SEE1002
Introduction to Computing for Energy and Environment

Dr. Keith Ngan
School of Energy and Environment
City University of Hong Kong

Tuesday, 12:00-13:30 (LI G600)
Thursday, 15:00-16:30 (LT-18)

Outline

1. Preliminary information
2. Key questions about the course
3. Syllabus and learning activities
4. Assessment
5. Resources

1.Preliminary information

Expectations

- It is expected that students will show up on time for all SEE courses.
- Students are expected to behave respectfully and in accordance with CityU regulations.

Instructor

Keith Ngan
Office: YEUNG B5436
Phone: 3442 6702

Email: keith.ngan@gapps.cityu.edu.hk
Zoom: https://cityu.zoom.us/j/3858927145

Office Hours

Monday 17:00-19:00
Friday 17:00-19:00

You can meet with me during office hours (in person or via Zoom) if you need extra help. You are also welcome to come at other times, but you should probably confirm with me ahead of time.

Teaching Assistants

DING Yaxin *(left half)*

Email: **yaxinding3-c@my.cityu.edu.hk**

WhatsApp: 85294214746

WeChat: 85294214746

CHEN Siru *(right half)*

Email: **siruchen-c@my.cityu.edu.hk**

WhatsApp: 67386558

WeChat: 15521315285

Responsibilities: assistance with computer labs, marking of homework

Note: During the computational labs, you should ask the TA who's been assigned to you, but this isn't strictly necessary.

2. Key questions about the course

Why is it important to learn about computing and computer programming?

- Nowadays computer skills are vital in industry and academia. Many jobs require programming knowledge.
- Many interesting and important problems in energy and environment can only be solved using computers.
- SEE students have asked for more computing-related courses. This course will provide you with fundamental knowledge that will be needed for future course.

Bottom line: computing skills are extremely valuable!

Looking ahead

Several courses offered in subsequent years feature a significant computational component.

- SEE2003 - Introduction to Energy and Environmental Data Analysis
- SDSC3002 - Data Mining
- SEE4112 - Sustainable Engineering Systems: Modelling and Analysis
- SEE4218 - Wind and Marine Energy
- SEE4204 - Environmental System Modelling
- SEE4996/4997 - Final Year Project

***Note*: Python will not necessarily be used in every course. However, what you learn in this course will be directly applicable.**

What is the level of this course?

- This is an introductory course. No computer programming knowledge will be assumed.
- Nonetheless this is not a GE course either. You will be expected to write programs of your own!
- However, there is no need to panic. We will emphasize the fundamental concepts.
- Nonetheless you will need to make an effort to keep up with the material.

What are the aims of this course?

1. Understand what can and can't be done with computers.
2. Introduce students to the basics of programming
3. Develop skills that can be used in future courses.

Why learn Python?

There are many computer programming languages…why have we chosen Python?

- Easy to learn
- Popular
- Looks good on CV
- Free
- Can be used for programming, solving equations, data analysis and plotting.

N.B. we will learn Python 3.

On Python 2 versus Python 3

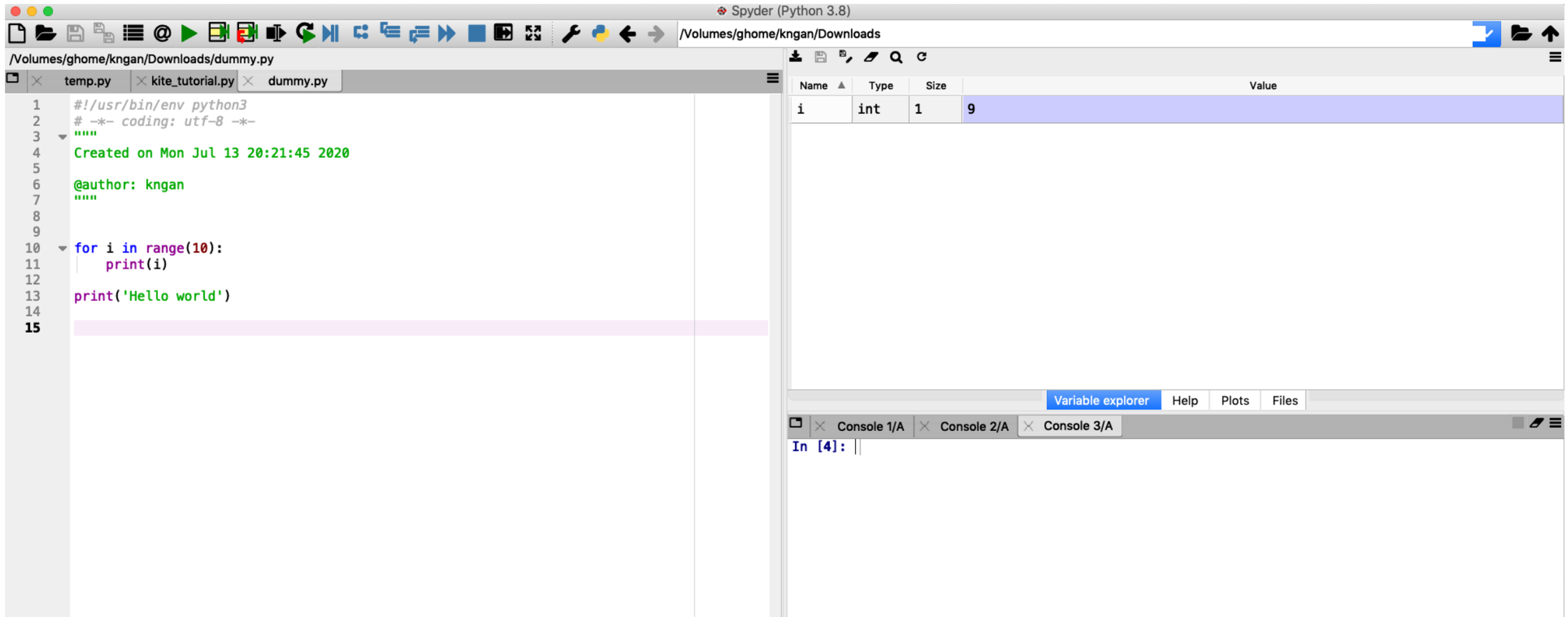
There are two versions of Python. Both are widely used.

