



Curtin University

Revision

ISYS2001, School of Marketing and Management

Today

- Review ULO
- Review Lectures
- Discuss Extended Learning Portfolio
 - Questions
 - Study Strategies

ULO

On successful completion of this unit students can:		Graduate Capabilities addressed
1	Formally state a problem, devise an algorithm to solve the problem, and translate the algorithm into a program using an appropriate programming language	 
2	Explain the use of different data types and algorithm constructs (sequence, selection, repetition) for different purposes	
3	Apply standard techniques and notations to develop document and apply a program testing strategy	
4	Demonstrate the use of a programming environment (integrated development environment- IDE) to edit, execute, and debug programs	

Curtin's Graduate Capabilities

	Apply discipline knowledge, principles and concepts		Innovative, creative and entrepreneurial		Effective communicators with digital competency
	Globally engaged and responsive		Culturally competent to engage respectfully with local First Peoples and other diverse cultures		Industry connected and career capable
Find out more about Curtin's Graduate Capabilities at the Curtin Learning and Teaching website: clt.curtin.edu.au					

Computational Thinking



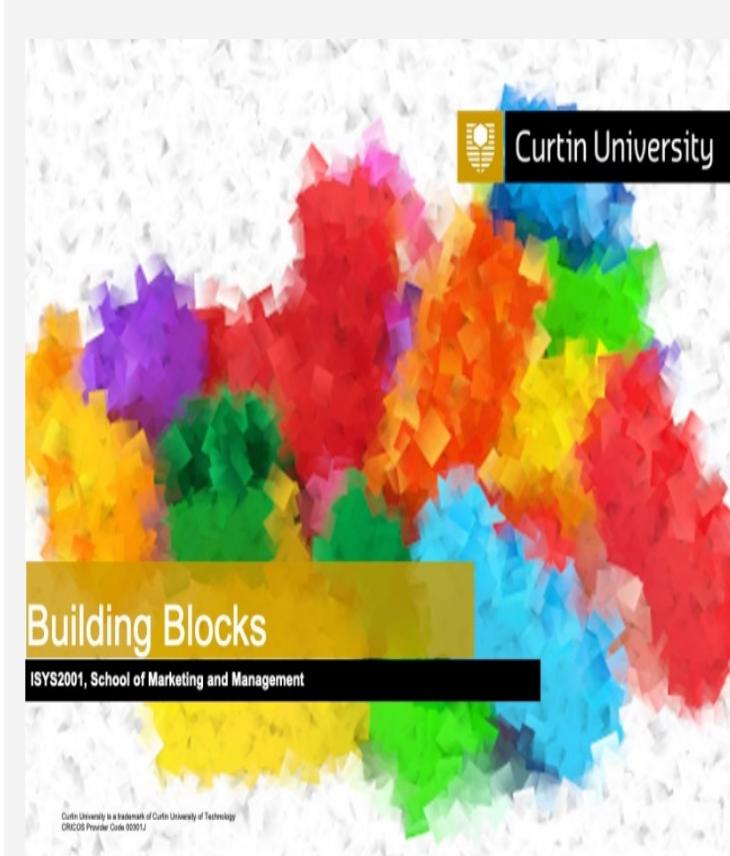
This week we introduce some fundamental concepts that we will explore this semester. This is to build an intuitive understanding of programming. In the workshop, we explore the programming environment and get a small taste of python.

Problem Solving



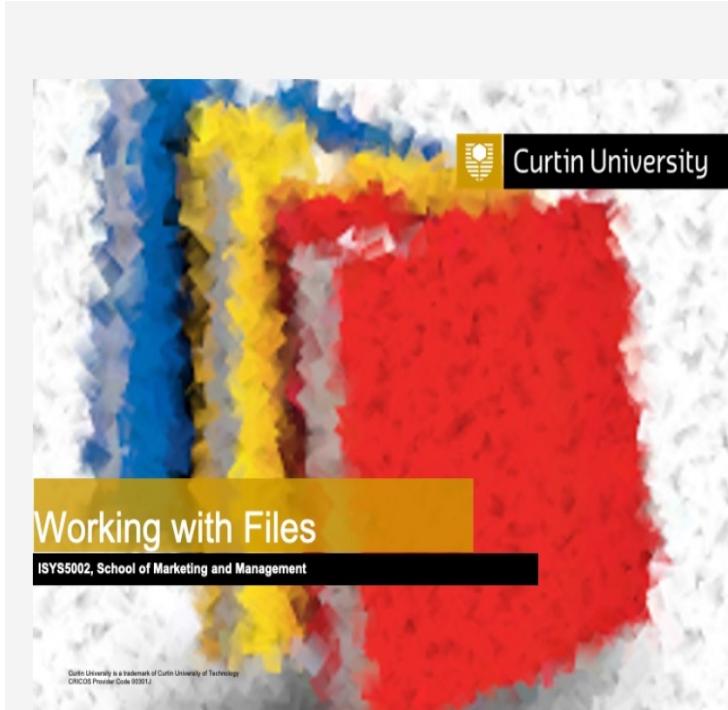
This week we introduce our problems solving framework. We look and different data types and introduce the six operations a computer can do. In the workshop, we explore the programming environment and python.

