BC COMS 2710: Computational Text Analysis

Lecture 2 – Python Overview

Announcements

- Tutorial 1.1
  - Should be submitted today (Tuesday 05/04)
- Tutorial 1.2
  - Should be submitted tomorrow (Wednesday 05/05)
- Tutorial 1.3
  - Should be submitted Friday (05/07)
- Homework 01:
  - Due Monday 05/10
- Reading Week 1
  - Due Sunday 05/09

Updated Rubric

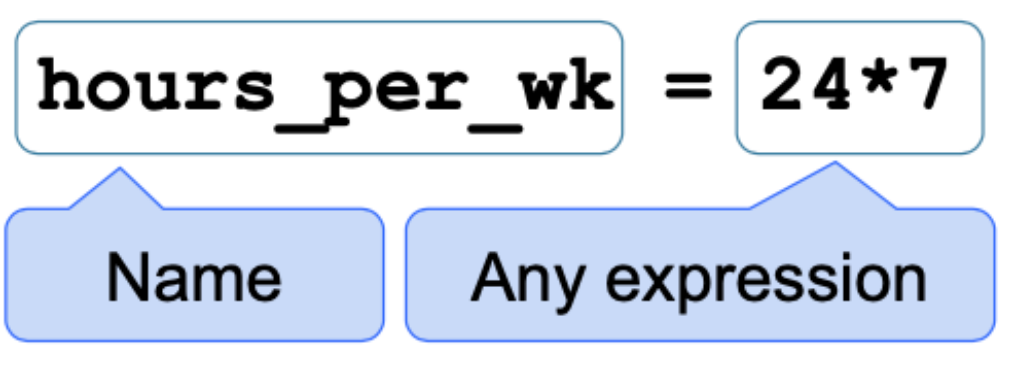
Participation	5%
4 Homeworks	30%
Reading reflections	15%
Daily Tutorials	20%
Final Project	35%

Python

- Popular for data science & software development
- Mature data science and computational text analysis tools
- Learn through practice and doing
- Follow along in the demos

Names & Variables

Assignment Statements



- Statements perform an action
  - don't have a value
- Assignment statement changes the meaning of the name to the left of the = symbol
- The name is bound to a value (not an equation)

Numbers

Numbers – Integers and Floats

Two real number types in Python

- int: an integer of any size
- float: a number with an optional fractional part

An int never has a decimal point; a float does

A float might be printed using scientific notation

Limitations on float values

- Floats have limited size (the limit is huge)
- Floats have limited precision of 15-16 decimal places
- After arithmetic, the final few decimal places can be wrong

Strings

Strings

A string value is a snippet of text of any length

- ‘a’
- ‘word’
- “there can be 2 sentences. Here’s the second!”

Strings consisting of numbers can be converted to numbers

- int(‘12’), float (‘1.2’)

Any value can be converted to a string

- str(5) becomes “5”

Discussion Question

Assume you have run the following statements:

```
x = 3
y = '4'
z = '5.6'
```

What is the source of the error in each example?

A. `x + y`
B. `x + int(y + z)`
C. `str(x) + int(y)`
D. `y + float(z)`

Types – Every value has a type

We've seen 5 types so far:

- int: 2
- float: 2.2
- str: 'Red fish, blue fish'

Types – Every value has a type

The type function tells you the type of a value

- type(2)
- type(2+2)

An expression's "type" is based on its value

- x = 2, y="hello"
- type(x), type(y) = ???

Conversions

Strings that contain numbers can be converted to numbers

- int("12")
- float("1.2")
- ~~float("one point two")~~ # Not a good idea

Conversions

Any value can be converted to a string

- str(6)

Numbers can be converted to other numeric types

- float(1)
- int(2.3). # DANGER: why is this a bad idea

Collections

Collections considered

Ordered:

- Lists
- Tuples

Unordered:

- Sets
- Dictionaries

Lists

- store multiple items in a single variable
  - fruit = [“bananas”, “apples”, ”oranges”]
- Order is preserved
- Access items with brackets
  - first_fruit = fruit[0]
  - second_fruit[= fruit[1]
  - last_fruit = fruit[-1]
  - What are the values assigned to these three names?