- Differences between them are minor.
- We will use Python 3. In previous years, Python 2 was taught.
- The notes, videos, labs and exercises all use Python 3.

Python distribution

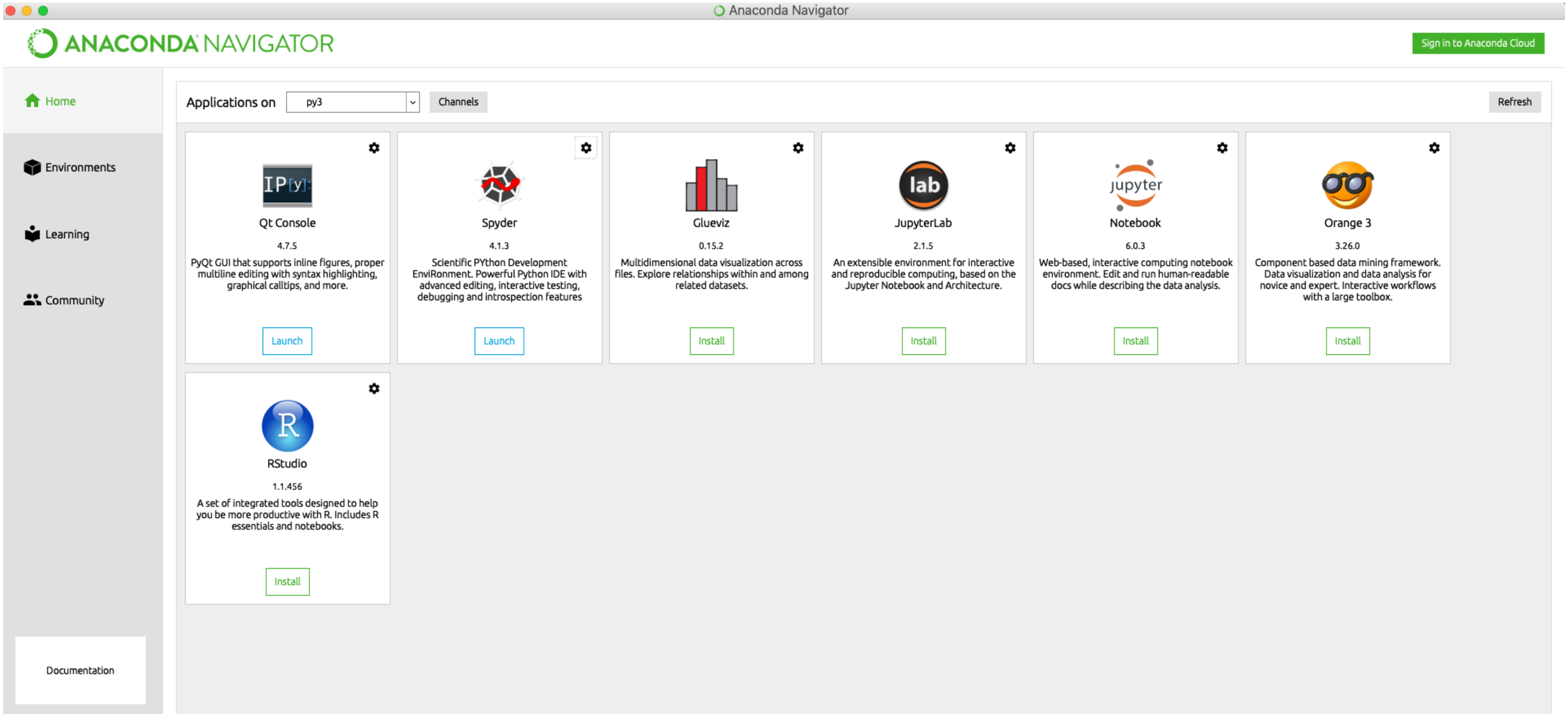
Python is a computer language. But there are different Python distributions. Each includes slightly different components or tools.

For our purposes, the integrated development environment is the most important part of a Python distribution. It includes tools for editing and debugging Python programs as well as for running them.