Building Blocks



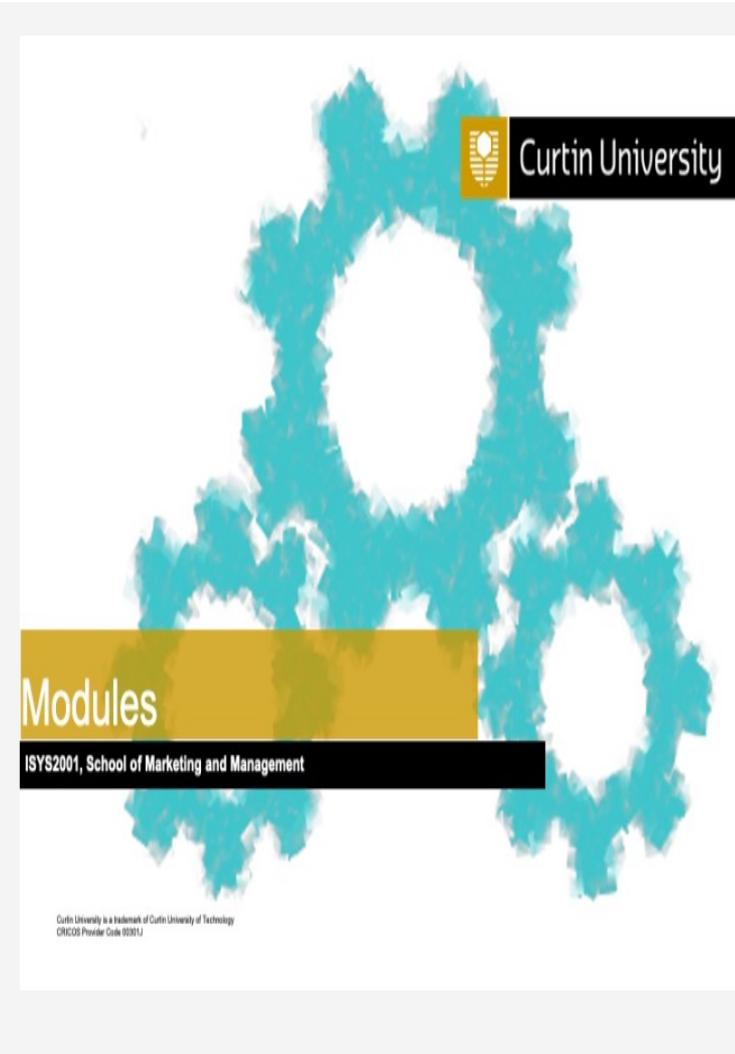
There are three main ways to break programs into smaller parts: functions, Objects and Modules. This week we look at Functions. Functions are like building blocks of code you can use over and over again.

