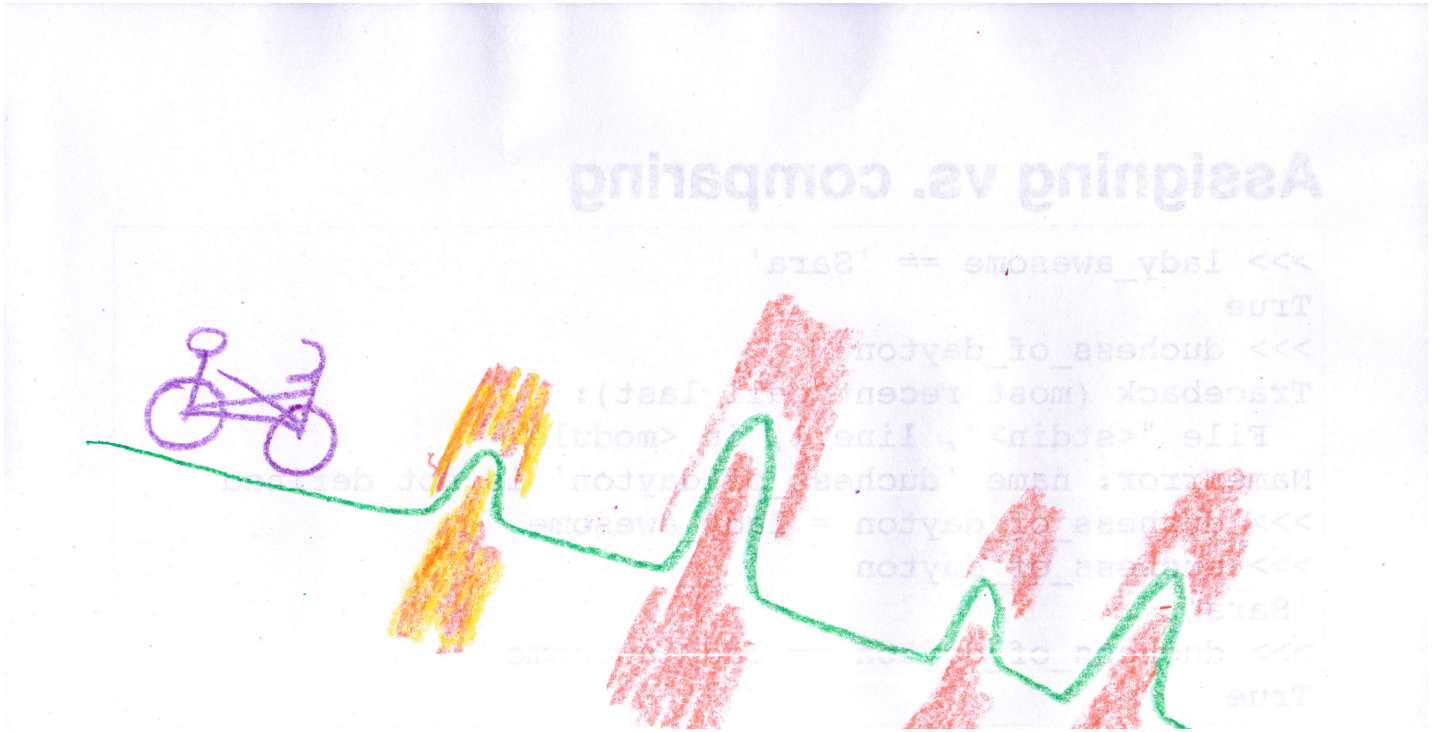
The Vision



The Reality



The Solution (?)



2? 3?

3

Installation

- python.org
- Anaconda
- PythonAnywhere.com

Python environments

- REPL: Improv comedy
- Double-click on a script (but don't)
- Run "script" from command line
- Special environments: Jupyter, IDLE, embedded in ESRI or game...

"Variables"

Why do we have names?

Variable: a name for a piece of data

"Assignment"

It's like a knighting ceremony!

```
>>> lady_buckeye = 'Sara'
>>> lady_buckeye
'Sara'
```

Assigning vs. comparing

```
>>> lady_buckeye = 'Sara'
>>> lady_buckeye == 'Sara'
True
>>> duchess_of_dayton
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'duchess_of_dayton' is not defined
>>> duchess_of_dayton = lady_buckeye
>>> duchess_of_dayton
'Sara'
>>> duchess_of_dayton == lady_buckeye
True
```

