

# CSCi 1012 [Section 10]

---



## Introduction to Programming with Python

Prof. Kartik Bulusu, CS Dept.

Course start date January 17, 2024

Lecture location 1957 E street Room 213

Lecture times Monday, 3:45 PM to 5:00 PM

### Wednesday-lab

3:45 PM to 5:00 PM

Section-30: MON 352

Section-31: SEH 4040

Section-34: TOMP 310

Section-35: TOMP 204

### Friday-lab

3:45 PM to 5:00 PM

Section-32: SEH 4040

Section-33: TOMP 309

Section-36: TOMP 306

Section-37: TOMP 107



School of Engineering  
& Applied Science

Spring 2024

THE GEORGE WASHINGTON UNIVERSITY

Photo: Kartik Bulusu

# Class Policy on Collaboration

- You may **not** discuss *Modules, Assignments, Quizzes and Exams* among yourselves.
- Each student is expected to work out the course deliverables **independently**.
- Under **no circumstances** may you look at another student's *Modules, Assignments, Quizzes and Exams*, or look for answers to *Modules, Assignments, Quizzes and Exams* anywhere other than in the text in the course website.
- You are encouraged to discuss the class material on Ed-discussion board or in-person with the instruction team.
- You may **not** discuss *Modules, Assignments, Quizzes and Exams* nor give out hints for the same on problems on the Ed-discussion board or with other students in-person.

***All violations will be treated as violations of the Code of Academic Integrity.***

# Announcements

- Please contact the offices concerned if you need any special accommodations
- We will support your request; you will need to contact them soon
- Office that supports your request will guide you on all special requests

- **Exam on Monday, 04/15 during the lecture.**

- Location: 1957 E street Room 213
- Duration: 60 minutes
- Start time: 3:45 PM
- Paper exam
- Closed everything (no notes, no Thonny)
- Multiple choice/ fill in the blanks/ write the output – similar to quizzes
- 25 questions
- Answer as many as you can; Up to 20 correct answers will be graded for 5 points each
- Total: 100 points
- 60 % to Pass the exam and the course
- Final grade will be according to the grading scale
  - Examination 30%, Assignment 25%, Modules 15%, Quizzes 20%, Professionalism 10%

- **Make-up Exam on Monday, 05/06 (exam week)**

- Only for students who Fail Exam on 04/15
- Location: 1957 E street Room 213
- Duration: 50 minutes; Start time: 5:20 PM
- Paper exam
- Closed everything (no notes, no Thonny)
- Multiple choice/ fill in the blanks/ write the output – similar to quizzes
- 20 questions (no choice, 5 points each)
- Total: 100 points
- 60 % to Pass the Make-up exam and the course
- Final grade (with grade scale) will not be higher than the lowest grade of the students who passed the exam on 04/15
- If you don't appear for the Exam on 04/15, you will NOT be permitted to take the Make-up Exam

## HWs

- Due dates
- Late work
- Extensions

- **48-hour extensions:**

- After the 4<sup>th</sup> extension your subsequent late submissions will not be considered toward final grade.

Date	Topic(s)	Wednesday Lab Date	Friday Lab Date	Assignment(s)
Week 11 [04/01/2024]	<b>while</b> loops, I/O	04/03/2024	04/05/2024	Unit 1 » Module 4 & Module 5 (Due <b>April 10, 2024</b> by <b>11:59 PM</b> )
Week 13 [04/15/2024]	<b>Examination</b> <b>In the lecture room</b> <b>1957 E Room 213</b> <b>Start time: 3:45 PM</b>	04/17/2024	04/19/2024	Unit 2 » Module 0 & Module 1 (Due <b>April 22, 2024</b> by <b>11:59 PM</b> )

CRN	Subj	Crse	Section	Building	Room	Make-up Exam Date	Duration: 50 minutes	Instructor
93983	CSCI	1012	10	1957 E	213	Only those who appeared and failed exam on April 15, 2024 are eligible		
						Monday, May 6, 2024 5:20pm-7:20pm		Bulusu

## Late Work

- **Late work is not accepted, with the following exceptions:**
  - Every student may turn in **as many as four (in total, not each) assignments or modules 48 hours after the deadline with no penalty**. Requesting an extension is not necessary.
- **Extensions will be granted should there arise circumstances beyond your control** that impede your ability to complete coursework.
  - Notify your professor as soon as feasible in these cases.
    - Examples of such circumstances include (but are not limited to) illness, death in the family, and loss of housing. To ensure fairness toward all students, we will request documentation of such circumstances.



