Programming for Scientists

COMP1730 & COMP6730

Australian
National
University

Introductions

Australian National University

- Co-convenors Semester 1, 2023:

Dan Andrews

Brian Parker

- Senior Tutors:

Malcolm MacDonald

Hancheng Shao

Alexei Khorev

Course structure

Australian National University

- Week 1 – Programming Basics, Variables and Expressions
- Week 2 – Functions, Conditionals and Iteration
- Week 3 – Strings, Lists and Dictionaries
- Week 4 – Abstraction (Canberra Day holiday on the Monday)
- Week 5 – Files and IO, Classes
- Week 6 – Libraries, Visualisation and Graphing

Dan

- Mid-semester break

- Week 7 - …

Brian

Introductory Lecture - format

Australian National University

- Orientation to python – let's look at some code first
- Learning to program
- Reference books and other reading
- Variables and Expressions (part I)

AND (at the end):

- Admin:
  - Lab class enrolments
  - Assessment
  - Other announcements

What does python code look like?

Australian National University

File: first_example_v0.2.py

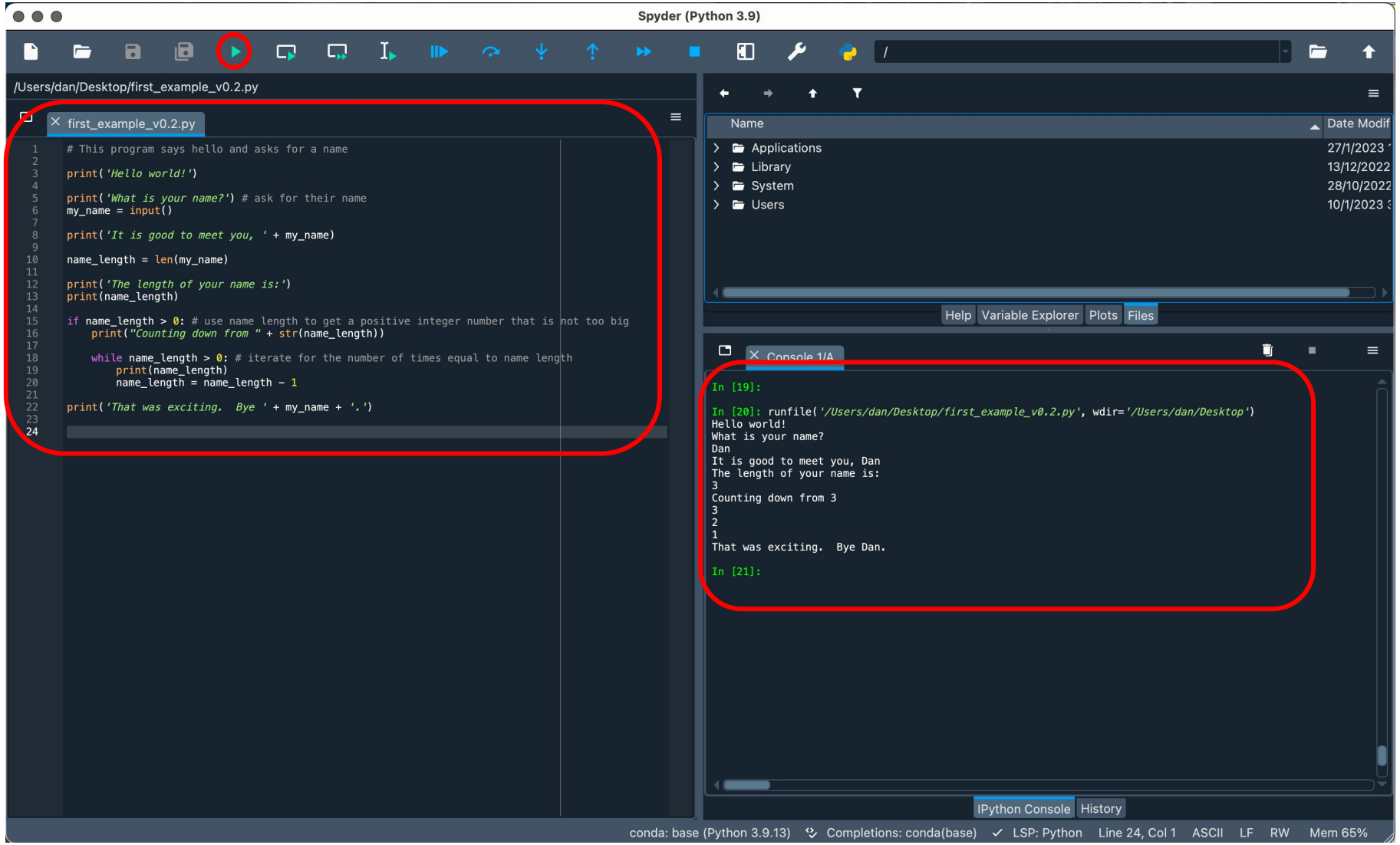
```
# This program says hello and asks for a name

print('Hello world!')

print('What is your name?')
my_name = input()

print('It is good to meet yo

name_length = len(my_name)

print('The length of your na
print(name_length)

if name_length > 0: # use na
    print("Counting down fro

    while name_length > 0: #
        print(name_length)
        name_length = name_length - 1

print('That was exciting.  Bye ' + my_name + '.')
```

```
(base) mhs-057255:Desktop dan$ python first_example_v0.2.py
Hello world!
What is your name?
Dan
It is good to meet you, Dan
The length of your name is:
3
Counting down from 3
3
2
1
That was exciting.  Bye Dan.
```

