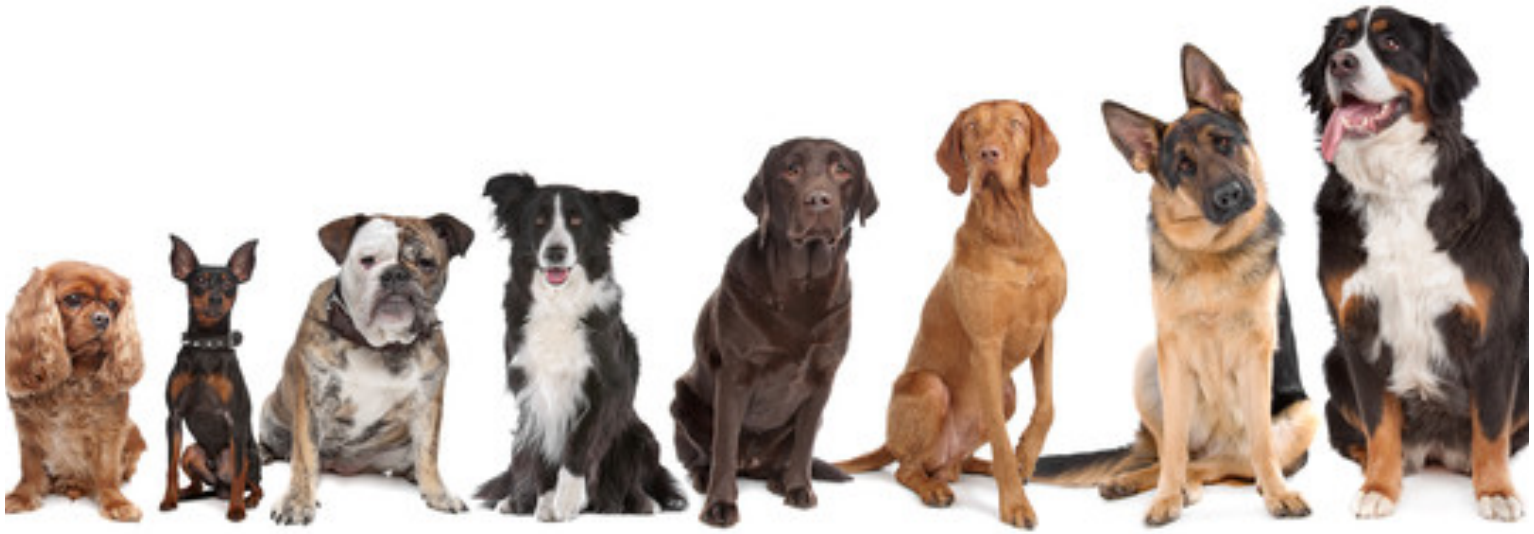
To do before we start:

1. Go to github.com/agithasnoname/pythonPart2 and click on the green "Clone or download" button, then Download Zip.
2. Open Anaconda Navigator and then open Jupyter Lab. Navigate to the folder you just downloaded in the file tree on the left.

   OR

   Open up a text editor and a command line shell (Mac Terminal, Anaconda Prompt, or Windows PowerShell. In the shell, navigate to the folder.



Python Fundamentals Part 2: Loops, Conditionals, and Files

COLBY WITHERUP WOOD

NU IT RESEARCH COMPUTING SERVICES

PART 1 – LAST WEEK

integers
floats
strings
lists
booleans
variables
functions
modules

PART 2 - TODAY

dictionaries
files
for loops
if statements
error handling
scripts

0

```
for person in row:
    person.clap()
```

1

```
for person in row:
    if person > 5'8":
        person.clap()
    else:
        person.clap()
        person.clap()
```

2

```
for p in row:
    if p > 5'8":
        p.clap()
    else:
        pass
```

3

```
for p in row:
    if p > 5'8":
        p.clap()
        p.clap()
    elif p > 5'6":
        p.clap()
    else:
        pass
```

4

```
for p in row:
    if p > 5'8" and p < 6':
        p.clap()
    else:
        pass
```

5

```
for p in row:
    if p > 5'8" and p < 6':
        p.clap()
    else:
        break
```

for loop syntax

```
for p in row:
    p.clap()
```

one tab or four spaces

for loop syntax

```
for p in row:
    p.clap()
```

must match exactly

if statement syntax

```
for p in row:
    if p > 5'8":
        p.clap()
```

Boolean

Let's code!

Checking for errors

Find the error

```
for person in row:
    if person > 5'3":
    person.clap()
    else:
    pass
```

Find the error

```
for person in row:
    if person > 5'8":
        person.clap()
    elif p > 5'6":
        person.clap()
        person.clap()
```

Find the error

```
for you in row:
    you.whistle()
```

try/except

```
for you in row:
    try:
        you.whistle()
    except CantWhistleError:
        pass
```

try/except

```
for you in row:
    try:
        you.whistle()
    except CantWhistleError:
        you.apologize()
```

exercise1.py

PART 1 – LAST WEEK

integers
floats
strings
lists
booleans
variables
functions
modules

PART 2 - TODAY

dictionaries
files
for loops
if statements
error handling
scripts

How would a computer know the heights of Jo, Rae, and Tom?

```
row = ["Jo", "Rae", "Tom"]
for person in row:
    if person > 5'8":
        person.clap()
    else:
        person.clap()
        person.clap()
```

Dictionaries

dictionary syntax

```
inches_dict = {"Jo": 60, "Rae": 68, "Tom": 65}
```

common shorthand
for dictionary

curly brackets

dictionary syntax

```
inches_dict = {"Jo": 60, "Rae": 68, "Tom": 65}
```

key colon value

Note: dictionaries may not always keep the same order

dictionary syntax

```
inches_dict = {"Jo": 60,
               "Rae": 68,
               "Tom": 65}
```

Can be written like this for clarity.

Let's code!

List of dictionaries

```
student_dict = [{"name": "Clancy",
                 "hw_grades": [10, 10, 9, 9],
                 "exam_grades": [94, 91, 97],
                 "participation": 10},
                {"name": "Reba",
                 "hw_grades": [8, 9, 8, 10],
                 "exam_grades": [89, 90, 94],
                 "participation": 10}]
```

exercise2.py

PART 1 – LAST WEEK

integers
floats
strings
lists
booleans
variables
functions
modules

PART 2 - TODAY

dictionaries
files
for loops
if statements
error handling
scripts

Two ways to open files

ORIGINAL (RARELY USE)

```
f = open("my_file.txt", "r")
do something with file
f.close()
```

This leaves the file needlessly open

'WITH AS' SHORTCUT (ALMOST ALWAYS USE)

```
with open("my_file.txt", "r") as f:
    save file as something else
    or save part of file
```

File automatically closes when you exit the indentation

The open function requires two parameters:

```
open(filename, what you're going to do with it)
```

Options for what you're going to do with it:

"r" read
"w" write (wipes the file clean if it already exists) ⚠️
"a" append (add to the end of whatever is already in the file)

If you are accessing a file in your current working directory, you can just include the filename, but if the file is in a different directory, you must include either the relative or absolute path.

Let's code!

Write your own conversion calculator!

There are several ways to do it.

Look at the file conversionMeasures.csv and see how it is formatted.

Use paper and pencil to draw out how you might create the calculator.

Open a new blank text file and save it as conversion.py. Write your code there.

If you want help, the file tips.txt has more specific instructions for one way to do it.