File IO



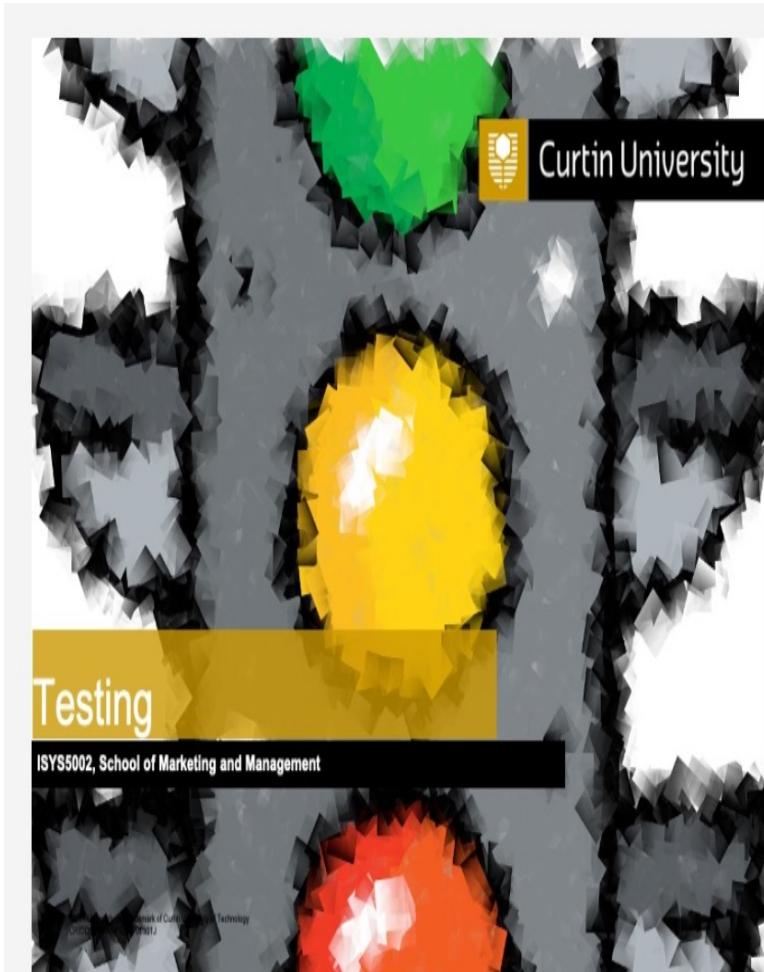
Reading and writing files is an essential programming skill. Python gives you a simple and powerful tool for working with files. In this module, we will show you how to read and write files. Input validation code verifies that user-supplied data, such as text from the input() function, is formatted appropriately. We look at three methods, try/except, isdigit() and using a module.

Modules



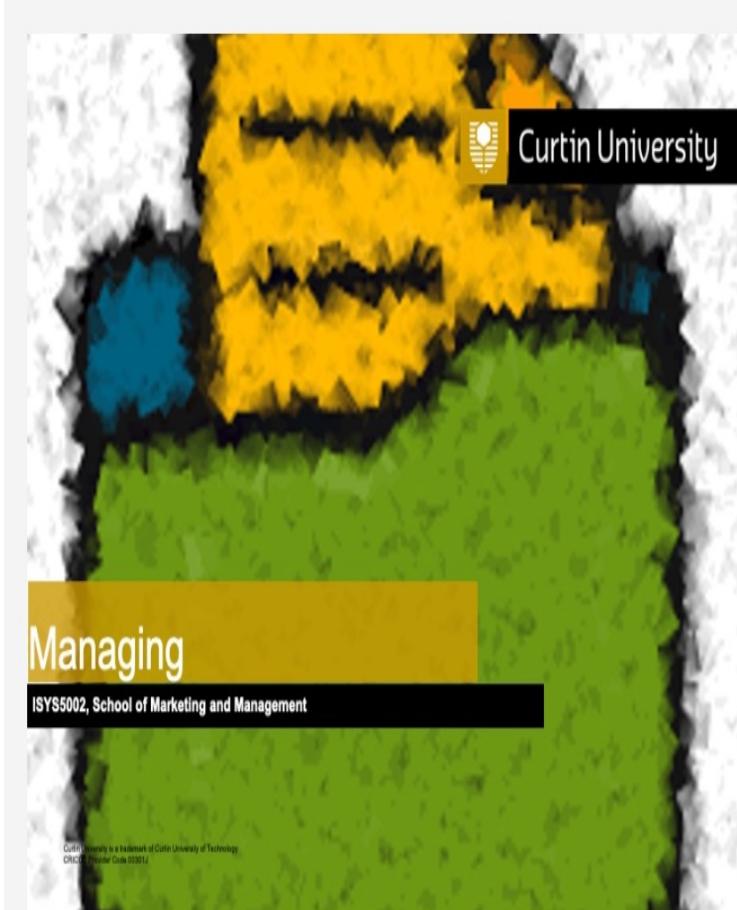
This week we explore modular design. We create a simple module and discuss packages. We extend our understanding of files and consider the processing of CSV files.