Example edited from: Sweigart, Automate the Boring Stuff with Python (2015)

Using an Integrated Development Environment (IDE):

Australian National University

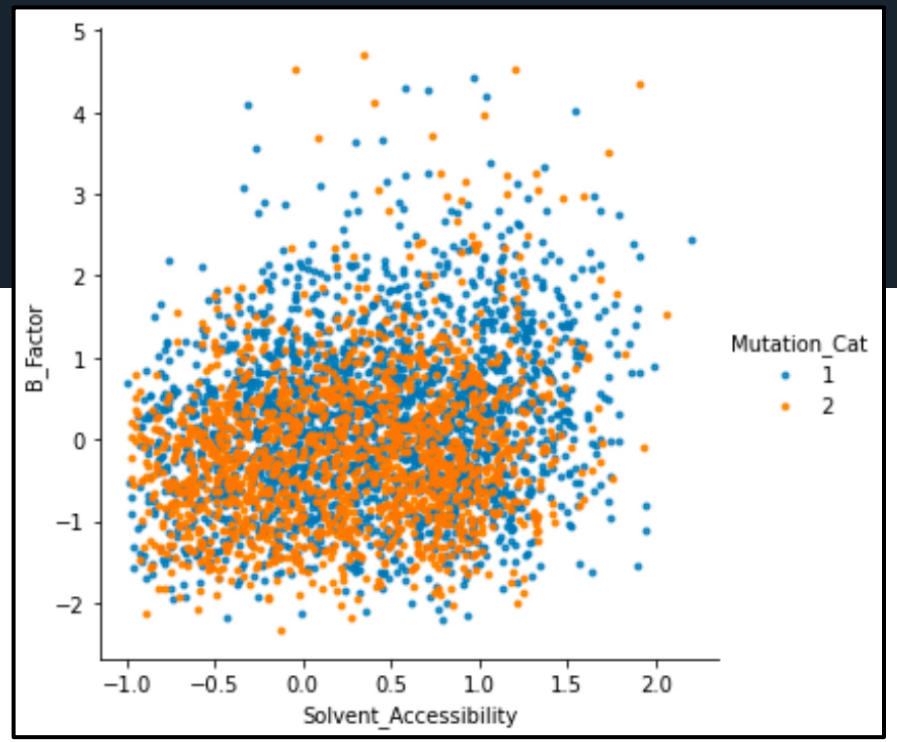


More complicated: Python as a toolbox

Australian National University

```python
import pandas as pd
import seaborn as sns
from sklearn.tree import DecisionTreeRegressor

mutations = pd.read_csv('/Users/dan/Desktop/Envision_manuscript_data/Gray_etal_SupplementaryTable_S2_cleaned.csv')

mutation_metrics = ["Residual_Function", "Solvent_Accessibility", "B_Factor"]

sns.lmplot( x="Solvent_Accessibility", y="B_Factor", data=mutations, fit_reg=False, hue='Mutation_Cat', legend=True, markers='.', x_jitter=True, y_jitter=True)

X = mutations.iloc[0:, [26, 30]].values # 26 == Solvent_Accessibility, 30 == B_Factor
y = mutations.iloc[0:,[6]].values # 6 is 'Mutation_Cat'

tree_clf = DecisionTreeRegressor(max_depth=2)
tree_clf.fit(X, y)

# Solvent_Accessibility: 91%
# B-factor: 187 (high, +ve)
prediction = tree_clf.predict([[2, 0.99]])
print("Residual function prediction: " + str(prediction[0]))
```

```
In [162]: runfile('/Users/dan/Desktop/second_example_v0.2.py', wdir='/Users/dan/Desktop')
Residual function prediction: 1.2415158371040724

In [163]:
```



What is programming?

COMP1730 & COMP6730

Reading: Textbook chapter 1 : Alex Downey, *Think Python*, 2nd Edition (2016)

Australian National University

Why Python?

