RECAP

Review Topics

- Command line
- GitHub
- Data types & structures
- Conditionals
- Iteration
- Functions
- Descriptive statistics
- NumPy

COMMAND LINE

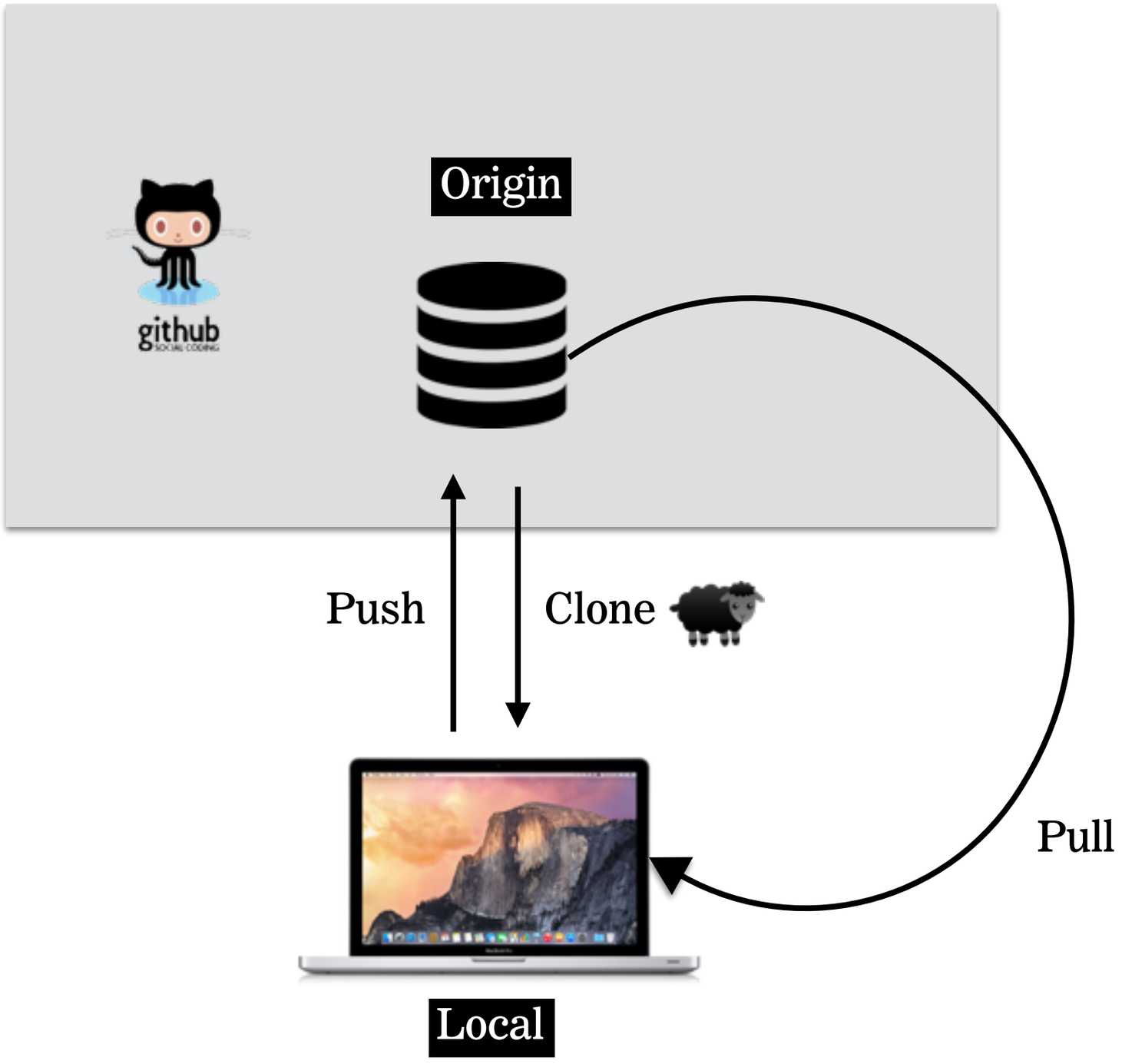
Command Line

- cd - change directory
- mv - move or rename a file
- ls - list directory contents
- rm - remove a file
- pwd - print working directory to the screen
- touch - create a file
- mkdir - make a directory
- <tab> - autocomplete