Testing



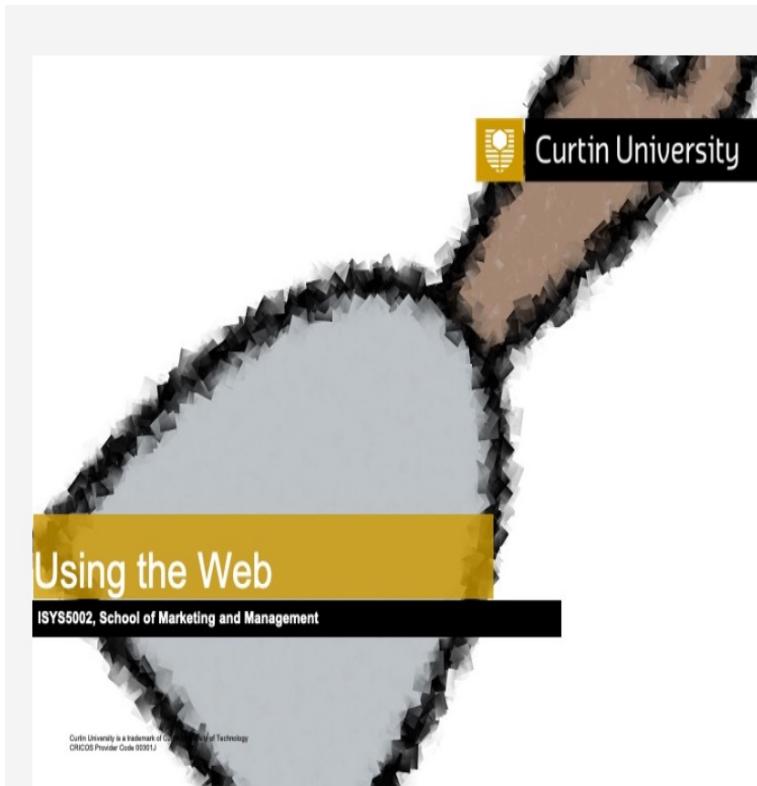
This week we look at testing. This extends work previous weeks introducing terminology and yet again more libraries. We look at testing from a users perspective and a developer.

Best Practices



This week we look at how to manage projects. This extends work previous weeks introducing terminology. We look at best practices from a single developer perspective.

Using the Web



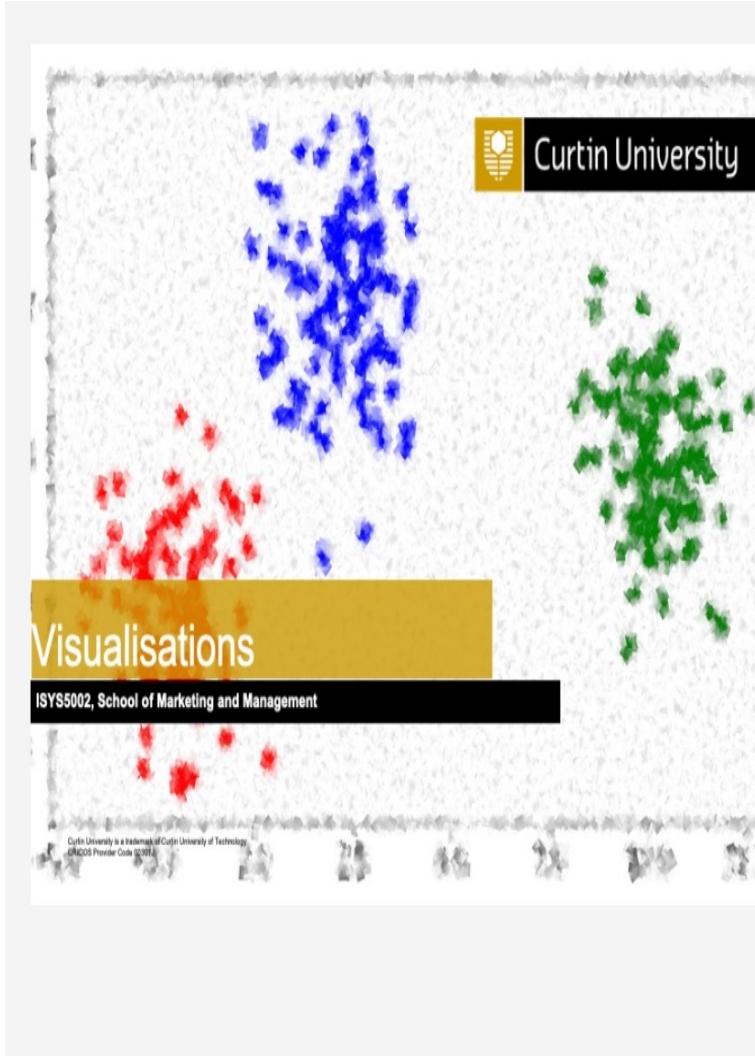
Web scraping is an automated method used to extract large amounts of data from websites. The data on the websites are unstructured. Web scraping helps collect these unstructured data and store it in a structured form. There are different ways to scrape websites such as online Services, APIs or writing your own code. This week, we'll see how to implement web scraping with python.