Australian National University

- This is not a course on programming in python. It is a course on programming that uses python
- Why python?
  - Python is a very popular programming language
    - Especially in science and engineering
  - Open source, available on most platforms
  - Huge external code libraries for doing just about everything, in:
    - Data Science
    - Machine Learning
    - Bioinformatics…
- We will use python 3 (beware older books that are python 2)

This course

Australian National University

- Is a first course in programming, in python
  - Focused in transferable, practical skills
  - Coding languages come and they go – but the good coding practice is relevant to all languages
  - Useful to those in science and engineering.
  - Not foremost teaching commercial software engineering
- A beginners course – no prior experience required. But this doesn't mean we are going to go slowly, or that it will be easy!
- We will use python 3

Coding as a craft

Australian National University

- Some recommendations:
  - Read widely and write code frequently. Practice, practice, practice.
  - This won't end well if it is already week 9 and this is the first time you are looking at the course.
  - *Textbooks and reading*: if you only attend one part of this course, make sure it is the tutorials (actually, these are compulsory). Though, these will be very hard if you don't at least attend lectures or do the course reading
  - In the beginning, as you start to write your first programs, it might feel bad as you make all the beginners mistakes. Don't worry and keep trying. Everyone starts here.
  - Error messages are your friend…

Suggested Reading

(there is no main textbook, but…)

Course Textbooks

Australian National University

- Main ‘text’: Alex Downey (2016) *Think Python*, 2nd Edition ★
- Other good books:
  1. Al Sweigart (2015) Automate the Boring Stuff with Python ★★
  2. Bill Lubanovic (2019) Introducing Python, 2nd Edition ★★★
- When reading other python books, make sure they are python 3!
- Be careful with web resources – some are great (eg. docs.python.org). Many aren’t.

Monty Python references: ★ light ★★ frequent ★★★ every code example

https://anulib.anu.edu.au

Australian National University | LIBRARY

Search ANU web, staff & maps ▼ | 🔍

My library record

Find & access | Collections | Research & learn | Using the library | News & events | About | Services for

Level 4 of Chifley Library closed due to repair work »

SuperSearch | **Catalogue search** | Full text e-journals

Search the catalogue by keyword...

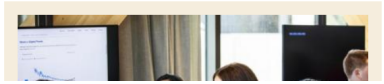
GO

advanced catalogue search »

about catalogue search »

E-resources & databases »

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z



Australian National

Australian National University | LIBRARY

Search ANU web, staff & maps ▼ | 🔍

My library record

Find & access | Collections | Research & learn | Using the library | News & events | About | Services for

» Find & access »

E-resources and databases starting with "S"