"Loop" variables

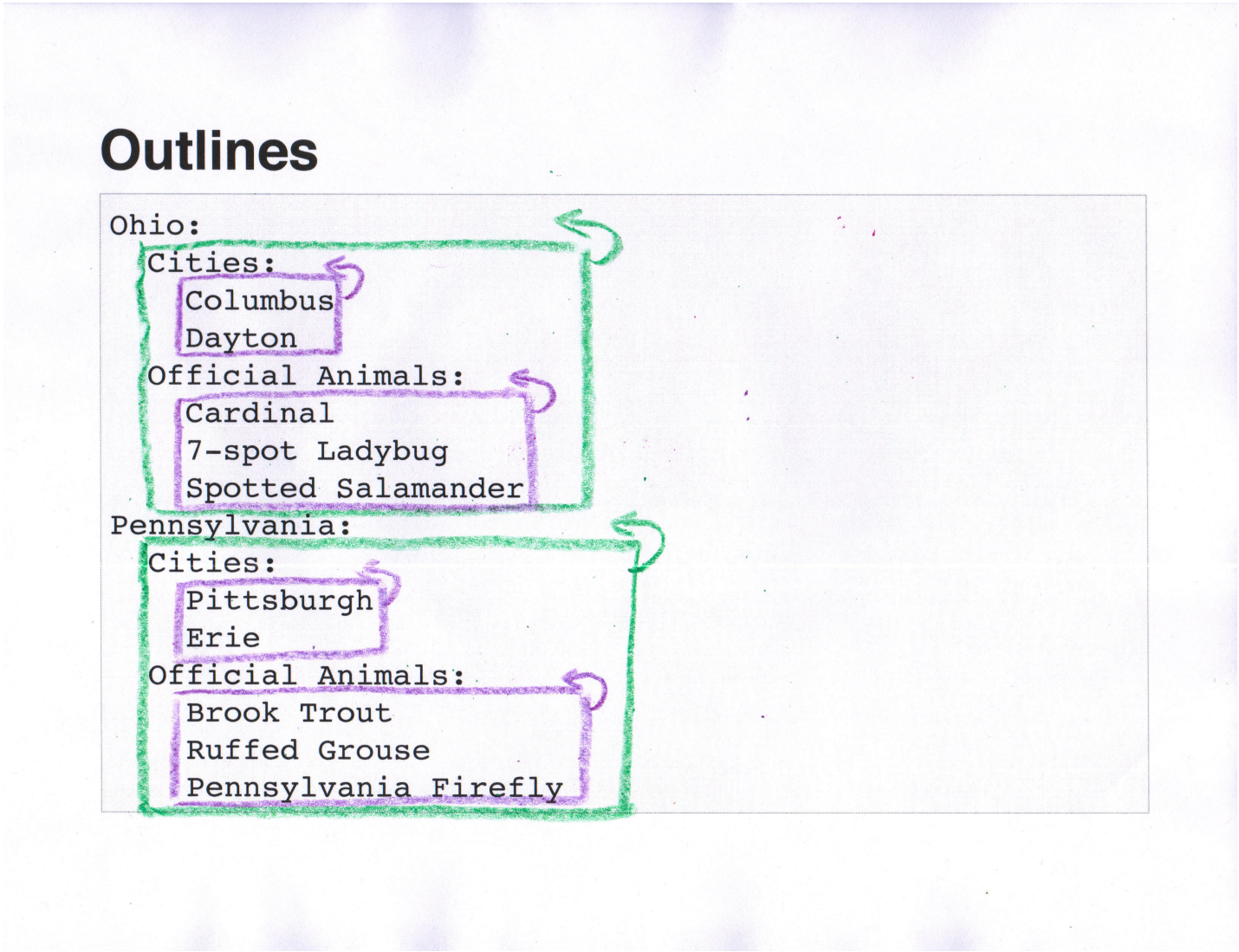
```
for runner in ['Sara', 'KC', 'Holly']:
    print('Go, ' + runner + ', go!')

Go, Sara, go!
Go, KC, go!
Go, Holly, go!
```

Outlines

```
Ohio:
  Cities:
    Columbus
    Dayton
  Official Animals:
    Cardinal
    7-spot Ladybug
    Spotted Salamander
Pennsylvania:
  Cities:
    Pittsburgh
    Erie
  Official Animals:
    Brook Trout
    Ruffed Grouse
    Pennsylvania Firefly
```