Data Management



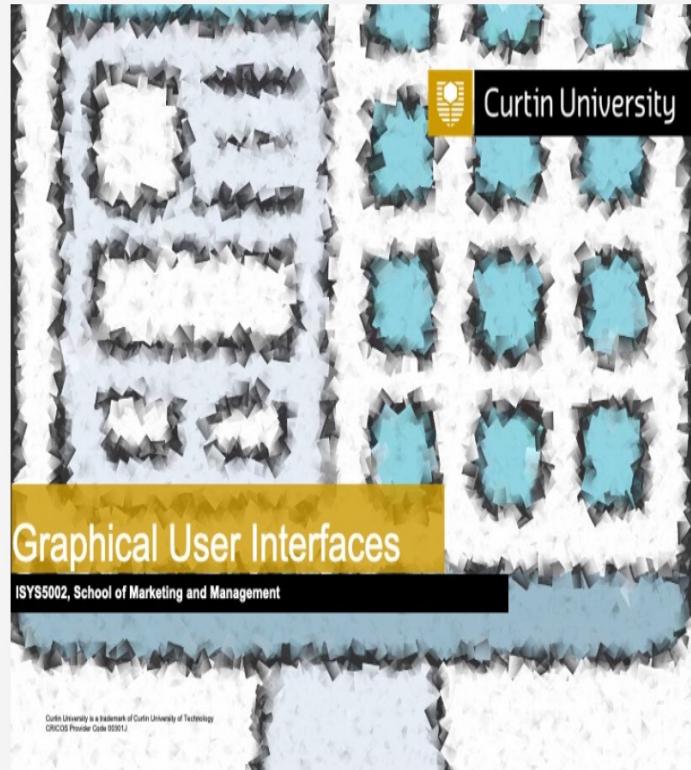
All programs process data, and many need to be able to save and retrieve that data. In this module, we look at flat files, SQLite and Pandas, to give your programs database functionality.

Visualisations



Numerous interactive outputs are supported by Jupyter Notebook, including the ipywidgets ecosystem and many interactive visualisation packages. With the proper settings, these are also supported in Colab Notebooks. This week we look at the various plotting and visualisation options

Graphical User Interfaces

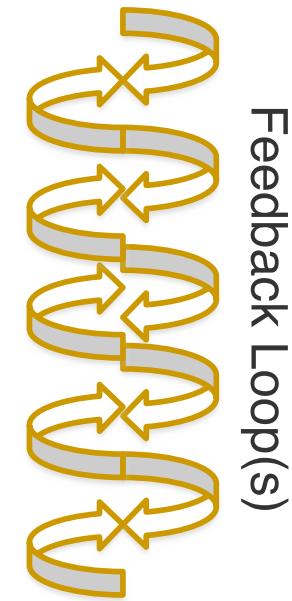


The GUI (Graphical User Interface) is the part of the Python language that allows us to create windows, buttons, menus, and other visual items on the screen. Users can interact with these elements to carry out different tasks.