Start over | Check the Full Text E-Journals list

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

Search by: (choose a subject) GO

Search by title: GO

Show all e-resource titles GO

1 to 25 of 44

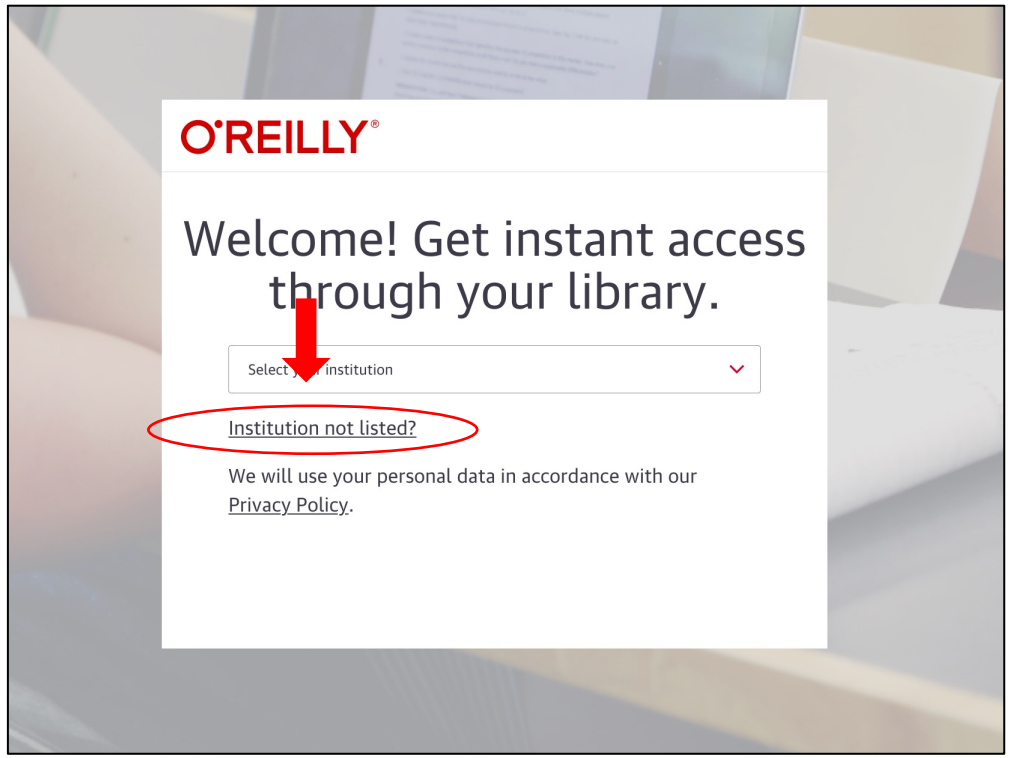
Next 19

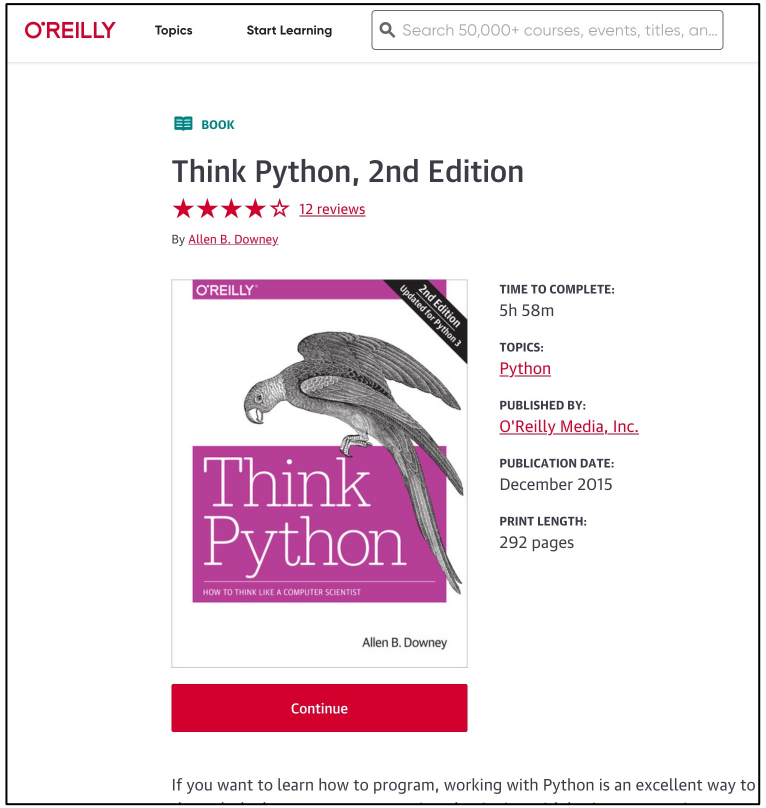
1. Safari Books Online [show details]
2. Sage Journals Online [show details]
3. Sage Knowledge (Sage Reference Online) [show details]
4. Sage Open [show details]

Safari Books

Australian National University

- https://safaribooksonline.com/library/view/temporary-access/





Variables (part I)

COMP1730 & COMP6730

Reading: Textbook chapter 2 : Alex Downey, *Think Python*, 2nd Edition (2016)

Australian
National
University

Variables – what are they?

Australian National University

- Contain a program's data whilst it is executing
- Assignment statements:

```
>>> message = 'And now for something completely different'
>>> n = 17
>>> pi = 3.141592653589793
```

Downey (2015) Think Python, 2nd Ed.

- In memory – the 'state' of the program:

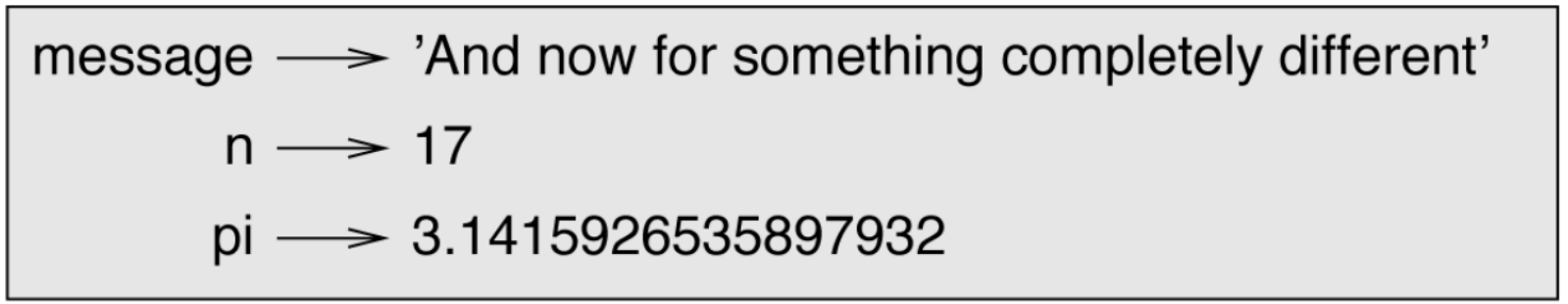


Figure 2-1. State diagram.

