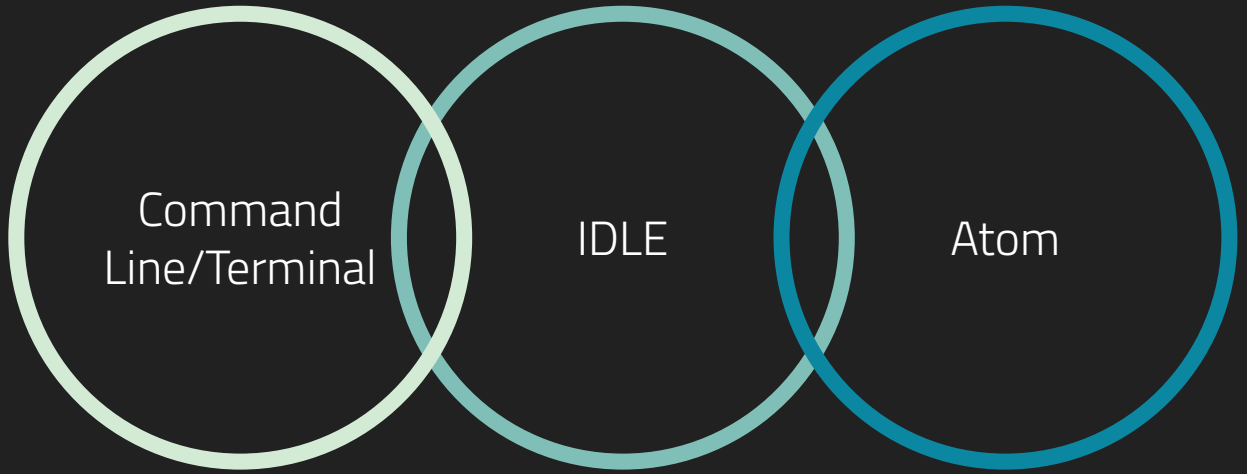
Intro to Programming + Python

Chelsea Parlett-Pelleriti

Commands in Python

*Python is an *interpreted* language, it does one line at a time, which means we can run it one line at a time



Scripts

Python_Dichot.py

```
from __future__ import division
from scipy import stats
import numpy as np
import pandas as pd
import random
import time
#----------------------------------------
def makeMeta():
    meta = {}
    meta["error"] =[]
    meta["Relationship?"] =[]
    meta["Dichot"] =[]
    meta["YES"] =[]
    return meta
#----------------------------------------

test = False
metaData = {"error": [], "Relationship?": [], "Dichot":[], "YES": []}
nExp = 1000
n = 20

error = range(5,250,5)
numTrial = range(0, nExp)

print "I'm Starting now :)"
start = time.time()
print start

for h in range(0,1):
    print h
    for i in error:
```

Parts of a Program

Things other people have written (or you wrote outside of this script)

import numpy

import numpy as np

from pandas import dataframe

```
1  from __future__ import division
2  from scipy import stats
3  import numpy as np
4  import pandas as pd
5  import random
6  import time
```

Parts of a Program

Functions:

(Blocks of reusable code you write yourself)

```
8   def makeMeta():
9       meta = {}
10      meta["error"] =[]
11      meta["Relationship?"] =[]
12      meta["Dichot"] =[]
13      meta["YES"] =[]
14      return meta
```

Parts of a Program

- Statements tell python to do something
- Expressions evaluates and returns a value (my_int +5)

```
# expression
2*2

# statement
x = 4
```

Comments

- # --One Line or Inline
- “” “” -- Multiline
- USE COMMENTS.
  - What does this do?
  - How does this do it?

```
# expression
2*2

# statement
x = 4
```

Whitespace

- Ignored in Expressions and Statements
- At the beginning of a line its “indentation”
- Python uses indentation/whitespace for SYNTAX
- Be CONSISTENT
- Try not to use <tabs>

```
1   # ignorable whitespace
2   x=10
3   x    =        20
4
5   #okay
6   x <= 10
7
8   # not okay
9   x <      = 10
10
11  # indentation
12  for i in [1,2,3,4,5]:
13      print(i)
14
```

Tokens

- Keywords if else list...
- Operators + - / * // % ** ...
- Punctuation/Delimiters () [] {} , ; : # ...
- Literals 123 7.6 True

Variables

Naming Conventions

DO

- Start with a letter or _ (but not really _)
- Use any length name!
- Pay attention to CaSe

```
Chelsea = 10
Chelsea = 10
_Hello = 10 # but don't
```

DON'T

- Start with a number
- Use a keyword

```
for = 10
123BUGS = 10
and = 10
```

Naming Conventions

- https://www.python.org/dev/peps/pep-0008/
- TL;DR do what you want but be CLEAR
- Make my life easier grading

Variable Creation

- Variable is created when it's first used (note: other languages are different)

```
1  Chelsea_is_the_best =  True
2
```

Variable Assignment

- Use =
- Left <- Right

```
1  Chelsea_is_the_best =  True
2  Chelsea_is_the_best = False
3
4  # not
5  True = Chelsea_is_the_best
```

What not to do

```
1  # what not to do
2  7 = chelsea_is_the_best + 1
3
4  chelsea_is_the_best + 1 = 2
5
6  print(x = 4)
7
```