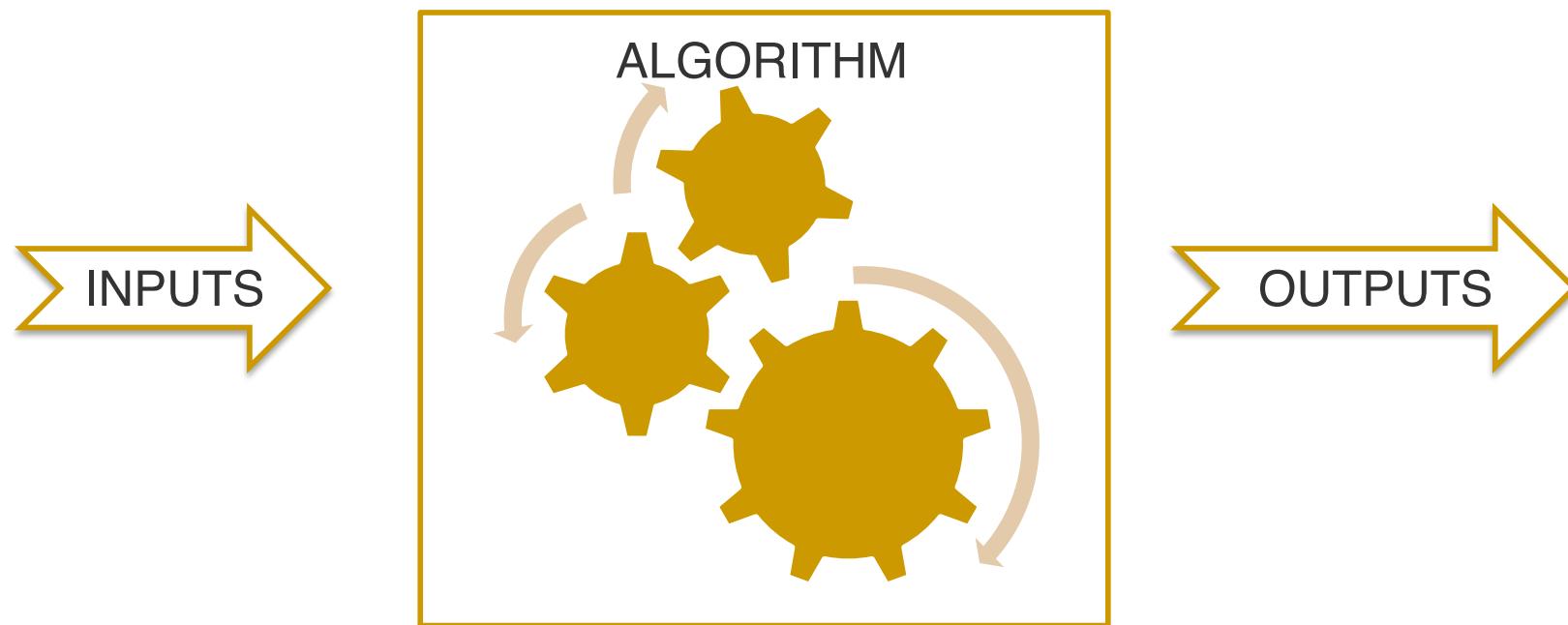
This week will look at the main aspects of GUIs and how you can use this feature in your Python Notebooks.

Problem Solving Methodology

- State the problem clearly
- Describe the input and output
- Work a simple example by hand
- Develop an algorithm (and convert to Python)
- Test solution with a variety of data



Program



Six Things

- Input
- Output
- Calculate (Add, multiply, less than etc.)
- Store (assignment)
- Decide (if-then)
- Repeat (for, while)