Lists – accessing sub-lists

fruit = [“bananas”, “apples”, “oranges”]

- Access multiple items:
  - sub_fruit1 = [0:2]
  - sub_fruit1 = ???

  - sub_fruit2 = [:2]
  - sub_fruit2 = ???

  - sub_fruit3 = [0:]
  - sub_fruit3 = ???

Lists – adding to lists

fruit = [“bananas”, “apples”, “oranges”]

- Adding at the end:
  - fruit.append(“grapefruit”)
- Modifying at a specific location:
  - fruit[1] = “strawberry”
  - fruit ???

Tuples

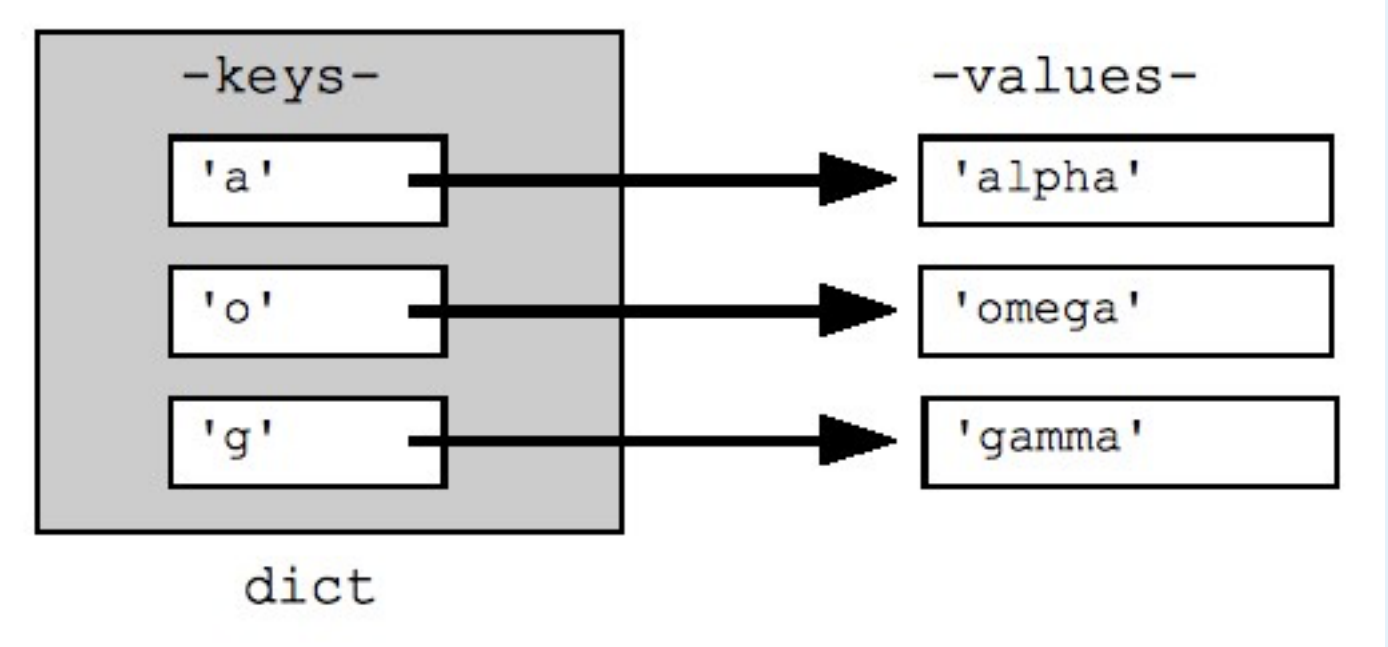
- Immutable lists
- play = ("Shakespeare", "A Midsummer Night's Dream", 1595)
- Used to group together related data

Set

- Unordered and unindexed collection
- authors = set(["Shakespeare", “Austin”, “Morrison”, “Woolf”])
- No duplicates