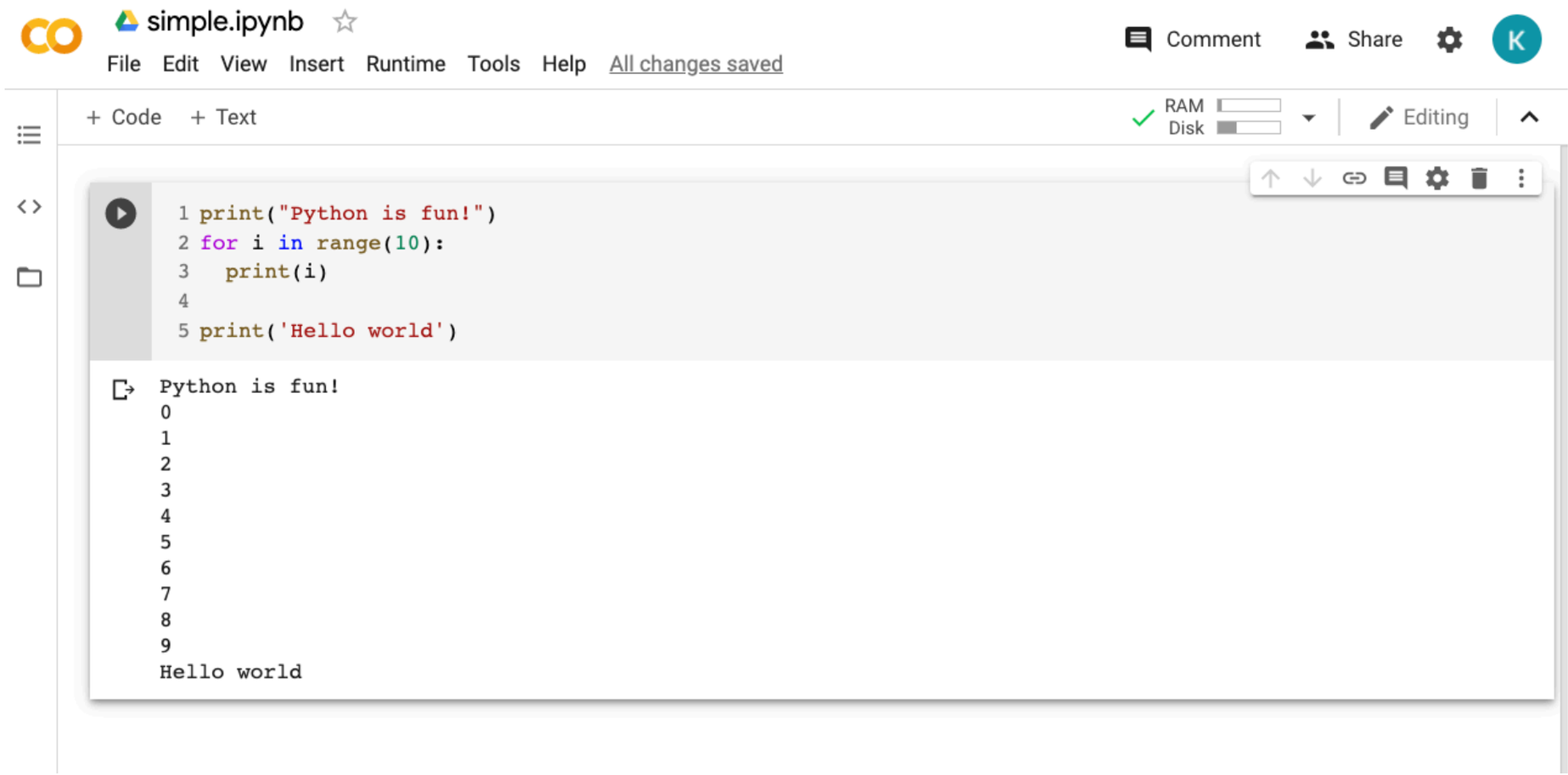
1. Anaconda

We will use Anaconda in this course. It's available from the Teaching Studio PCs. There are other Python distributions but Anaconda is one of the most popular.



2. Google Colab

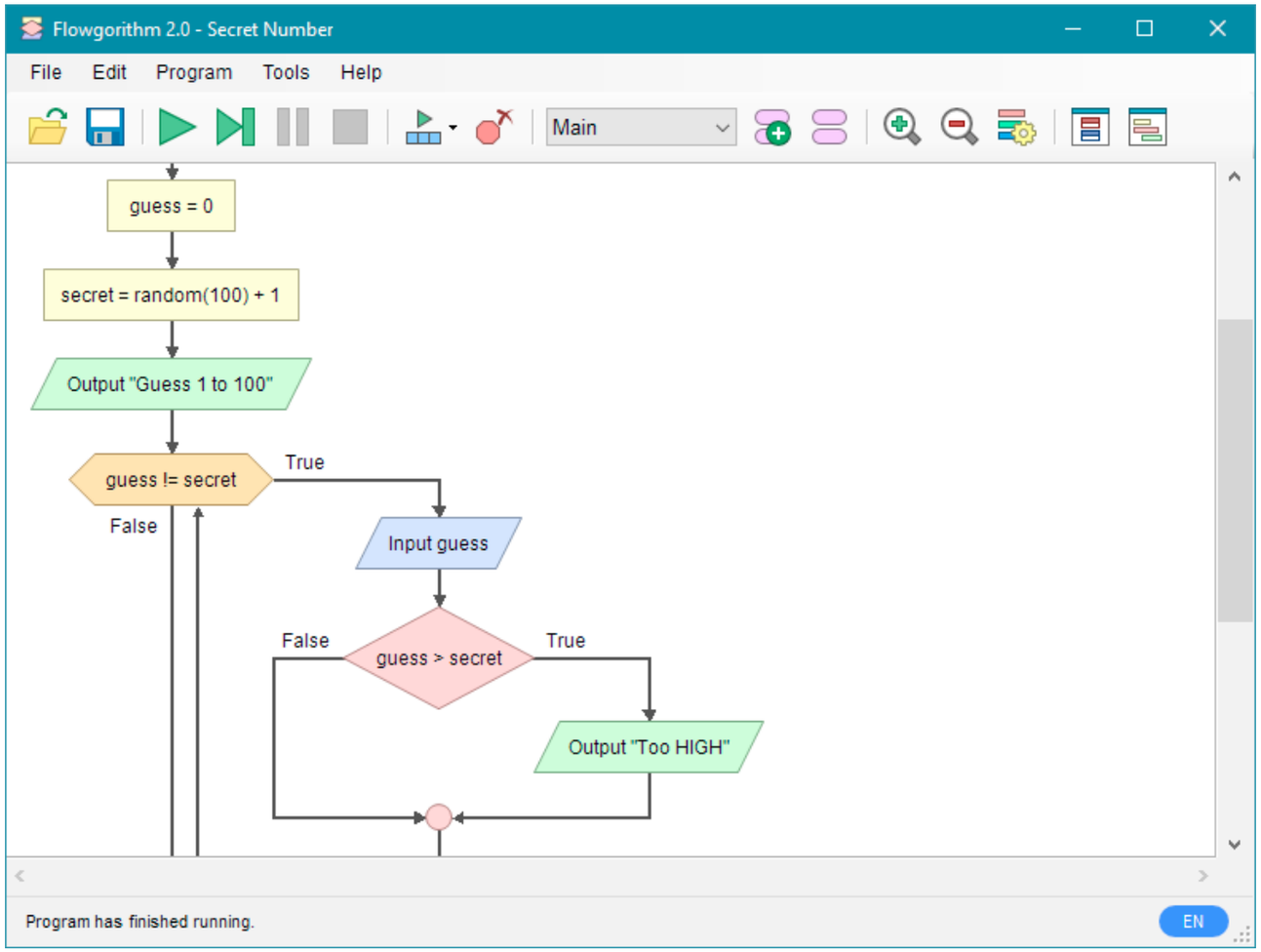
Anaconda is a big program that takes time to be installed. As an alternative you can use Google Colab. However, it's missing features that are present in Anaconda.