## Submission Tips

- Pay attention to file names

extra spaces  
↓

`assignment1.zip`  $\neq$  `assignment 1.zip`

`caesar_shift.py`  $\neq$  `caesar_shift .py`

↑  
extra spaces

- Pay attention to small details

```
x is 3 and j is 2, x + j = 5
x is 3 and j is 3, x + j = 6
x is 3 and j is 4, x + j = 7
x is 3 and j is 5, x + j = 8
x is 3 and j is 6, x + j = 9
x is 3 and j is 7, x + j = 10
x is 3 and j is 8, x + j = 11
```

$\neq$

*missing  
commas*  
↓

```
x is 3 and j is 2 x + j 5
x is 3 and j is 3 x + j 6
x is 3 and j is 4 x + j 7
x is 3 and j is 5 x + j 8
x is 3 and j is 6 x + j 9
x is 3 and j is 7 x + j 10
x is 3 and j is 8 x + j 11
```

↑  
*extra space*      *missing =*

- Don't add extra endline spaces

`print("++++", end = "")`

++++

+ +

+ +

++++

*What Auto Grader Sees...*

'++++\n+ +\n+ +\n++++'

`print("++++", end = " ")`

++++

+ +

+ +

++++

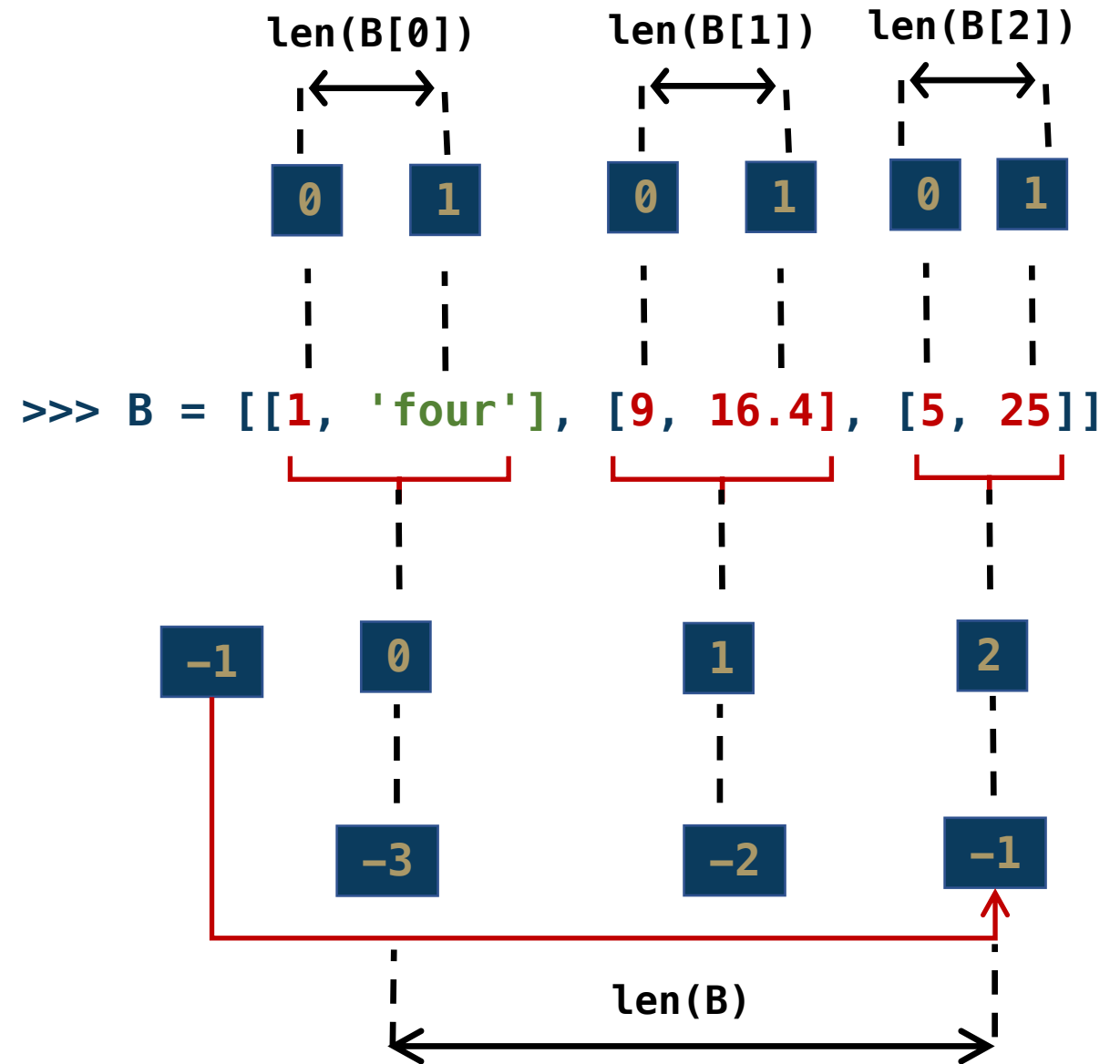
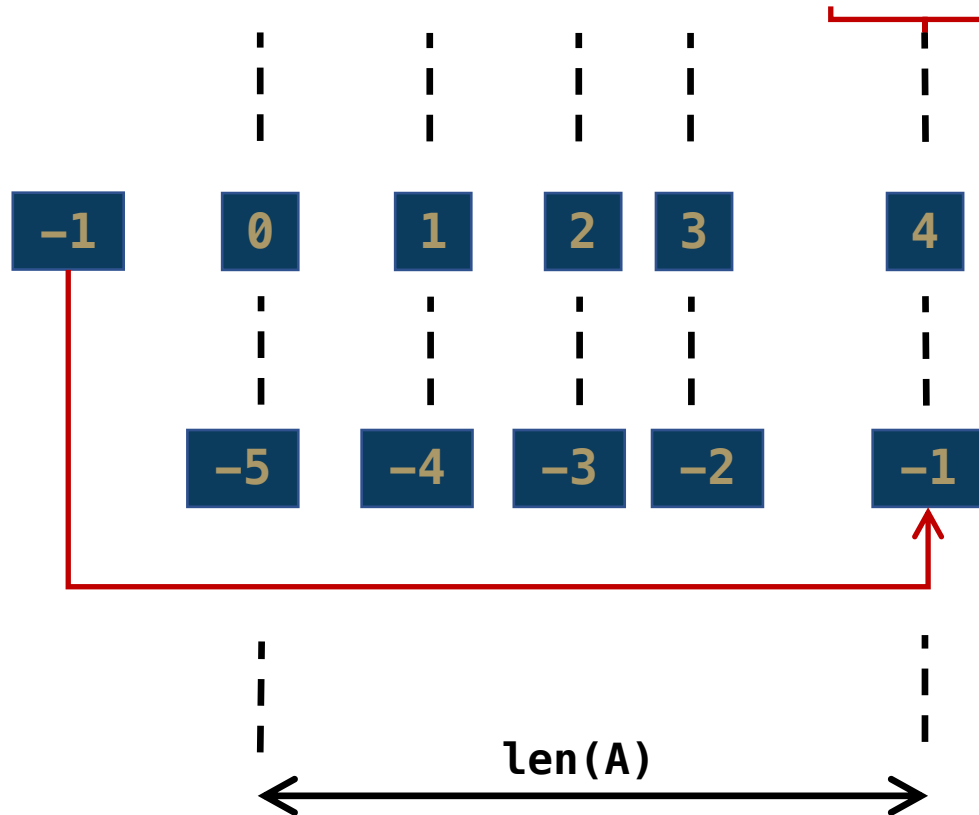
↑  
*extra spaces*