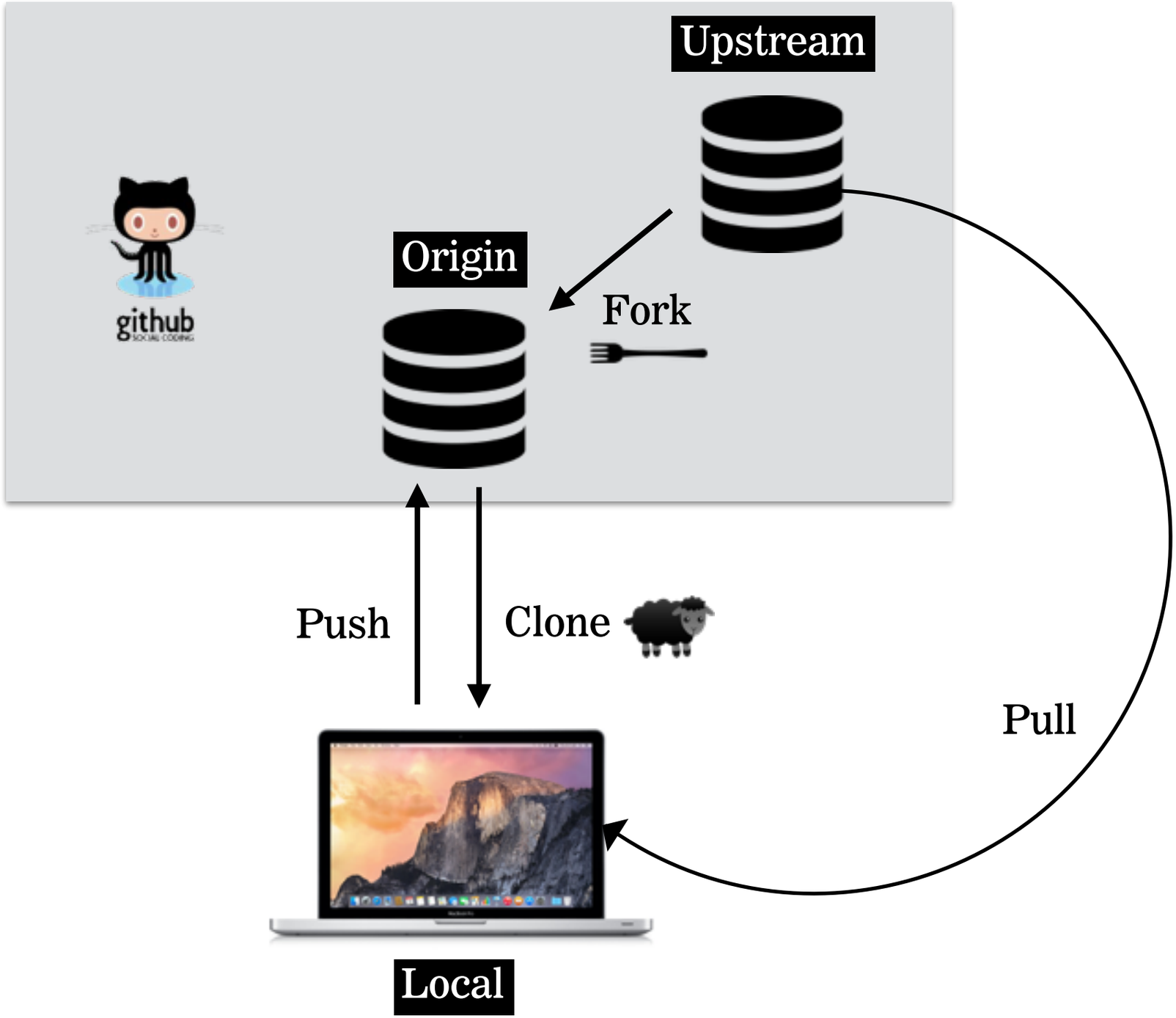
https://linuxjourney.com/lesson/the-shell

GITHUB

GitHub



GitHub



GitHub

- Forking - Saving a copy of someone else's online repo to your online repo on GitHub.com.

- Cloning - Making a local copy (on your machine) of a repo (can be one of your's or someone else's)

- Pulling - Updating your local copy from the online repo. Can be from the origin or the upstream.

- Pushing - Updating your online (origin) repo with your local changes.

GitHub

Upstream - The place you forked from (and added as an upstream)

Origin - The repo you cloned to your local machine

Master - The branch name. Master in terms of our file analogy is the final_final_seriouslythistime.ppt version. You'll learn more about creating feature branches later in the course.

