



# Introduction to Python

21 February 2022 - 4 March 2022

Instructor: Lucy Havens

# Day 2 Recap

Conditionals

Loops and Iteration

Handling Errors

Functions

Methods

Modules

**Questions about  
assignment 2?**

# **Assignment 2**

# **Challenges**

# DAY 3

# Recursion

Solving a problem by repeatedly solving smaller versions of the same problem

# Recursion

Solving  
the same problem  
over and over again.

f the



**Let's code!**

# More Data Types: List

Ordered, mutable (changeable) collection of items of any data type

[1, -5, 3, 8, 11]    ["hi", "hey"]    [True]    [3, "abc", -20.1]    []

1 or more dimensions (matrix)

[[1], [2, 4], ["X"]]    [[10, 20, 30], [20, 0, 10], [30, 20, 10]]

Allows duplicates

x = [False, False, True, False]

Indexable

x[2]

>> True

# More Data Types: Dictionary

Mutable collection of key:value pairs, accessible by key

`d = {1:10, 2:20, 3:30, 4:40}`      `d[1] > 10`

Keys can be strings or integers, values can be any data type

`{"1": "A", "2": "B", "3": "C"}`      `{1: {1: "a"}, 2: ["b"], 3: (1,2)}`

Keys cannot repeat, values can repeat

`{1: ["centre", "for"], 2: ["data", "culture"], 4: ["centre", "for"]}`

Keys and values are accessible as lists with corresponding methods:

`d.keys()`

`>> [1, 2, 3, 4]`

`d.values()`

`>> [10, 20, 30, 40]`

# More Data Types: Tuples

Ordered, indexable collections of items of any data type

`t = (True, False, 10, "number")`

Indexable

`t[-1] >> "number"`

Items can repeat

`("fruit", "vegetable", "fruit", "bread")`

Tuples and their items are immutable

# More Data Types: Set

Unordered, unindexable collection of unique, immutable items

```
s = {"carrot", "broccoli", "green bean", "carrot"}  
s >> {"carrot", "broccoli", "green bean"}
```

Items may be of data type string, int, float, or boolean

```
{True, False} {10.2, "number"} {2, 4, 6}
```

Can add and remove items to a set

```
s.add("peas")  
s >> {"carrot", "peas", "broccoli", "green bean"}
```

# Characteristics of Data Types

	Ordered?	Indexable?	Mutable? (Changeable?)	Allow duplicates?
List	Yes	Yes	Yes	Yes
Dictionary	Yes	No	Yes	Keys - No Values - Yes
Tuple	Yes	Yes	No	Yes
Set	No	No	Change set item - No Add/remove items - Yes	No

**Let's code!**

# Equivalence: `==` and `!=`

Equal to: `==`

True `==` False  
`>>` False

Not equal to: `!=`

True `!=` False  
`>>` True

# Note: = vs. ==

Assign a value to a variable with a **single** equals sign

```
x = "hello world"
```

```
x
```

```
>> "hello world"
```

# **Containment: in, not in**

Check whether or not a value is contained in a list, set, or tuple

```
my_list = [2, 4, 8, 16, 32]
```

```
2 in my_list
```

```
>> True
```

```
my_set = {"apples", "bananas", "oranges"}
```

```
"peaches" not in my_set
```

```
>> True
```

**Let's code!**

# Assignment

`python-basics-3.ipynb`

*Note: please be sure to try your hand at the “Try it” and “Challenge” cells in this Notebook! If you get stuck, don’t worry, we’ll talk through people’s different approaches in class.*

*Created by Nathan Kelber and Ted Lawless for JSTOR Labs under Creative Commons CC BY License*

# Further Resources: W3 Schools

- Python: <https://www.w3schools.com/python/>
- Data types: [https://www.w3schools.com/python/python\\_datatypes.asp](https://www.w3schools.com/python/python_datatypes.asp)
- Lists: [https://www.w3schools.com/python/python\\_lists.asp](https://www.w3schools.com/python/python_lists.asp)
- Dictionaries: [https://www.w3schools.com/python/python\\_dictionaries.asp](https://www.w3schools.com/python/python_dictionaries.asp)
- Tuples: [https://www.w3schools.com/python/python\\_tuples.asp](https://www.w3schools.com/python/python_tuples.asp)
- Sets: [https://www.w3schools.com/python/python\\_sets.asp](https://www.w3schools.com/python/python_sets.asp)

# THANKS EVERYONE!

Office hours available on Wednesday  
*Contact me on Teams to schedule*

Next (last!) class: Friday (4 March)