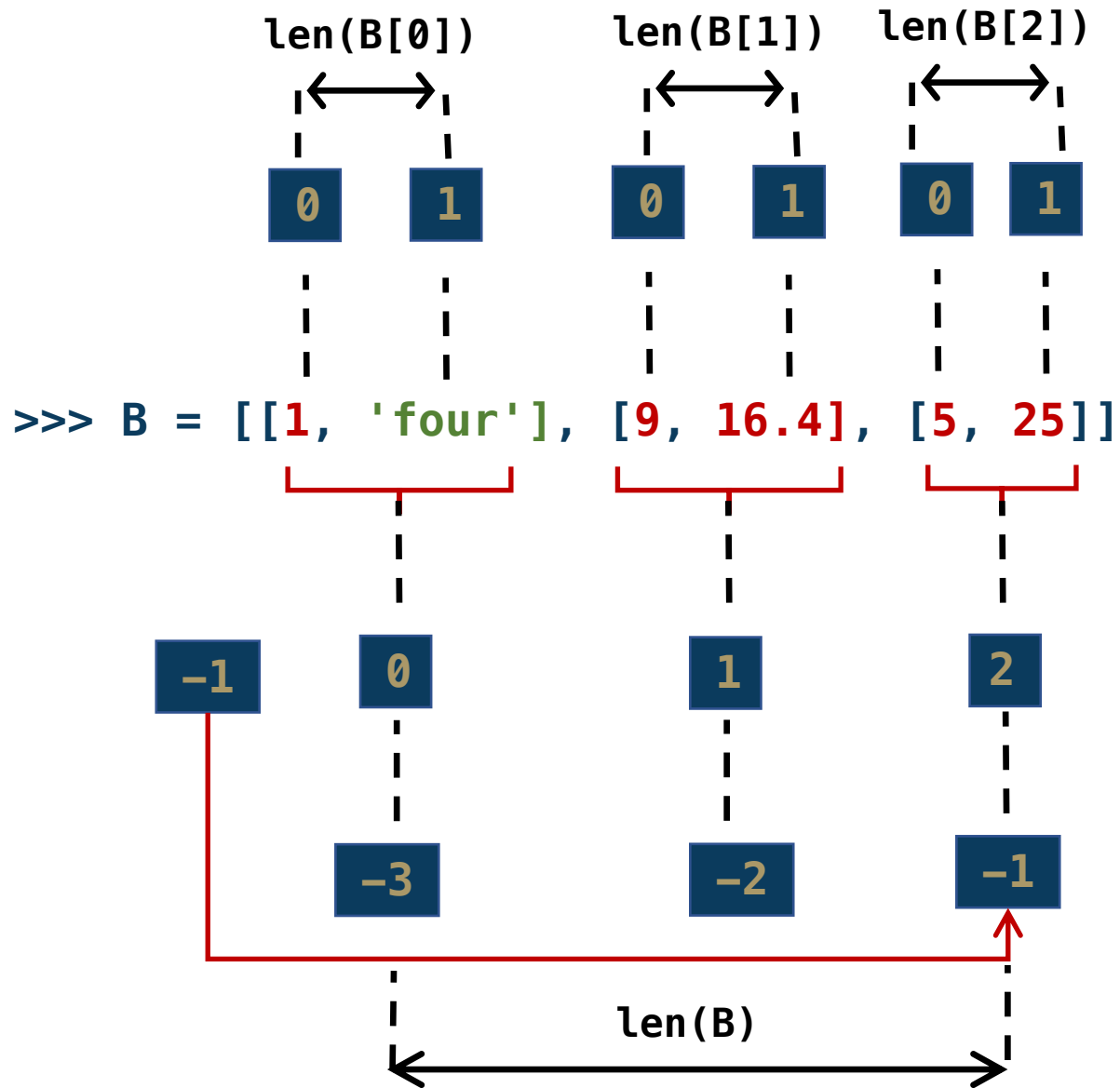
$\neq$   
'++++ \n+ +\n+ +\n++++ '