Types of variables (in python)

Australian National University

- All variables have a type – and you will get an error is you store an incompatible value in the wrong type (eg. a string value in an integer variable type)
- Or try to do something inappropriate with a data type (eg, print an integer as a string)
- Basic data types:
  - int – integer
  - float – decimal values
  - str – strings of one or more characters
  - bool – Boolean values, `True` or `False`
- And variables that contain multiple values of basic data types:
  - List and Tuple – sequences an index
  - Dict – a hash, key-value pairs

Table 1-2: Common Data Types

Data type	Examples
Integers	`-2, -1, 0, 1, 2, 3, 4, 5`
Floating-point numbers	`-1.25, -1.0, -0.5, 0.0, 0.5, 1.0, 1.25`
Strings	`'a', 'aa', 'aaa', 'Hello!', '11 cats'`

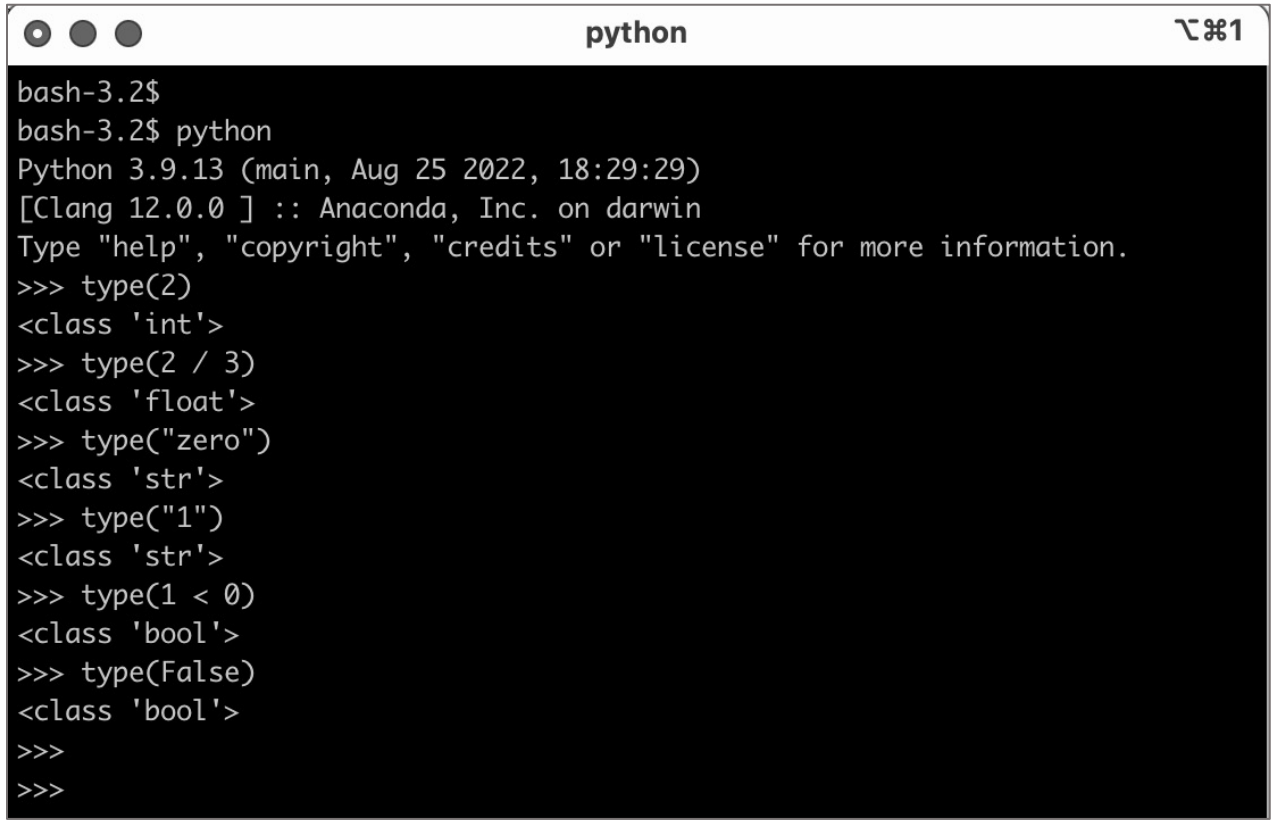
Sweigart (2019) Automate the Boring Stuff with Python, 2nd Ed.

Every variable has a type

Australian National University

- Variable types in python:
  - Integers (type `int`)
  - Floating-point numbers (type `float`)
  - Text strings (type `str`)
  - Truth or Boolean values (type `bool`)
- Variable types determine what we can do with values (and sometimes what the result is)

- The `type()` function tells us the type of a variable:



Terminal or iTerm programs on a Mac

Numeric types: `int`

Australian National University

- `int` types represent the mathematical integers (positive and negative whole numbers) (0, 1, 2, -1, -17, 4096,…)
- Values of type `int` have no inherent size limit in python

```
>>>
>>> 2 ** (2 ** 2)
16
>>> 2 ** (2 ** (2 ** 2))
65536
>>> 2 ** (2 ** (2 ** (2 ** 2)))
200352993040684646497907235156025575044782547556975141926501697371089405955631145308950613088093334810103823434290726318182294938211881266886950636476154702916504187191635158796634721944293092798208430910485599057015931895963952486337236720300291696959215610876494888925409080591145703767520850020667156370236612635974714480711177481588091413574272096719015183628256061809145885269982614142503012339110827360384376787644904320596037912449090570756031403507616256247603186379312648470374378295497561377098160461441330869211810248595915238019533103029216280016056867010565164675056803874152946384224484529253736144253361437372908830379460127472495841486491593064725201515569392262818069165079638106413227530726714399815850881129262890113423778270556742108007006528396332215507783121428855167555407334510721311242739956298271976915005488390522380435704584819795639315785351001899200002414196370681355984046403947219401606951769015611972698233789001764151719005113346630689814021938348143542638730653955296969138802415816185956110064036211979610185953480278716720012260464249238511139340046435162386756707874525946467090388654774348321789701276445552940909202195958575162297333357615955239488529757995402847194352991354376370598692891375715374000198639433246489005254310662966916524341917469138963247656028941519977547770313806478134230959619096065459130089018888758808473362595606544488850144733570605881709016210849971452956834406197969056546981363116205357936979140323632849623304642106613620022017578785185740916205048971178182040018728293994344618622432800983732376493181478984811945271300744022076568091037623999203492023906626264491909167985461515778839060397720759279378852241294301017458086862263369284725851403039615558564330385450688652213114813638408384778263790
```

- Note: can’t use commas to format integers for readability
  - Write `128736` not `1,282,736`

Numeric types: `float`

Australian National University

- Floating-point numbers (type `float`) approximate the mathematical real numbers
- Values of type float have limited range and limited precision
  - Min/max ± 1.79 x 10^{308}
  - With a few exceptions to this limit
  - Though this is the typical limit – the actual limits depend on the python implementation
- Type `float` also has special values ± `inf` (infinity) and `nan` (not a number)

String variables

Australian National University

- Strings (type `str`) represent text
- A string literal is enclosed in single or double quote marks

```
>>> "Hello world"
'Hello world'
>>> '4" long'
'4" long'
```

- A string (in python) can contain other types of quote mark, but not the one used to **delimit** it
- More about strings (so much more) in a coming lecture

Exercises

Australian National University

- Complete Exercises 2-1 and 2-2 on Page 18 & 19 of Think Python.

Reading

- Chapter 1 & 2 of *Think Python* AND/OR
- Chapter 1 of *Automate the Boring Stuff with Python*

Course Organisation

COMP1730 & COMP6730

Australian
National
University

Course Admin, Information and Contacts

Australian National University

- https://comp.anu.edu.au/courses/comp1730/
- Wattle for announcements, forums, quizzes, surveys and assignment submission
- Read the Wattle news and announcements!
- To ask a question:
  - Use the discussion forum on Wattle
  - Ask your tutor in labs
  - Use the Teams channel during lectures
  - For private matters, use the course email: comp1730@anu.edu.au
  - Always use your ANU email address, to avoid the spam filters

Schedule overview

Australian National University

- Two lectures per week
- All lectures will be presented live and will be recorded
- Follow content and schedule: https://comp.anu.edu.au/courses/comp1730/content/
- One 2-hour lab per week, starting from Week 2
  - **Before Fri 24th Feb - Sign-up for a lab class** with MyTimetable (linked via Wattle): https://mytimetable.anu.edu.au/odd/student
- Assessments will be due at **midnight on Saturday** of weeks when due
- You are expected to spend another 6 hours per week studying the course:
  - doing the recommended reading
  - solving all lab exercises, and
  - time spent to practice coding