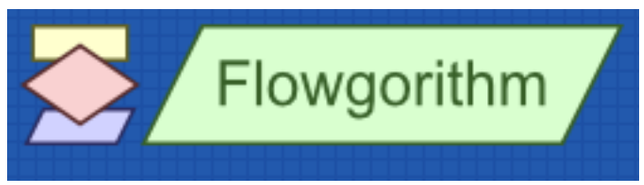


Another disadvantage of Colab is that your programs are run on Google's servers, which slows things down a bit. *In principle, you can write programs on your phone!*

3. Flowgorithm

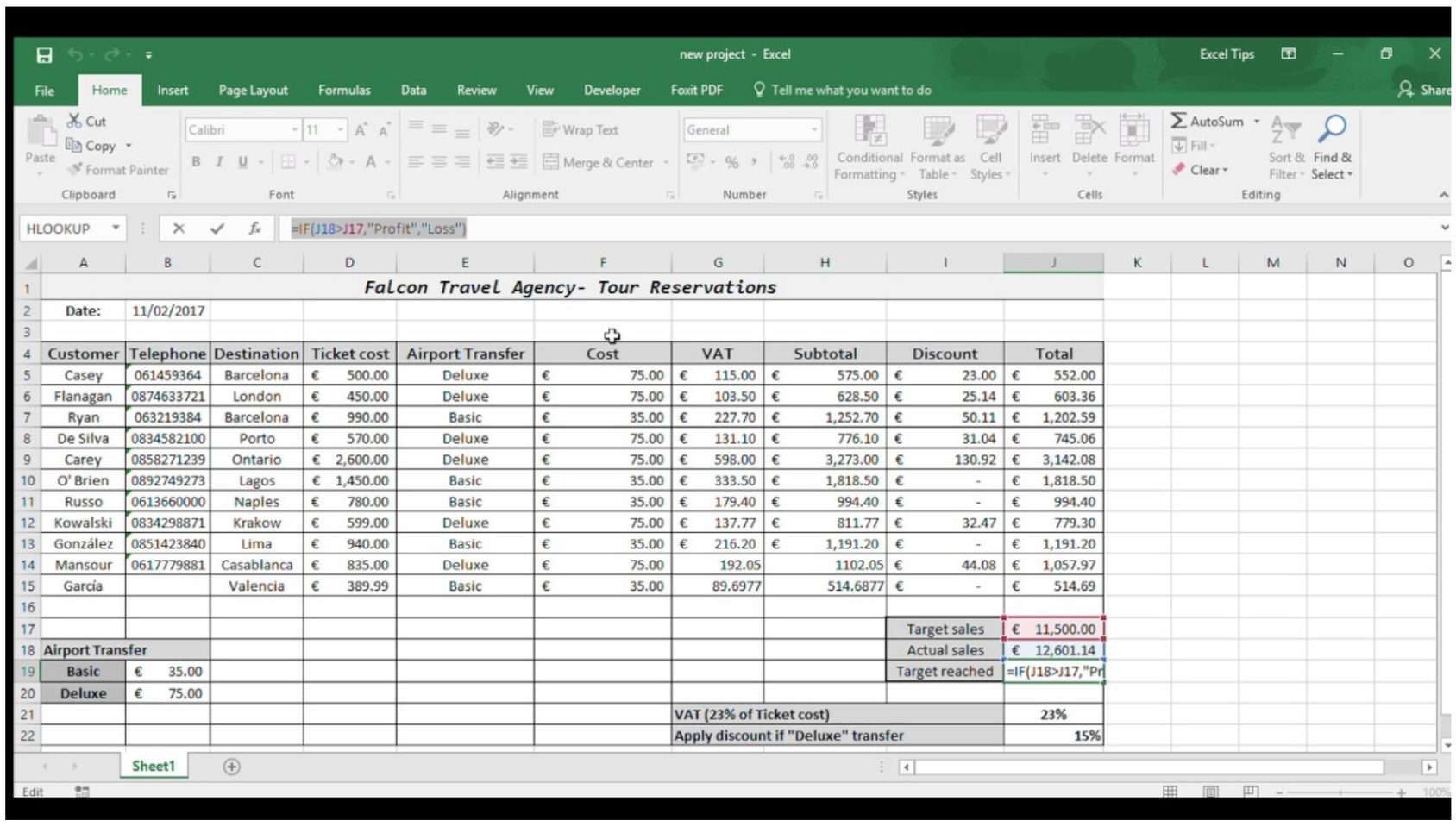
During the first half of the course, we will also learn a very simple programming language, Flowgorithm, that introduces basic programming concepts. It will help you learn Python more quickly.





4. Excel

We will also ***briefly*** review the use of Excel, which is a spreadsheet. Excel is used mostly for data analysis. However, it can also be used for programming.



3. Syllabus and learning activities

Course Outline

Part 1: Introduction to computing

Part 2: Elements of Python programming

Section 1: Data and variables

Section 2: Data structures

Section 3: Branching or decision making

Section 4: Loops

Section 5: Functions

Flowgorithm

Part 3: Basic Python programming

Section 1: Modules

Section 2: Structure of a Python program

Section 3: Good programming practices

Part 4: Python for science and engineering

Section 1: File input and output

Section 2: NumPy and SciPy

Excel

Course Outline

Part 1: Introduction to computing ~1 week

Part 2: Elements of Python programming

Section 1: Data and variables

Section 2: Data structures

Section 3: Branching or decision making

Section 4: Loops

Section 5: Functions

Part 3: Basic Python programming

Section 1: Modules

Section 2: Structure of a Python program

Section 3: Good programming practices

Part 4: Python for science and engineering

Section 1: File input and output

Section 2: NumPy and SciPy

Course Outline

Part 1: Introduction to computing

Part 2: Elements of Python programming

Section 1: Data and variables

Section 2: Data structures

Section 3: Branching or decision making

Section 4: Loops

Section 5: Functions

~6-7 weeks
(up to midterm)

Part 3: Basic Python programming

Section 1: Modules

Section 2: Structure of a Python program

Section 3: Good programming practices

Part 4: Python for science and engineering

Section 1: File input and output

Section 2: NumPy and SciPy

Course Outline

Part 1: Introduction to computing

Part 2: Elements of Python programming

Section 1: Data and variables

Section 2: Data structures

Section 3: Branching or decision making

Section 4: Loops

Section 5: Functions

Part 3: Basic Python programming

Section 1: Modules

Section 2: Structure of a Python program

Section 3: Good programming practices

~2-3 weeks

Part 4: Python for science and engineering

Section 1: File input and output

Section 2: NumPy and SciPy

Course Outline

Part 1: Introduction to computing

Part 2: Elements of Python programming

Section 1: Data and variables

Section 2: Data structures

Section 3: Branching or decision making

Section 4: Loops

Section 5: Functions

Part 3: Basic Python programming

Section 1: Modules

Section 2: Structure of a Python program

Section 3: Good programming practices

Part 4: Python for science and engineering

Section 1: File input and output

Section 2: NumPy and SciPy

~1-2 weeks

Structure of classes

Starting next week, classes will be structured as follows:

- Computational labs will be held on Tuesdays. They will run from **12:00-13:30.** You will do practice quizzes and lab exercises. You will be able to ask questions before writing a *Weekly Concept Quiz* on the *current week's material*.