## Lists & indexing

```
>>> A = [1, 'four', 9, 16.4, [5, 25]]
```

An element of a list





```
>>> len(B) # number of list elements
3
```

```
>>> B[2] # the third element in the list
[5, 25]
```

```
>>> B[3] → 
```

```
>>> B[1][1]
16.4
```

```
>>> B[-1][0]
5
```

```
>>> B[0][1]+1
>>> i=3; B[i-2][1]+1
```

# List of lists of lists

One needs to keep track of the square brackets !

```
A = [ [ [1,2], [3,4], [5,6] ], [ [7,8], [9,10], [11,12] ] ]
```

```
print(A[0])
```

```
[[1, 2], [3, 4], [5, 6]]
```

Get the third element of the outermost element:

```
A = [ [ [1,2], [3,4], [5,6] ], [ [7,8], [9,10], [11,12] ] ]
```

```
print(A[0][2])
```

```
[5, 6]
```

Get the second element of the third element of the outermost element:

```
A = [ [ [1,2], [3,4], [5,6] ], [ [7,8], [9,10], [11,12] ] ]
```

```
print(A[0][2][1])
```

```
6
```



# NumPy

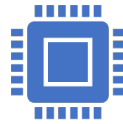
```
import numpy as np
```



NumPy (**Numerical Python**) is an open source Python library that's used in almost every field of science and engineering.



Very large lists (million of elements or more) can slow down a program.



A list-of-lists is even slower for large sizes, and takes up a lot of memory.



NumPy arrays (we will often call them simply “arrays”) were created as a separate structure in Python to enable efficient processing of lists of numbers, especially multidimensional lists.

```
>>> import numpy as np
>>> a = np.array([1, 2, 3])
```

```
>>> import numpy as np
>>> A = np.array([[ -1, 2],[3, 4]])
>>> B1 = A * 6
>>> B2 = A * (1/6)
>>> len(B1)
>>> np.shape(B2)
```

