

# Introduction to Python

Jordan Boyd-Graber

University of Maryland

August 26, 2024

# Using Python

- Command line
- Notebooks
- VM

## Start Python, Variables

```
hk:~ jbg$ python3
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 26 2016, 10:47:25)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type `help`, `copyright`, `credits` or `license` for
>>> 3 + 2
5
>>> a = 5
>>> a * 5
25
```

## Define Function

```
>>> def logistic(x):  
...     return 1.0 / (1 + exp(-x))  
...  
>>>
```

## Errors!

```
>>> logistic(3.0)
```

```
Traceback (most recent call last):
```

```
File <stdin>', line 1, in <module>
```

```
File <stdin>', line 2, in logistic
```

```
NameError: name 'exp' is not defined
```

```
>>> from math import exp
```

## Importing Functions

```
>>> logistic(-100)
3.7200759760208356e-44
>>> logistic(100)
1.0
>>> logistic(0)
0.5
```

# Python: Datatypes

Jordan Boyd-Graber

University of Maryland

August 26, 2024

# Datatypes

Programming is about manipulating data. Different types of informations are stored differently.

- integers
- floats
- strings
- lists
- dictionaries.



# Datatypes

```
>>> 22//7
```

```
3
```

```
>>> s = 22/7
```

```
>>> s
```

```
3.1428571428571428
```

```
>>> 22 % 7
```

```
1
```

# Strings

```
>>> s = " I am the very model of a modern Major-General "  
>>> s.strip()  
'I am the very model of a modern Major-General'  
>>> s.find("am")  
3  
>>> s.replace("modern Major-General", "Gilbert caricature")  
' I am the very model of a Gilbert caricature '
```



# Dictionaries

```
>>> d = {}
>>> d[3] = 4
>>> d[3]
4
>>> d[2]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 2
>>> d[3] = 2
>>> d[3]
2
```

# Putting it Together

- Functions, classes

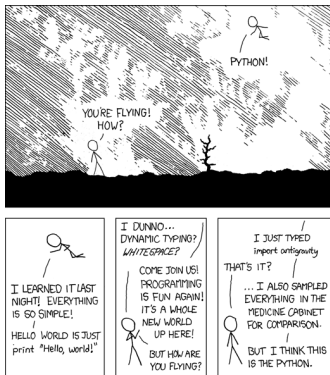
# Python: Control Flow

Jordan Boyd-Graber

University of Maryland

August 26, 2024

# What Makes Python Odd



- Some languages use brackets / parens
- Python uses whitespace
- Loops / if statements

## Is this the right language to start with?



- Python-inspired syntax
- `where` clause has unit tests for function
- Concise, recursive (but optional) data declarations



## If statements

```
>>> sheeps_clothing = "wool"
>>> if "wolf" in sheeps_clothing:
...     print("RUN")
... elif len(sheeps_clothing) > 4:
...     print("Haircut time")
... else:
...     print("All is well")
...
All is well
```

## For loops

Need to take a list as input

```
>>> sum = 0
>>> for i in range(100):
...     sum += i
...
>>> print(sum)
4950
```

## List comprehensions

```
>>> sum(x for x in range(10))
```

```
45
```

```
>>> sum(x for x in range(10) if x % 2 == 0)
```

```
20
```

## String formatting

```
>>> "%s foo %i %0.3f" % ("blah", 8, 0.5)
'blah foo 8 0.500'
```