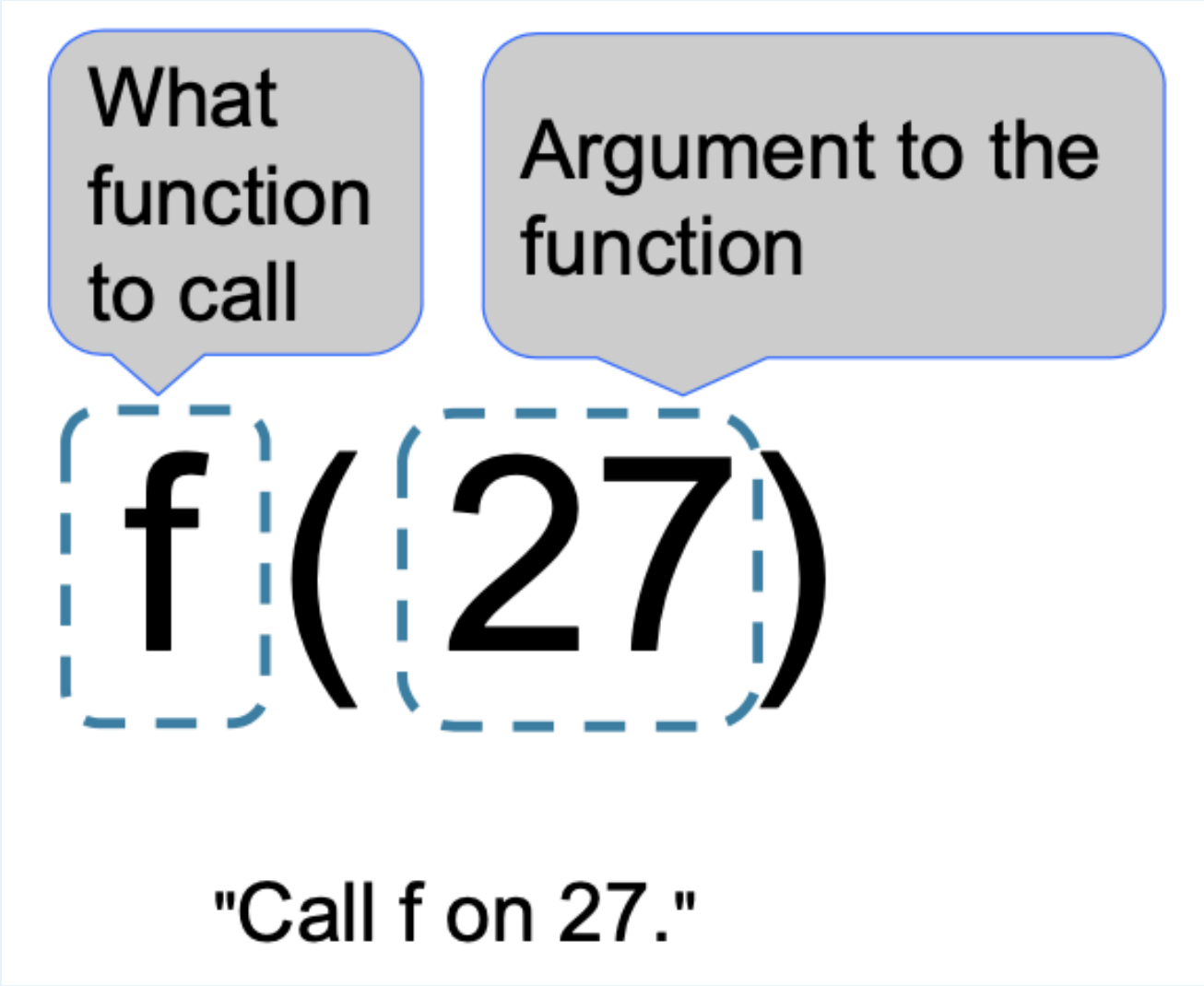
Dictionary

- Store data values in *key:value* pairs.
- Ordered, changeable, no duplicates
- {“a”: 1,
  “b”: 2,
  “c”: 3}