## Element-by-element addition made easy by numpy

```
>>> list_1 = [1, 2, 3]
>>> list_2 = [4, 5, 6]
```

Puts the lists together

### Concatenation of Lists with + operator

```
>>> list_3 = list_1 + list_2
>>> list_3
[1, 2, 3, 4, 5, 6]
```

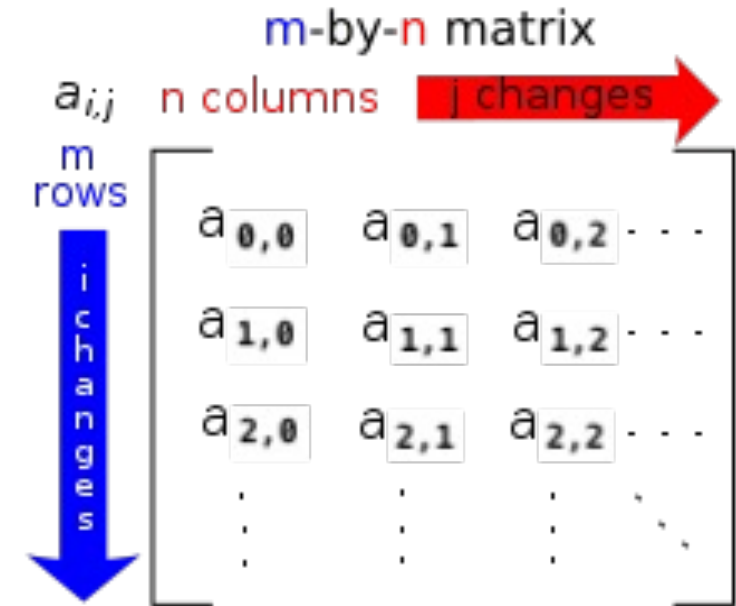
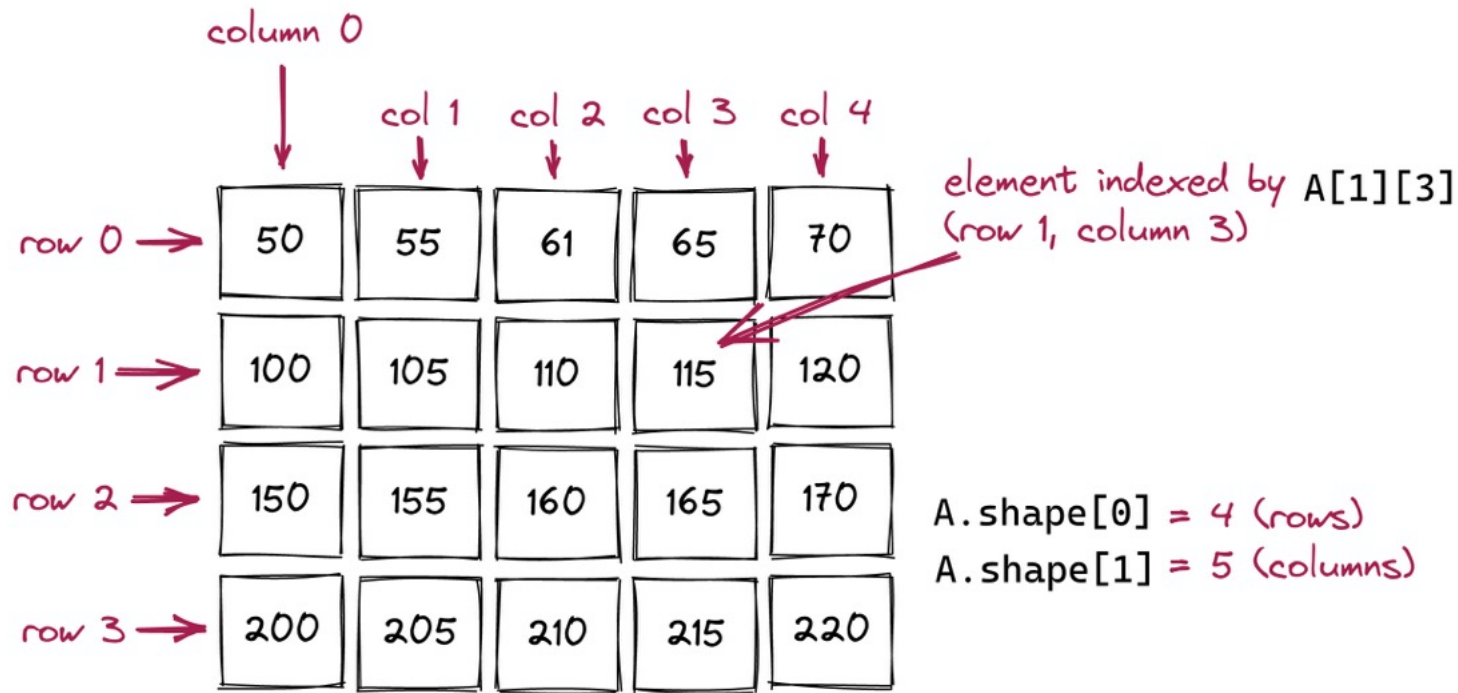
### Element-by-element addition

```
>>> list_4 = []
>>> list_4.append(list_1[0]+list_2[0])
>>> # Use a loop to repeat on all elements
```

```
1 import numpy as np
2
3 A = np.array([1, 2, 3])
4 B = np.array([4, 5, 6])
5
6 C = A + B           # Direct element-by-element addition
7 print(C)           # [5, 7, 9]
8
9 D = np.add(A, B)    # The same, via the add() function in numpy
10 print(D)           # [5, 7, 9]
11
12 E = B - A           # Elementwise subtraction
13 print(E)           # [3, 3, 3]
```