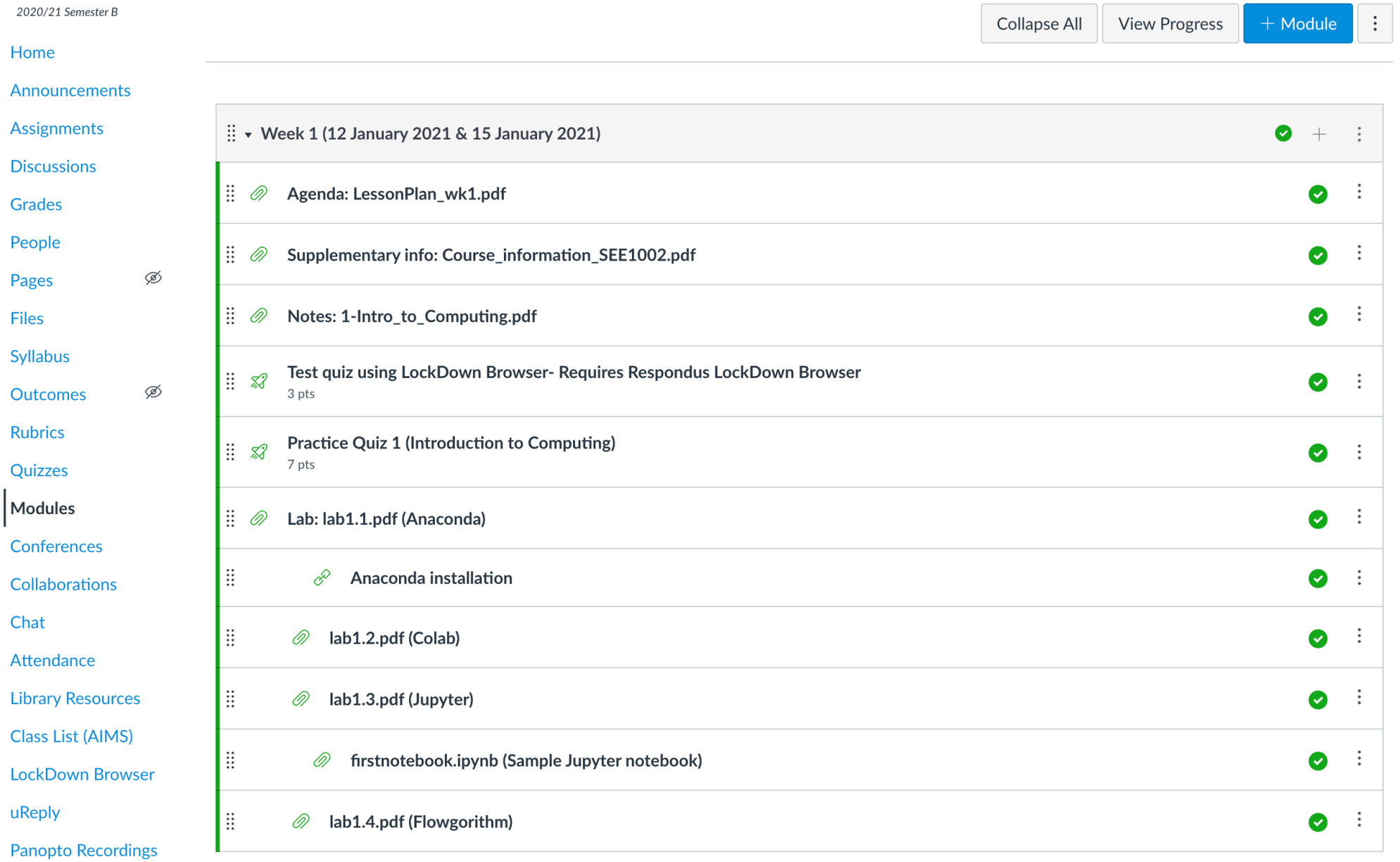
- Tutorials will be held on Thursdays. The quizzes and labs will be reviewed and the current week's material will be introduced. They will run from **15:00-16:30.** There will be a *Weekly Programming Quiz* based on the current week's material.

Formal lectures will be de-emphasized. The overviews will not cover everything. A complete set of video lectures will be uploaded.