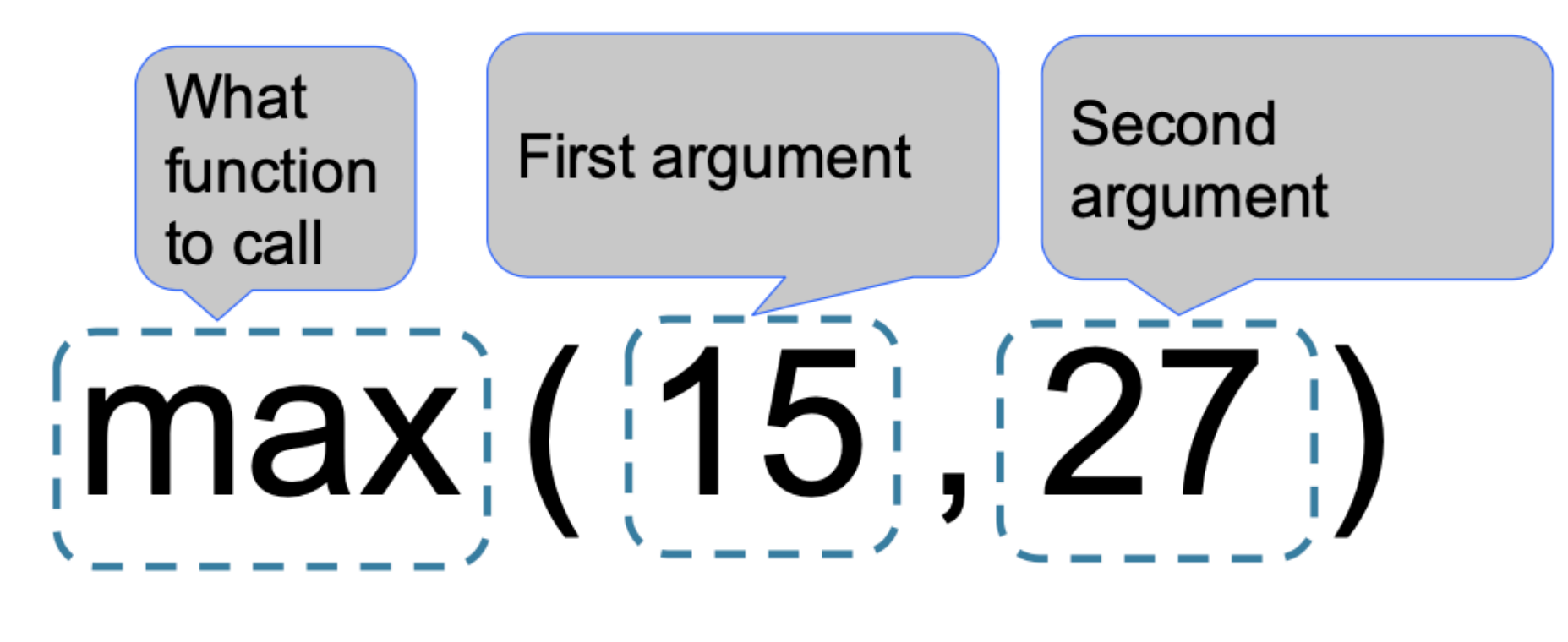


Functions

Anatomy of a Call Expression



Anatomy of a Call Expression



Python Built-in Functions

		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

Control Statements

Loops

Say we have a list of author names, how can we find the length of each name?