Core Concepts

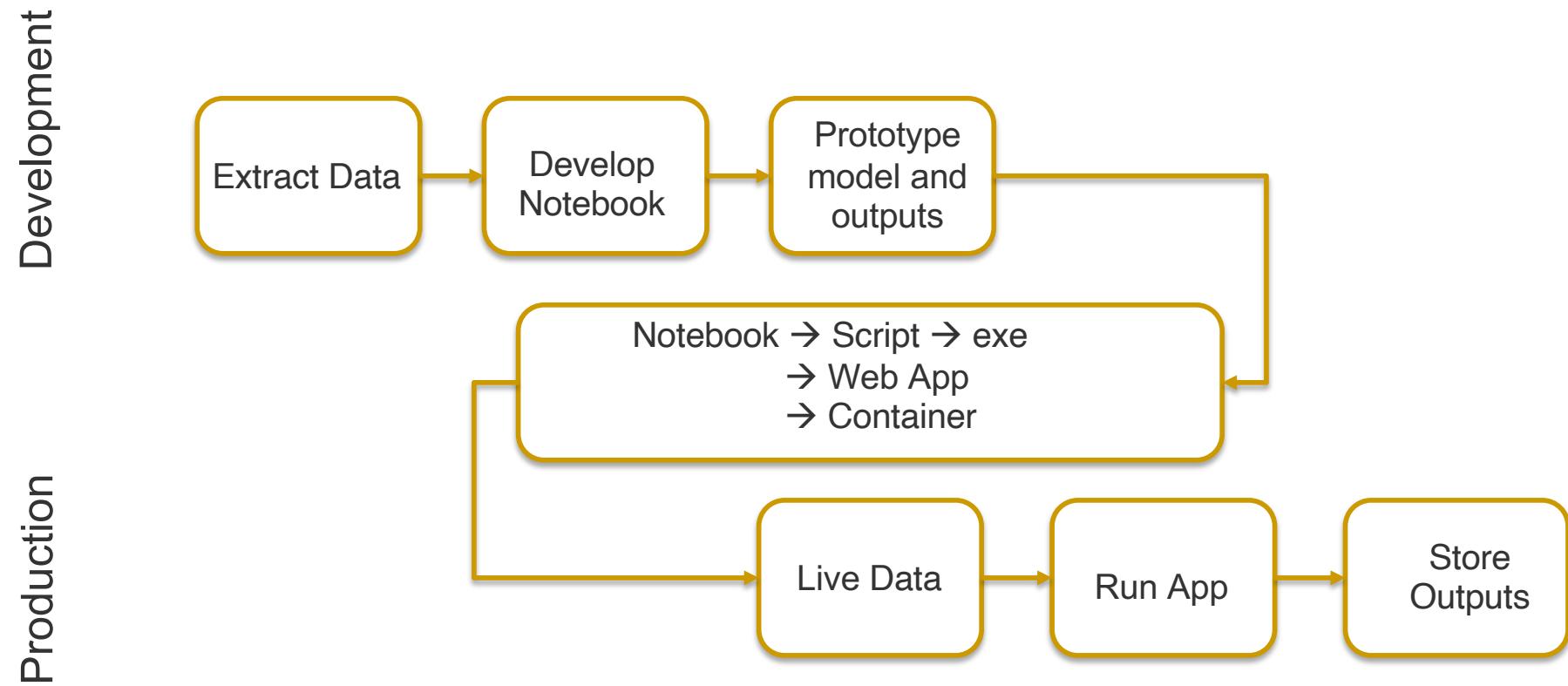
- Algorithms
 - Expressed in structured English (Pseudocode)
 - Sequence (must put things in correct order)
 - Selection (if-then-else)
 - Repetition (while)
- Encapsulation (group things)
 - Manage complexity
 - Data example – lists group values/variables
 - Code example – function group expressions

*Coding is a small part of
Programming*

(Our) Development Environment

- Python Notebook
 - Code Cells
 - Text Cells (Markdown)
 - Interactively run cells
 - Output in notebook
- Workflow
 - Edit in notebook
 - Frequently save to GitHub
 - Deploy Binder/NBViewer/Report