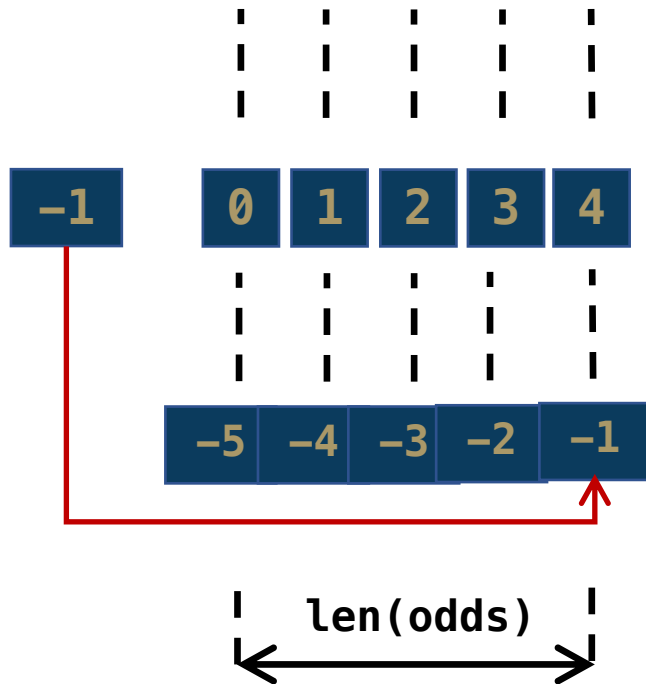
Demo

# Bookkeeping in 2D arrays

Source: [http://en.wikipedia.org/wiki/Matrix\\_\(mathematics\)](http://en.wikipedia.org/wiki/Matrix_(mathematics))
[Demo](#)

# Indexing and Slicing Lists and numpy-arrays

```
>>> odds = [1, 3, 5, 7, 9]
```



Retrieve list-elements or numpy-array-elements with a range of values

```
>>> import numpy as np
>>> odds = np.array([1, 3, 5, 7, 9])
```

*start*      *stop*

```
>>> odds[2:5]
array([5, 7, 9])
```

*start* *stop* *step*

```
>>> odds[0:5:2]
array([1, 5, 9])
```

*start* *stop* *step*

```
>>> odds[-1:-5:-2]
array([9, 5])
```

```
>>> odds[5:0:-2]
array([9, 5])
```

***start:*** at the index value  
***step:*** up or down at the increment value (default = 1)  
***stop:*** at the index value but not including it