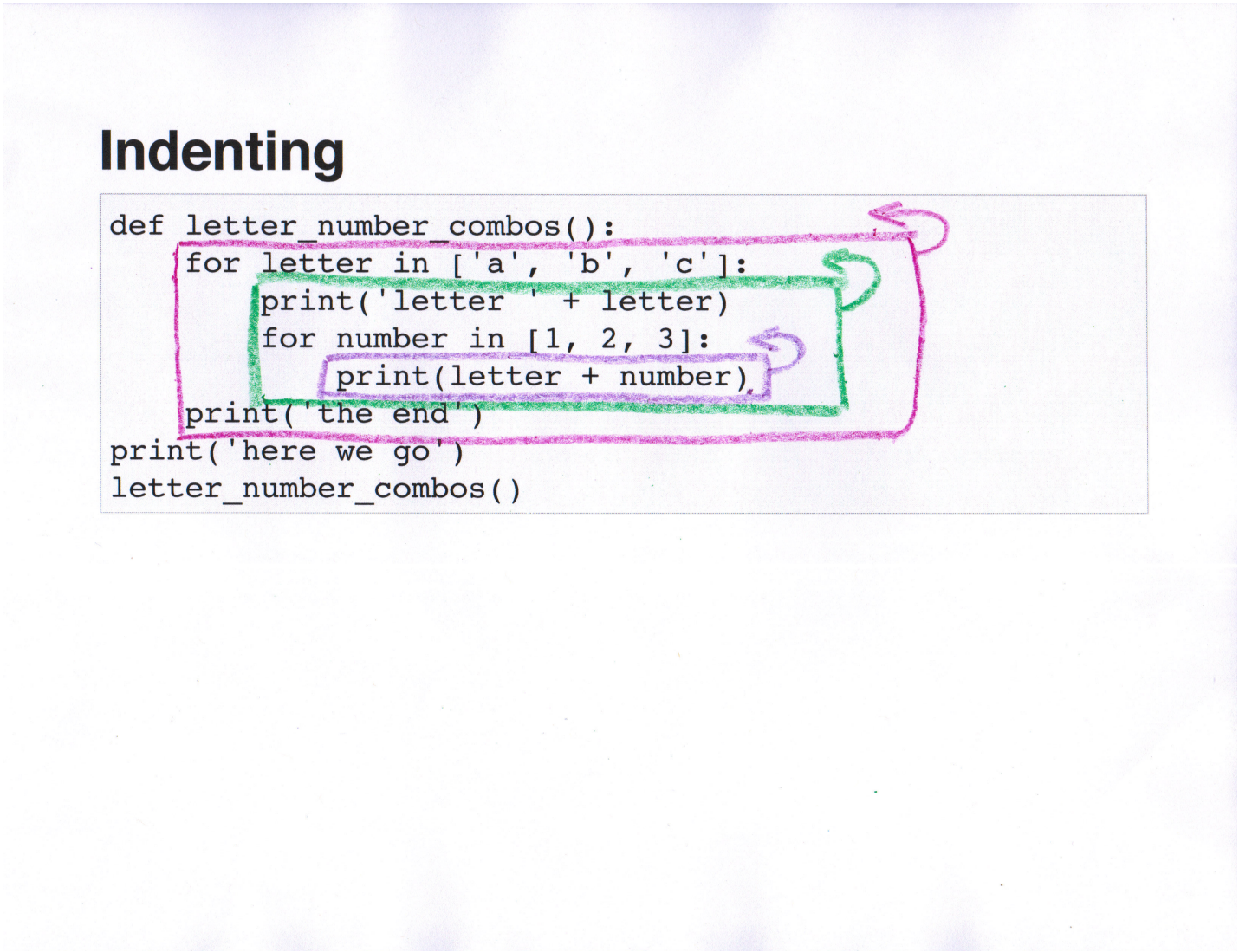
Outline


Outlines
Ohio:
Cities:
Columbus
Dayton
Official Animals:
Cardinal
7-spot Ladybug
Spotted Salamander
Pennsylvania:
Cities:
Pittsburgh
Erie
Official Animals:
Brook Trout
Ruffed Grouse
Pennsylvania Firefly

Indenting

```
def letter_number_combos():
    for letter in ['a', 'b', 'c']:
        print('letter ' + letter)
        for number in [1, 2, 3]:
            print(letter + number)
    print('the end')
print('here we go')
letter_number_combos()
```

Indenting


Indenting
def letter_number_combos():
for letter in ['a', 'b', 'c']:
print('letter ' + letter)
for number in [1, 2, 3]:
print(letter + number)
print('the end')
print('here we go')
letter_number_combos()

"Functions"

```
def taco_salad(chips, beans, cheese, salsa):
    salad = chips + ', ' \
        + beans + ', ' \
        + cheese + ', ' \
        + salsa
    return salad

>>> taco_salad('blue corn chips', 'black beans',
...     'monterrey jack', 'mild salsa')

'blue corn chips, black beans, monterrey jack, mild salsa'
```