DATA TYPES & STRUCTURES

Data Types & Structures

- Numbers
  - Ints: -100, -3, 0 ,5, 566
  - Floats: -100.23, 0.0, 5.69383

- Can not be iterated over

Data Types & Structures

- Lists - [ ]
- Dictionaries - { }
- Tuples - ( )
- Strings - “ ” or ‘ ’

- Can be iterated over
- Can be concatenated
- Mutable (i.e., can change values): Lists and Dictionaries
- Immutable: Tuples and Strings
- All but Dictionaries can be indexed and sliced

CONDITIONALS

Conditionals

Q: What is a conditional?
A: A statement that evaluates something as either true or false

Pseudocode:
If the temperature is greater than or equal to 95 degrees:
 stay inside
Else if the temperature is greater than 45 but less than 95:
 go outside
Else
 stay inside

Conditionals

Q: What is a conditional?
A: A statement that evaluates something either true or false

Example:
temp = 66

```
if temp >= 95:
    print ‘stay inside’
elif temp > 45 and temp <95:
    print ‘go outside’
else:
    print ‘stay inside’
```

ITERATION

Iterating

Q: What is iteration?
A: A process that loops through each element in an iterable.

Pseudocode:
for each element in iterable:
    do something to that element

Iterating

Example 1:

```
my_list = [("one", 1.0), ("two", 2.0), ("three", 3.0)]

for each_tuple in my_list:
    print each_tuple[0]
```

indexing the tuple element

Example 2:

```
my_list = [("one", 1.0), ("two", 2.0), ("three", 3.0)]
```

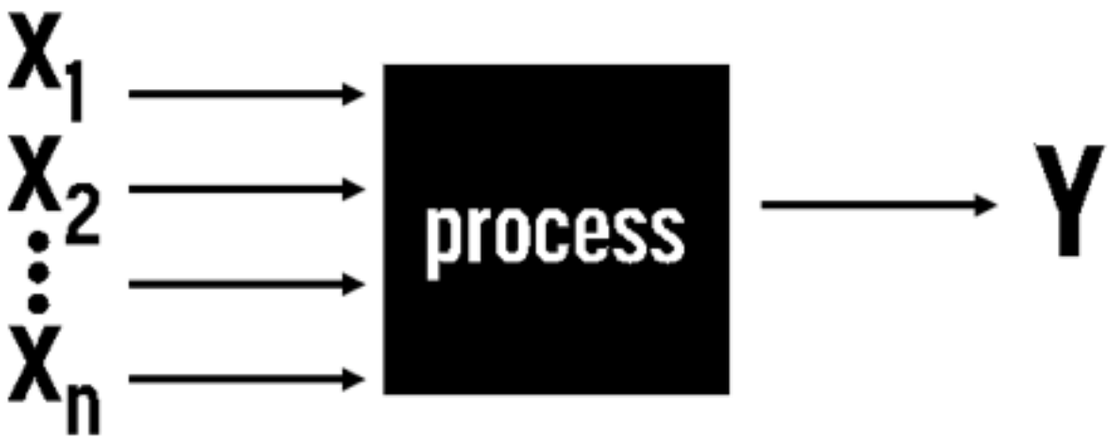
Unpacking the tuples

```
for tup_1, tup_2 in my_list:
    print tup_1
```

FUNCTIONS

Q: What is a function?
A: A function is an encapsulated block of code that is called upon to perform some procedure. It can be called by passing in one or more data types or structures, or it can be called without passing anything to it. It typically returns a data type or data structure.

Functions



inputs

output

Q: Why use functions?
A: DRY - Don’t repeat yourself.
Functions make code reusable and modular.

```
def the_doubler(num_to_double):
    result = num_to_double * 2
    return result

the_doubler(4)
the_doubler(20)
```

Pass in 4, function returns 8

Pass in 20, function returns 40

Example 1:

```
def get_a_number():
    result = randint(1, 10)
    return result

get_a_number()
```

Define the function

Do some series of operations

Return something back to the call

Call the function

Parameters passed in.
This function takes two.

Example 2:

```
def add_two_numbers(number_one, number_two):
    result = number_one + number_two
    return result
```

```
add_two_numbers(3, 4)
```

Parameters being passed to function

Example 3:

```
def get_list_mean(number_list):
    num_total = sum(number_list)
    element_count = len(number_list)
    result = num_total/element_count
    return result

mean_val = get_list_mean([1, 2, 3, 4, 5])
```

Assigning the result of the call to a variable

DESCRIPTIVE STATISTICS

Descriptive statistics

Measures of central tendency, i.e. “averages”:

Mean (arithmetic, geometric, harmonic, …)
Median
Mode