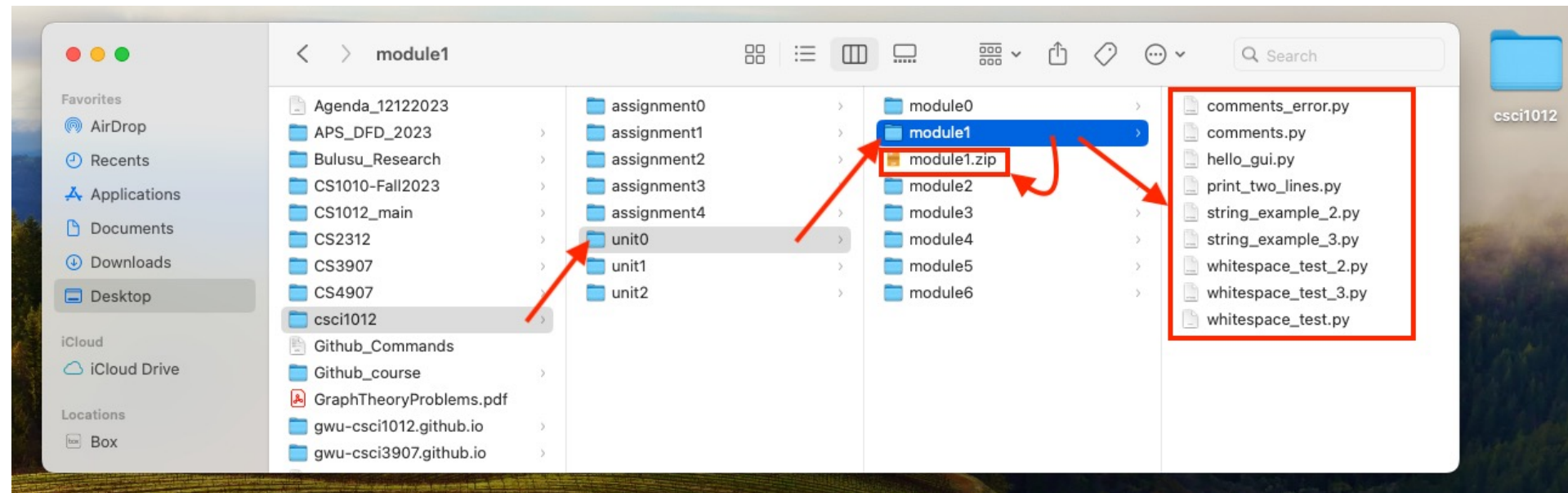
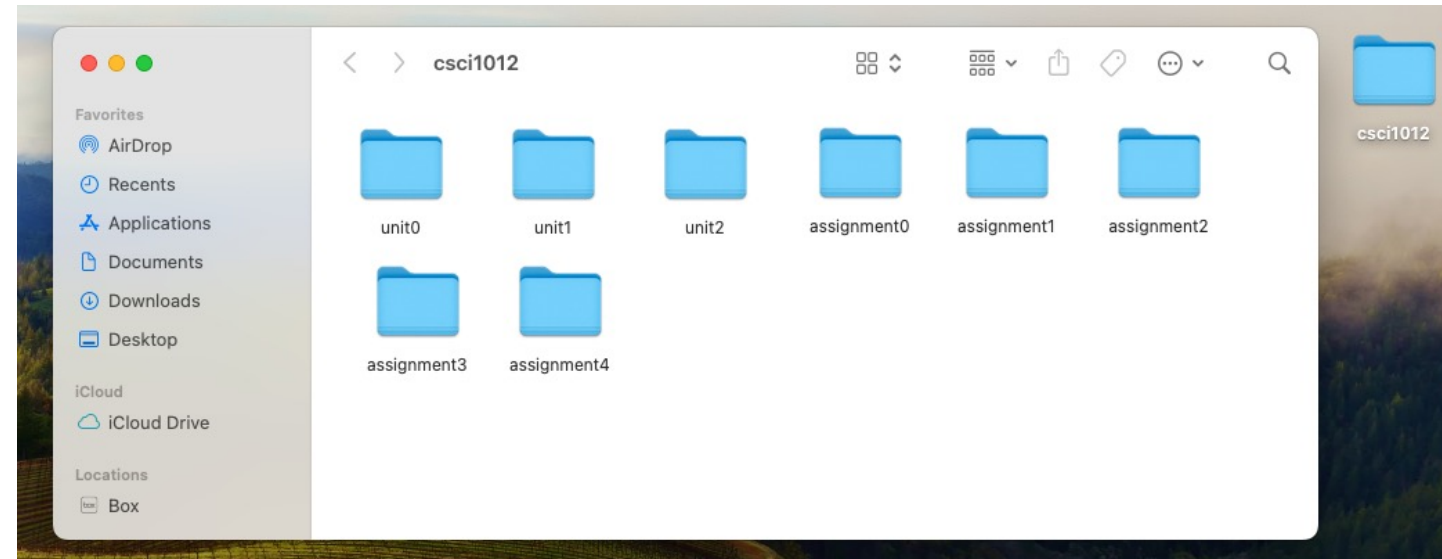
## File-folder-structure

`module0.zip` (correct)

`Module0.zip` (wrong: starts with uppercase)

`module 0.zip` (wrong: space before 0)

`module0.docx` (wrong: not a zip).



See you all in the Wednesday and Friday Labs!