More goodies

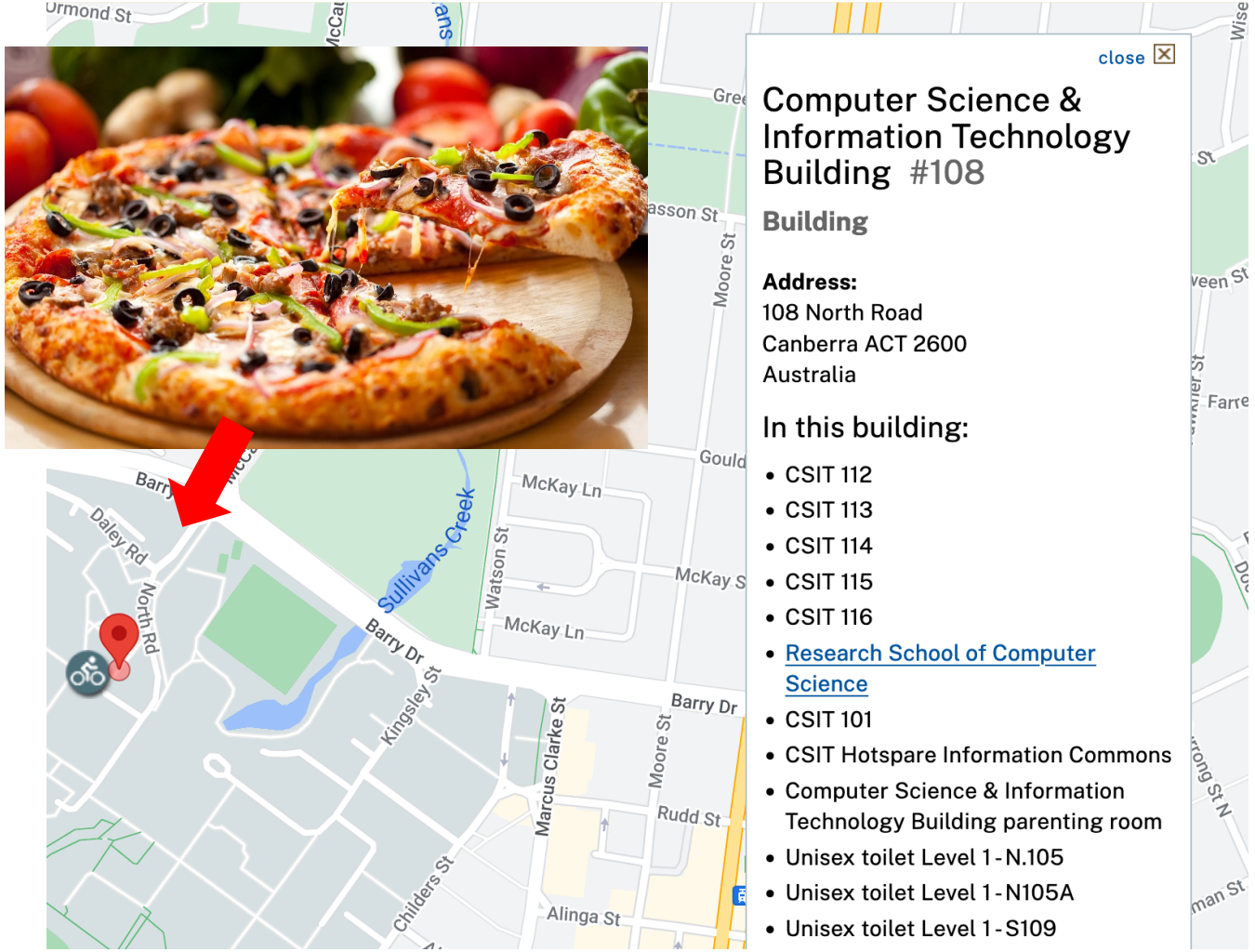
Australian National University

- As of this semester, we are trialing weekly drop-in sessions for 1-to-1 tutor contact:
  - Weds – 10-11am – CSIT Building (#108), room N111
  - Weds – 1-2pm – CSIT Building (#108), room N115/116
  - Thurs – 4-5pm – CSIT Building (#108), room N111
  - Fri – 3-4pm – To Be Advised
- For in-lecture questions and answers, use the Teams group set up for COMP1730/6730
  - Alexei will be on hand to answer specific questions about the lecture content.

CSSA Install-Fest!

Australian National University

- What: Week 1 Python Install-Fest
  - Get your python install working with experienced help
  - A COMP1730 tutor (Jon) will be on hand especially
- When: Week 1 - Friday 24th Feb at 5pm
- Where: CSIT Building, rooms N109, N112 from 5PM onwards and N113 and N114 after 6PM



Assessment (preliminary)

Australian National University

- Final exam – 50%
- Individual assignment – 30%
- Homework assignments – 15%
- Lab participation – 5%

- Individual assignment is a take-home programming assignment
- There will be a *viva*/oral component of the assignment assessment
  - Held during a lab session at end of semester
  - Will require you to discuss your solution, to decide your mark
- The assessment scheme will be final at end of week2. Any changes will be announced.

Week	Assessment
2	Homework 1 released & due
3	Homework 2 released & due
5	Homework 3 released & due
6	Assignment released
8	Assignment due
9	Assignment *viva* (done in labs)
10	Homework 4 released & due
11	Homework 5 released & due
Examination period	Final exam

Check Wattle for updated information

Changed assessment for S1 2023

Australian National University

- There will be a final exam, but the form that takes is still unfinalised
- There is going to be a *viva* component (in Week 9) of the assignment assessment
  - Each student will have an additional verbal assessment of their assignment work that will inform their mark. This will be 25% of the final assignment mark.
  - Students are expected to have a thorough knowledge of their own work and be able to speak in detail about their answers and solutions

Useful Links:

Australian National University

- Install python - if you want to start, follow the instructions to install python (via Anaconda) on your laptop: https://comp.anu.edu.au/courses/comp1730/labs/install/

- Lab materials - this is where to find the labs: https://comp.anu.edu.au/courses/comp1730/labs/

- And the assessment description: https://comp.anu.edu.au/courses/comp1730/assessment/

Wattle Discussion forum

Australian National University

In general, this is where you should go to ask questions

3 simple rules:

1. Read before you post.
   - Before posting a question, check if your question has already been answered

2. Give you post a good, descriptive topic
   - Don't write '*A question*'. Write something like '*Variable assignment: why does the value not change?*'

3. You **may not post** solutions to assignment problems

These rules are good etiquette and apply to any online forum.