With different data

```
def taco_salad(chips, beans, cheese, salsa):
    salad = chips + ', ' \
        + beans + ', ' \
        + cheese + ', ' \
        + salsa
    return salad

>>> taco_salad('white corn chips', 'pinto beans',
...     'Velveeta', 'hot salsa')

'white corn chips, pinto beans, Velveeta, hot salsa'
```

Function "arguments"

```
def taco_salad(chips, beans, cheese, salsa):
    salad = chips + ', ' \
        + beans + ', ' \
        + cheese + ', ' \
        + salsa
    return salad

my_chips = 'blue corn chips'
some_beans = 'black beans'
good_cheese = 'Monterrey jack'
the_salsa = 'mild salsa'

taco_salad(my_chips, some_beans, good_cheese, the_salsa)
```

Multiple choice: GREEN

```
flavor = 'moose tracks'
top_with = 'caramel'

def sundae(ice_cream, topping):
    return (ice_cream + ' + ' + topping)

sundae(flavor, top_with)
```

Multiple choice: BROWN

```
flavor = 'moose tracks'
top_with = 'caramel'

def sundae('moose tracks', 'caramel'):
    return (ice_cream + ' + ' + topping)

sundae(flavor, top_with)
```

Multiple choice: ORANGE

```
flavor = 'moose tracks'
top_with = 'caramel'

def sundae(ice_cream, topping):
    return (ice_cream + ' + ' + topping)

sundae(ice_cream, topping)
```

Multiple choice: BLUE

```
flavor = 'moose tracks'
top_with = 'caramel'

def sundae(ice_cream, topping):
    return (ice_cream + ' + ' + topping)

sundae('flavor', 'top_with')
```

No arguments

```
def favorite():
    return 'chocolate chip'

>>> favorite()
'chocolate chip'
```

Call with *()*

```
>>> favorite
<function favorite at 0x10b5010c8>
```

```
>>> favorite()
'chocolate chip'
```

No return

```
def make_yourself_a_sandwich(bread, filling):
    sandwich = bread + filling + bread

>>> make_yourself_a_sandwich('whole wheat', 'tuna salad')
>>> sandwich
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sandwich' is not defined
```

"Assign" the return

```
def make_me_a_sandwich(bread, filling):
    sandwich = bread + ', ' + filling + ', ' + bread
    return sandwich

>>> my_sandwich = make_me_a_sandwich('whole wheat', 'tuna sa
>>> my_sandwich
'whole wheat, tuna salad, whole wheat'
```

"Modules"

```
name,species,kg,notes
Alfred,wart hog,22,loves turnips
Gertrude,polar bear,312.7,deep thinker
Emily,salamander,0.3,
```

Getting the eggbeater

Store to kitchen: "*install*"

Kitchen to countertop: "*import*"

Missing import

```
>>> animal_data = csv.reader('animals.csv')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'csv' is not defined
>>> import csv
>>> animal_data = csv.reader('animals.csv')
```

csv is not on the countertop

Missing install

```
>>> pandas.read_csv('animals.csv')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pandas' is not defined
>>> import pandas
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named pandas
```

Going to the store

```
$ pip install pandas
Collecting pandas
...
$ python
>>> import pandas
>>> animal_data = pandas.read_csv('animals.csv')
```

Huh?

```
pip install pandas
```

pip

Pip Installs Packages

pandas

Name of package

PyPI

the "store"

Advice

- Hitchhikers' Guide to Python
- help@python.org, reddit
- websearch for blogs
- User community

"virtualenv"

image: Joergelman @ pixabay

pyvenv: a kitchen-builder

```
$ pyvenv ~/virtualenvs/pyohio
$ . ~/virtualenvs/pyohio/bin/activate
$ pip install requests
$ python
>>> import requests
>>> exit()
$ rm -rf . ~/virtualenvs/pyohio/bin/activate
```

Git

- Version control tool
- Install `git`
- What's a "repo"?
- GitHub
- `git clone <repo URL>`
- GitKraken

Projects

- Arcade game
- Office automation
- Web scraping
- Web application
- Raspberry Pi
- Analytic math: SAGE
- Numeric math: Pandas
- Text analysis
- Phone programming
- Bots!

Getting help and getting connected

- Beginners' Guide
- Hitchhikers' Guide
- Community!
  - help@python.org
  - tutor list
  - PyOhio
  - Python User Groups

Full-scale tutorials

- Udemy course
- CodeAcademy
- Invent with Python books
- http://learnpython.org/

Do not fear!

@catherinedevlin

PyOhio 2016

github.com/catherinedevlin/python-over-the-bumps