i) Notes

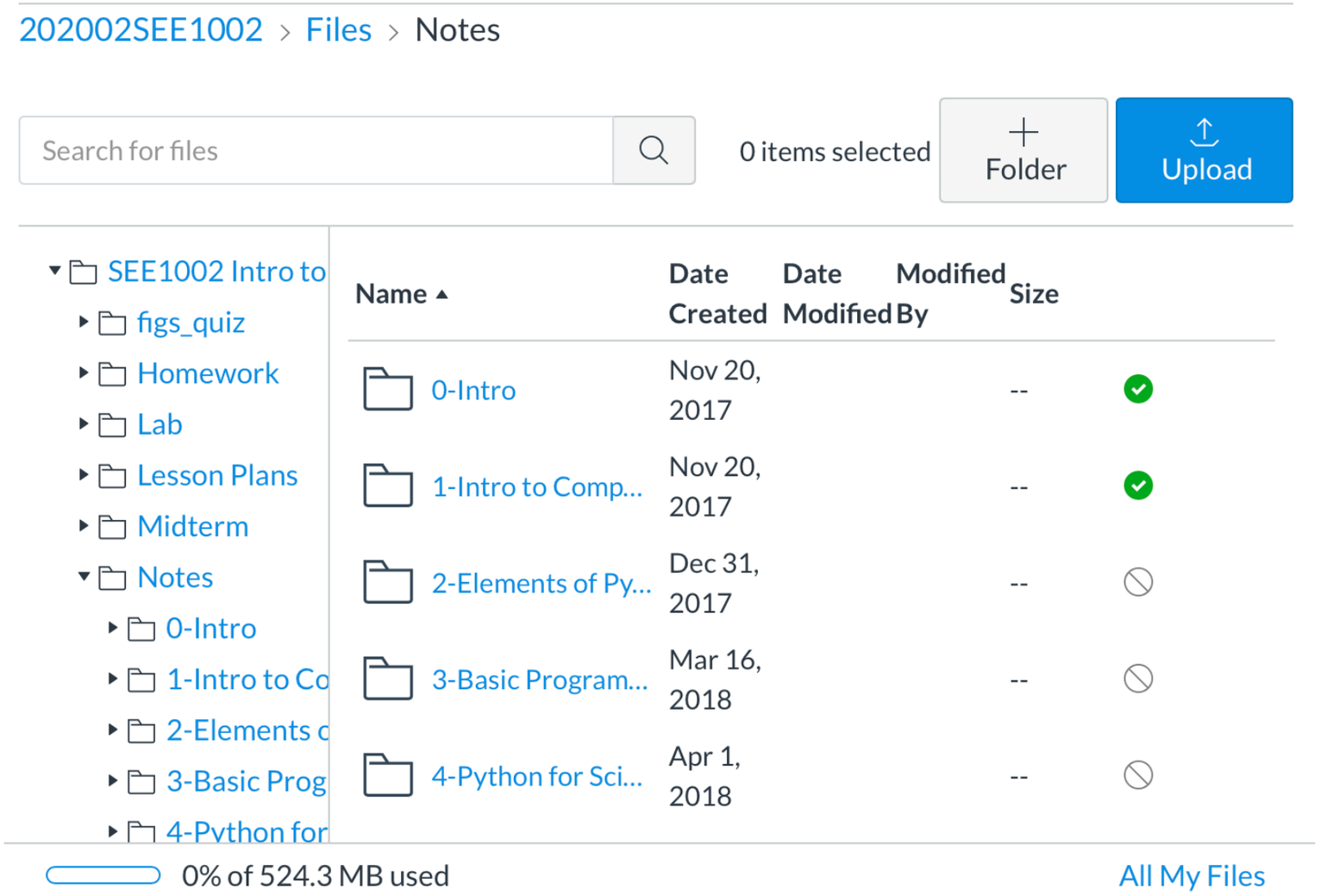
- Lecture notes will be uploaded to Canvas in advance of the Tuesday class.
- On Thursday, the most important points from the notes will be reviewed. There will also be an overview of the next week's notes.
- The lecture notes will be organised by section, one file per section.
- Lecture notes may be updated after they are posted. (*However, the changes will be minor.*)

Basic Canvas navigation



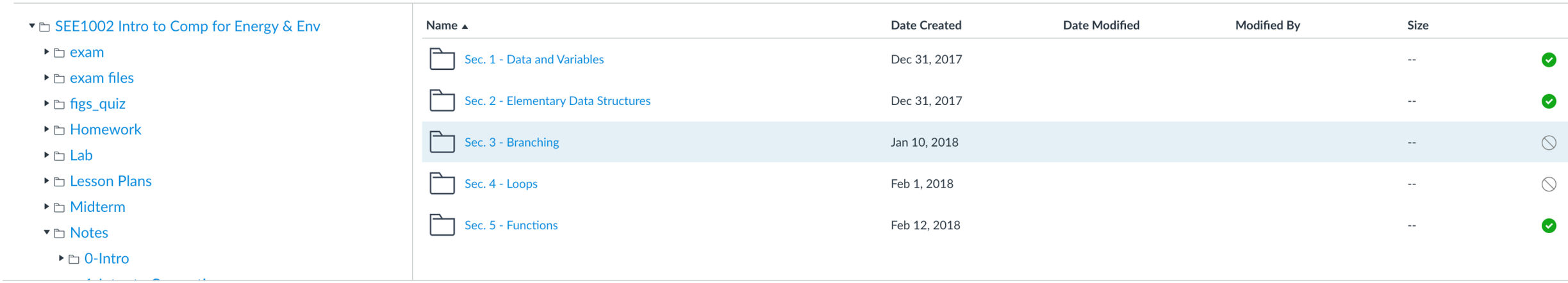
Look under ‘modules’ to find the material for a specific class.

Canvas files



Sometimes looking under ‘files’ may be more convenient.

Organisation



Material will be organised by section.

Examples

- In the notes many examples are covered. You should try to understand them thoroughly.
- *Recommendation*: try the examples on your own!

Example 8: logical operators

```
In [28]: weekday=True
```

```
In [29]: weekend=False
```

```
In [33]: not weekday
Out[33]: False

In [34]: not weekend
Out[34]: True
```

```
In [35]: weekday and weekend
Out[35]: False

In [36]: weekday or weekend
Out[36]: True
```

ii) Video lectures

- Although formal lectures will not be given, video lectures will be uploaded.
- You can watch them ahead of time (***recommended***!) or during the lab.
- *Note*: video lectures will start with Section 2.

iii) Labs

- In addition to quizzes and examples included with the notes, there will be separate labs.
- In these labs you are required to write programs.

You will not be required to submit lab reports. However, questions on the quizzes, midterm and final will be based on the lab exercises.

SEE1002
Introduction to Computing for Energy and Environment

Part 2: Elements of Python programming

Sec. 1: Data and variables

SEE1002 Computer Lab

Sec. 2: Elements of Python Programming
Lab 2.1a: Variables in Flowgorithm

In this section we will learn how to do some basic operations in Flowgorithm.

1 Declaring a variable in Flowgorithm

In Flowgoritm we must declare a variable before we can use it. This amounts to choosing the data type. This is the procedure:

1. Create a new Flowgorithm program by clicking on `File/New`.
2. Open the shape selector by clicking on the line between `Main` and `End`.
3. Select `Declare`.
4. Double-click on the `Declare` box. This will bring up a `Declare Properties` window. See Fig. 1.
5. Click on the drop-down menu for `integer`. Note the different choices that are available to you. For concreteness choose the integer type.
6. Next you need to name the variable. Call it `i`. Your Flowchart will now look like Fig. 2

2 Question 1: Assigning a variable in Flowgorithm

Once a variable has been declared, we can assign a value to it.