Assessment

Australian National University

- All assignment deadlines are hard – no late submissions will be accepted. Unless previous permission has been granted.
- Regarding deferred assessments and special consideration, please read: https://www.anu.edu.au/students/program-administration/assessments-exams
- Please note that "*any submitted work may be subject to an additional oral examination*", which can change the assessment mark in any way.

Academic honesty

Australian National University

- Submitted code will be checked computationally for evidence of plagiarism.
- If evidence of plagiarism is found in individual homework problems, the mark for that individual homework will not be posted, until all homeworks have been assessed. In the context of all homeworks if it is decided there is evidence of repeated plagiarism, students will be interviewed for possible action of academic misconduct.
- The take-home assignment and exam will also be checked for evidence of academic misconduct.
- **What is okay**: for the homework, discussing the programming problems and approaches to solve them with other students is allowed, provided that no code is exchanged and that the final solution and code is written individually. In this case, the other students involved in the discussion must be listed in a comment at the top of the homework.
- For the final exam and take-home assignment must be individual work. You may not discuss the questions or your answers with anyone (this includes any on-line forum).
- Note that in all cases every line of code submitted must be fully written by you from scratch (and not just a modified copy of a version from the internet), and must be fully understood and explainable by you. Sufficient inline comments should be provided to make clear that you understand the code.
- Note on large language models and other code generators: generative AI models such as github copilot, chatGPT, Bing chatbot etc can be used by students for the homeworks and take-home assignment to explore solutions and understand their own code. They will not be allowed for the final exam. But in all cases the final code submitted by the student must be fully written and understood by the student, as described above.
- If you are unsure, please ask your tutor or the convenors.

Academic Honesty – Policy:

Australian National University

If we find evidence of cheating or copying of work, we will:

1. Review the potential copied work
2. Raise a flag on the student work as ‘under investigation’
3. Further evidence of cheating for a given student will result in all homework and/or assignment marks being set to 0 for that student

Studying remotely

Australian National University

- All course material and lectures will be available online
- Online options for labs are available, in addition to in-person ones
- If you are in Canberra, it is strongly recommended that you attend in-person

Important To-Dos (https://comp.anu.edu.au/courses/comp1730/news/2023/02/10/Important-TODOs/):

Australian National University

1. Complete the **demographic information questionnaire** on course Wattle page
2. Sign up to a **lab class!**
   - Do this via myTimetable: https://mytimetable.anu.edu.au/odd/student
   - Link also accessible from Wattle page
   - Do this by end of week 1 (say Fri 24 Feb)
   - Labs start in week 2
   - In-lab assessment starts in week 3
3. Complete the On/Off Campus declaration on course Wattle page
4. Login to STREAMS: https://cs.anu.edu.au/streams/
5. **Prepare for the labs!** Attend lectures, read lab instructions – and attempt some of the exercises before attending your lab
6. Make sure you have a working python programming environment:
   - Install Anaconda on your own computer
     - Go to https://www.anaconda.com/products/distribution
     - Current installation will give you python 3.9 or later
     - Includes that Spyder IDE as part of installation
   - Or, install another python3 implementation
   - Or, verify that you can reliably use the VDI
7. For more information, read: https://comp.anu.edu.au/courses/comp1730/labs

CECC Class Representatives

Class Student Representation is an important component of the teaching and learning quality assurance and quality improvement processes within the ANU College of Engineering and Computer Science (CECC).

The role of Student Representatives is to provide ongoing constructive feedback on behalf of the student cohort to Course Conveners and to Associate Directors (Education) for continuous improvements to the course.

Roles and responsibilities:

- Act as the official liaison between your peers and convener.
- Be available and proactive in gathering feedback from your classmates.
- Attend regular meetings, and provide reports on course feedback to your course convener
- Close the feedback loop by reporting back to the class the outcomes of your meetings.

• Why become a class representative?

- **Ensure students have a voice** to their course convener, lecturer, tutors, and College.
- **Develop skills sought by employers**, including interpersonal, dispute resolution, leadership and communication skills.
- **Become empowered**. Play an active role in determining the direction of your education.
- **Become more aware of issues influencing your University** and current issues in higher education.
- **Course design and delivery.** Help shape the delivery of your current courses, as well as future improvements for following years.

- Note: Class representatives will need to be comfortable with their contact details being made available via Wattle to all students in the class.

- For more information regarding roles and responsibilities, contact:
- ANUSA CECC representatives: sa.cecs@anu.edu.au

Want to be a class representative? Nominate today!

Please nominate yourself to your course convener by end of Week 2, Sem 1, 2023.

You are free to nominate yourself whether you are currently on-campus or studying remotely.

Interested? Write to comp1730@anu.edu.au or talk to us after the lecture.