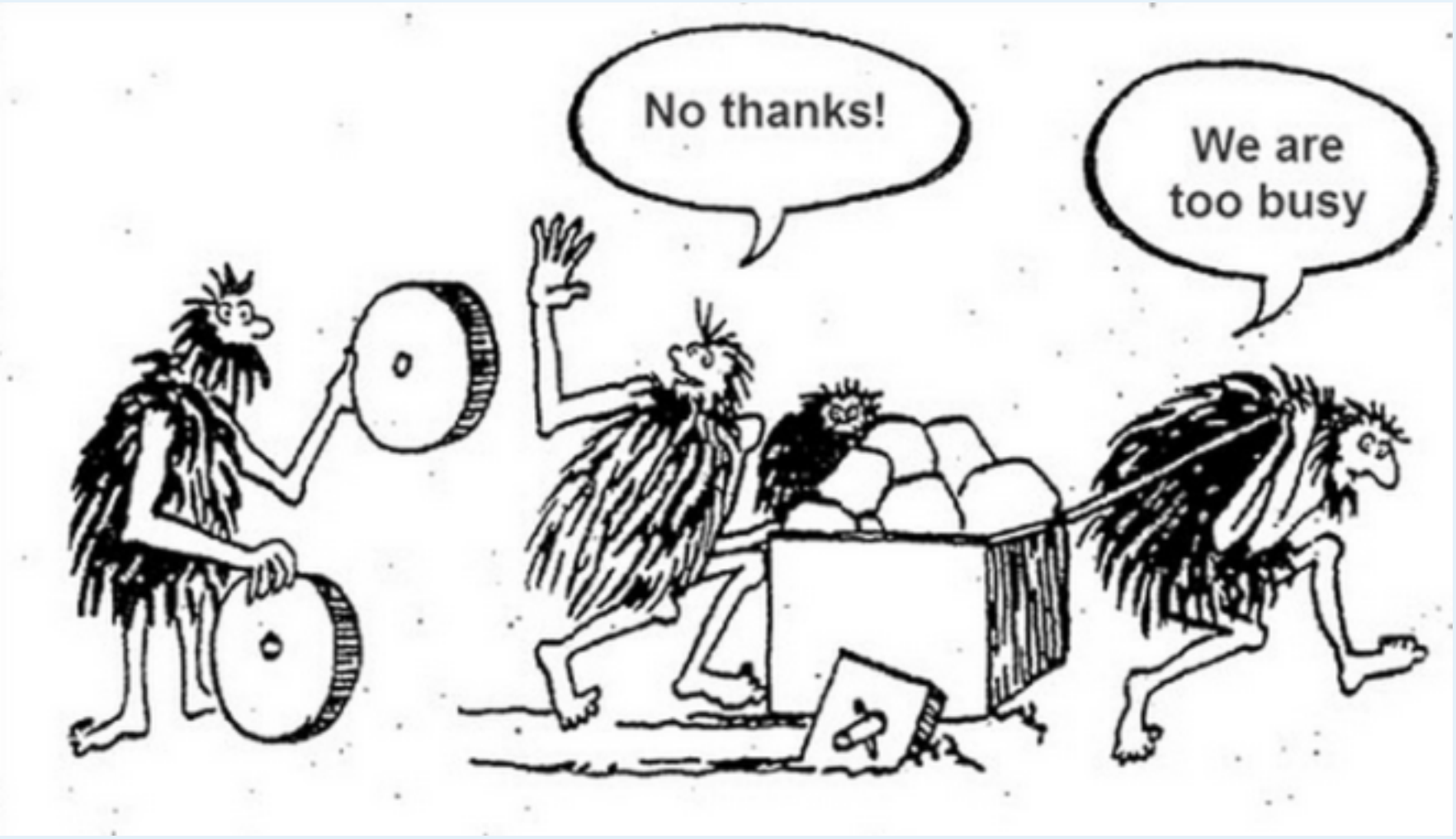
Conditionals

Only apply computation under certain scenario

Libraries

Libraries

- Don’t reinvent the wheel



Installing and Accessing Python Libraries

- Install via command line:
  - pip install <Library Name>
- Access library in python:
  - import <Library Name>

Libraries we will use

- BeautifulSoup – webscraping

- Nltk – Processing text
- Spacy - Processing text
- little_mallet – Topic Modeling

- Pandas - Tables
- Matplotlib - Visualization
- Numpy – Vectors

- Sklearn – Machine Learning