1. Click on the line below the `Declare` box, open the shape selector, and select `Assign`.
2. Double click on the `Assign` box and enter the value in the right-hand box and the variable in the left-hand box.
3. For concreteness now store the value `1` in the variable `i`. Your Flowchart will now look like Fig. 3

File Edit View Run Kernel Tabs Settings Help

/ ... / midterm / code /

Name

- msc_solutions.ipynb
- phd_solutions.ipynb
- Untitled.ipynb
- Untitled1.ipynb

lab2.1b.ipynb

Markdown

Python 3

SEE1002 Computer Lab

Sec. 2: Elements of computer programming

Lab 2.1b: Basic Python data types and operations

Note: this lab can be completed using Spyder, Jupyter Notebook or Google Colab.

---

1. Basic Python data types

It's easy to check on the data type of a value. With Spyder, we can use the ? operator; however, it doesn't work with Jupyter. However, we can use the `type` function.

[5]:
```
type(1)
```

[5]:
```
int
```

If you'd like to check on more than one variable, use `print`. The output is a bit strange, but don't worry about this.

[6]:
```
print(type(1.0))
print(type('This is our first lab'))
print(type(True))
```

```
<class 'float'>
<class 'str'>
<class 'bool'>
```

Question 1.1 What are appropriate data types for the following values?

1. -34524523
2. It's more fun to compute!
3. 5.34×10^3
4. Python is fun
5. False
6. 7.2e-5

0 1 Python 3 | Idle

Mode: Command Ln 1, Col 1 lab2.1b.ipynb

Jupyter Notebook

iv) Online **practice** quizzes

- In order to learn any language, you need lots of practice.
- Each section will include several online practice quizzes covering most of the material.
- They are meant to give you more practice. They will not count for marks.
  - Questions are chosen randomly.
  - You can take a quiz as many times as you like.

Quiz 1 (Introduction to Computing)

This is a preview of the published version of the quiz

Started: Jan 14 at 10:16pm

Quiz Instructions

Question 1 1 pts

A program is modified by changing its data.

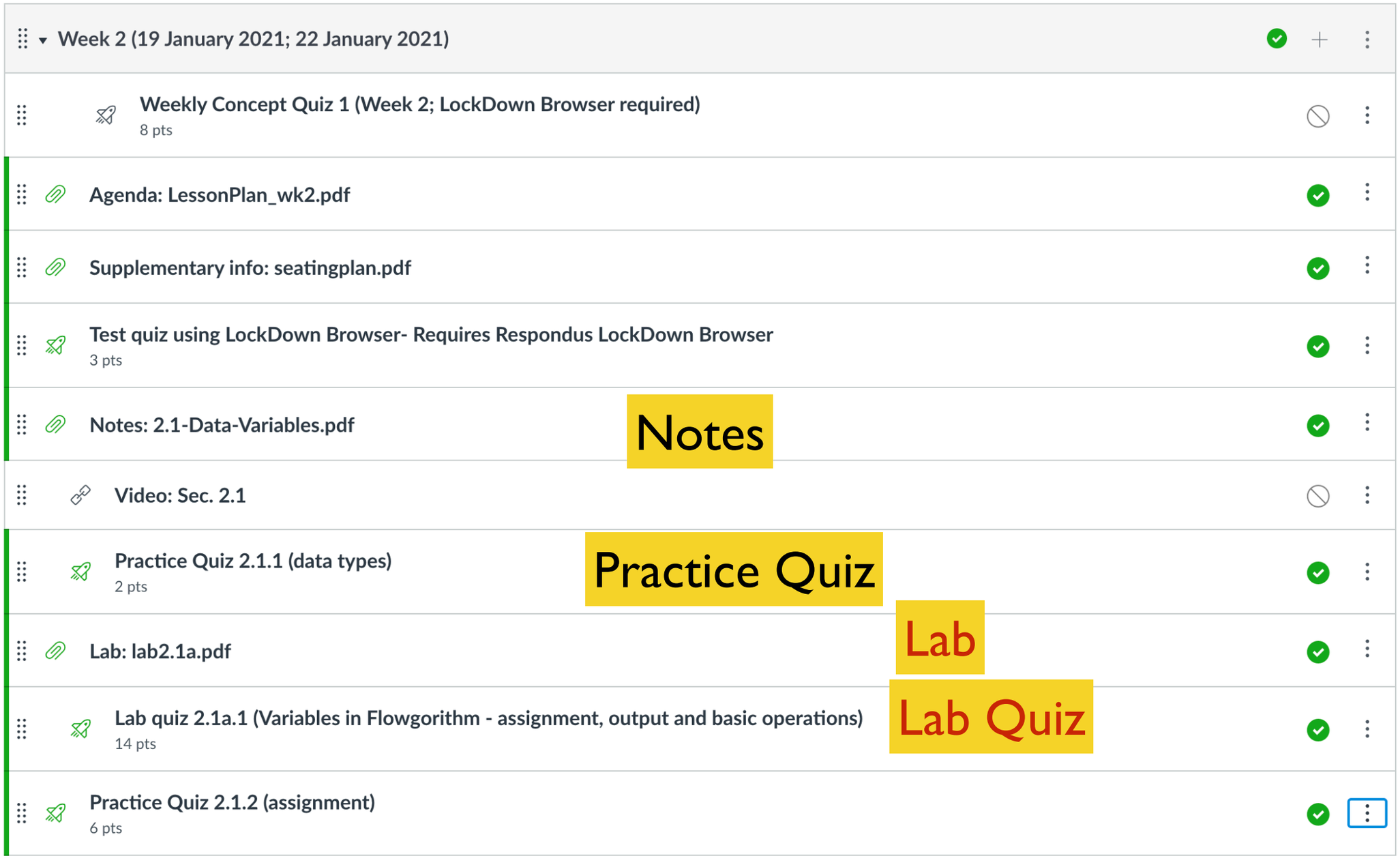
- True
- False

Next ▸

Quiz saved at 10:16pm Submit Quiz

Content

Note that there are separate practice quizzes *on the notes and the labs*.



v) Weekly Concept Quiz (*for marks*)

A Weekly Concept Quiz will be held during the Tuesday lab.

- They will be largely based on the practice quizzes.
- They will test your knowledge of basis concepts (e.g. definitions and syntax)
  - These quizzes will be administered via the Lockdown Browser.

- Duration: around 10 minutes
- First online quiz: next Tuesday

Online quizzes will focus on the ***current week's material****.*

vii) Weekly programming quiz

A Weekly Programming quiz will also be held during the Thursday class.

- **They will also test your ability to write short programs (based on lab exercises)**
- They will emphasize comprehension of the notes and labs.
- The LockDown Browser will also be used.

- Duration: around 15 minutes
- **First programming quiz: Week 3 *(to be confirmed)***

Programming quizzes will focus on the ***current week's material****.*

LockDown Browser

If you haven't already done so, please download a copy of LockDown Browser:

- Windows download
- MacOS download
- Installation guide

With LockDown Browser, you will not be able to use any other applications on your computer.

v) Lesson plan

- A lesson plan will be uploaded for each class. It's similar to what you will find on Canvas, but additional notes will be included.
- It describes the material you will be responsible for learning during the week in question:
  - Videos
  - Lecture notes
  - Practice quizzes
  - Labs
- You are free to complete the items in any order you like, but it's recommended that you follow the order as written.

For ambitious students: the following week's materials will also be available on Canvas.

Plan for Week 2
Monday

Recommended order

1. **Sec 2.1.1 - Basic Data Types**
   - Watch 2.1.1.mp4;
   - Do Quiz 2.1.1
2. **Lab 2.1a: Variables in Flowgorithm**
   - Do Lab Sections 1-5
   - Do Lab Quiz 2.1a.1 (Section 2)
   - Do Lab Quiz 2.1a.2 (Sections 3-5)
3. **Sec 2.1.2 - Assignment**
   - Watch 2.1.2.mp4; Do Quiz 2.1.
4. **Sec 2.1.3 - Basic operations**
   - Watch 2.1.3.mp4; Do Quiz 2.1.3
5. **Lab 2.1a: Variables in Flowgorithm**
   - Do Lab Quiz 2.1a.3 (Flowgorithm - Python)

viii) Homework

Since there will be plenty of in-class work, homework will not be an important part of the course.

- However, the quizzes will not test your ability to write long programs.
- You will write complete programs for homework assignments.
- There will be around 4 assignments in total. You'll be given at least 1.5 weeks to complete them.
- The questions will be similar to but more difficult than lab questions.

Advice

This course should be approached differently from other courses.

- An active approach is crucial.
  - Try examples on your own
  - Do lab exercises
  - Read notes carefully
- Students who get the most out of the course ask questions.
- Lab time should not be wasted.
- Don't try to memorise the quiz answers. *You need to understand basic concepts and learn how to apply them.*

4. Assessment

Assessment details

No	Description	Weighting
1	Quizzes	25%
2	Lab participation	5%
3	Problem sets	10%
4	Midterm (1.5 hours)	20%
5	Final exam (2 hours)	40%

To pass the course students will need to achieve at least 30% on the coursework and 30% on the final examination. This is a CityU requirement.

N.B. This doesn't guarantee that you'll pass the course!

Lab participation mark

- You will be marked based on class attendance and participation.
- To make things easier for the TAs, please sit in the same seat each week.
- The seating plan will be recorded next week.

Problem sets

- Basic requirements
  - All program codes must be uploaded to Canvas before the deadline. *No Canvas submission = no marks!*
  - Detailed requirements will be given later.
- Late assignments will be penalised (25% per day).

Advice about homework

- It's important that you try your best to solve the problems by yourself.
- You won't learn very much if you always rely on others.
- If you're stuck, then ask for help. It's very easy to waste a lot of time trying to get a computer program to work…

On marking

- Getting the right answer is important but it's not the only thing.
- You also need to write a program in a reasonable way.
- In order to get partial marks, you also need to explain your logic. (*We'll explain how later how this can be done.*)

University plagiarism policy

- The University has very strict rules regarding plagiarism and academic honesty:

> "**you are expected to present your own work, give proper acknowledgement of other people's work, and honestly report your scholarly findings. Violations of academic honesty are regarded as serious offences in the University. Acts such as plagiarism and fabrication of research findings can lead to disciplinary action. Most commonly the penalty is failure in a course, but in the most serious cases expulsion from the University and debarment from re-admission may occur".**

- If you're unsure of what constitutes plagiarism please consult http://www6.cityu.edu.hk/ah/plagiarism.htm

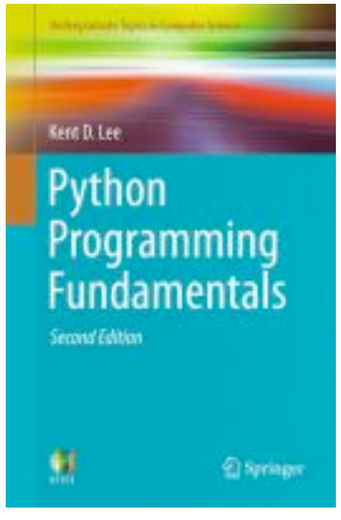
N.B. Don't copy programs written by your friends. This is easily recognised by the Turnitin software.

Plagiarism policy for online quizzes

- No communication whatsoever with neighbours is allowed.
- Violators will be penalised. Both students will automatically receive 0. Subsequent penalties will receive penalties of greater than 100%.
- Continued violations will be reported to the School's Academic Conduct Committee.

5. Learning resources

Main References



Python Programming Fundamentals by K. D. Lee (Springer, 2nd edition, 2011). [Electronic version available from the Library at]

Introduction to computation and programming using Python by J.V. Guttag (MIT Press, 2nd edition, 2016).
[electronic version available from the Library at https://julac.hosted.exlibrisgroup.com/permalink/f/1pqrrkl/CUH_IZ21413366910003408]

If any of the lecture material is unclear, consult these references first.

Other References

There are many, many references on Python. For example, you can find lots of tutorials or videos on the web.

The most comprehensive is the official Python tutorial

https://docs.python.org/3/tutorial/index.html

Online documentation is also available within Anaconda. *We will cover this later.*

General comment about reference books

- Many CityU students do not bother looking at reference books...
- But please note that Canvas lecture notes are rarely complete.
  - If you rely solely on the printed slides, you may miss out on some important information.
  - Reference books will help fill the gaps in your understanding.
- Seeing the same material presented in a different way is a good way to learn!