Typical Workflow



What is Web Scraping

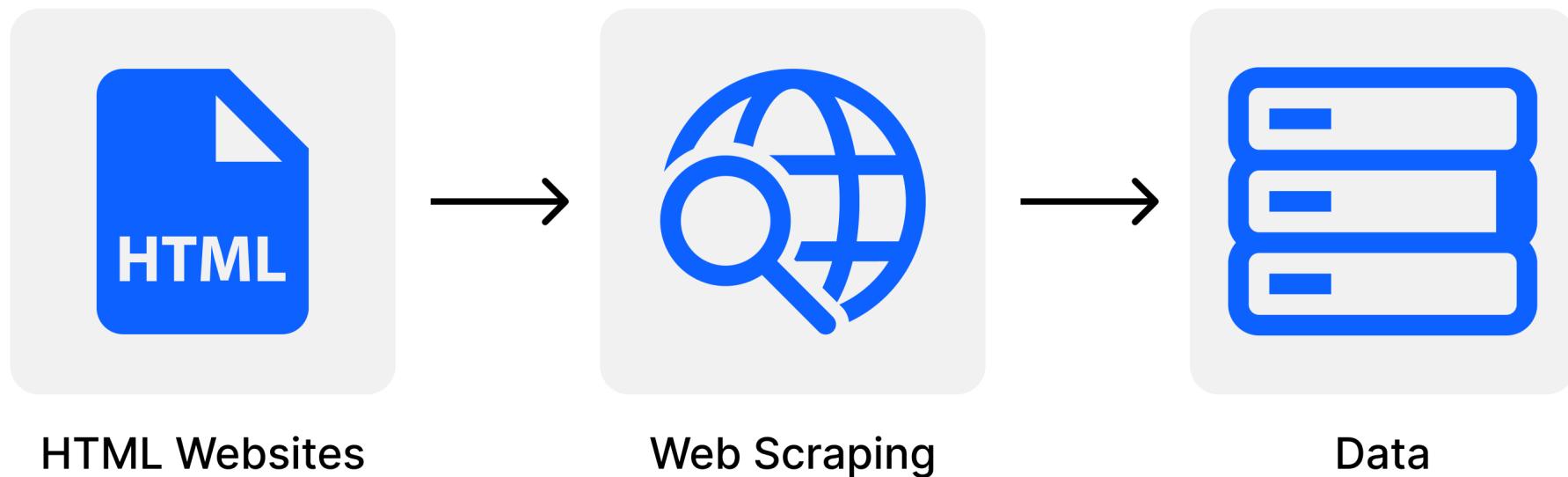


Image Source: <https://i.imgur.com/6zM7JBq.png>

Python Graph Gallery

■ Plot + Code



```
# libraries
import matplotlib.pyplot as plt
import numpy as np

# create data
values=np.cumsum(np.random.randn(1000,1))

# use the plot function
plt.plot(values)
```

<https://python-graph-gallery.com/line-chart/>

ISYSS2001 – Revision

Curtin University is a trademark of Curtin University of Technology
CRICOS Provider Code 00301J



Curtin University

App in a notebook

Jupyter example_app Last Checkpoint 2 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 2

In [1]:

```
from __future__ import division
import ipywidgets as ipw

output = ipw.Text(value="0", layout=ipw.Layout(width="212px"), disabled=True)

def on_click(btn):
    if btn.description == "=":
        try:
            output.value = str(eval(output.value))
        except:
            output.value = "ERROR"
    elif btn.description == "AC":
        output.value = ""
    elif btn.description == "del":
        output.value = output.value[:-1]
    else:
        output.value = output.value + btn.description

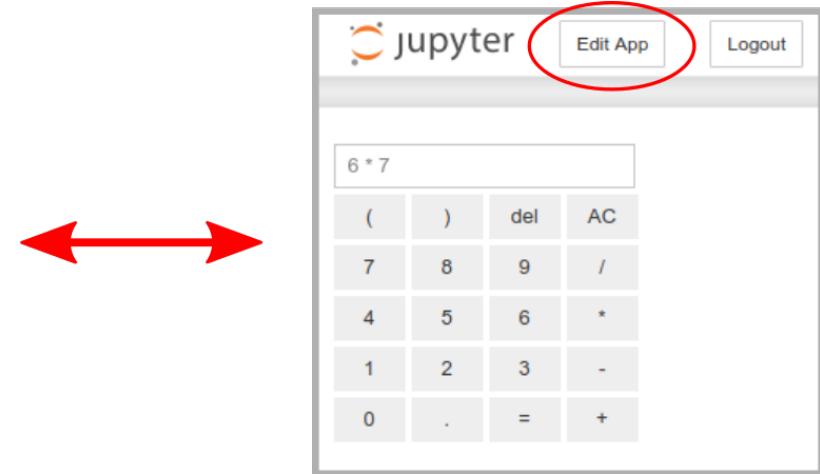
def mk_btn(description):
    btn = ipw.Button(description=description, layout=ipw.Layout(width="50px"))
    btn.on_click(on_click)
    return btn

row0 = ipw.HBox([mk_btn(d) for d in ("(", ")")])
row1 = ipw.HBox([mk_btn(d) for d in ("7", "8", "9", "/")])
row2 = ipw.HBox([mk_btn(d) for d in ("4", "5", "6", "*")])
row3 = ipw.HBox([mk_btn(d) for d in ("1", "2", "3", "-")])
row4 = ipw.HBox([mk_btn(d) for d in ("0", ".", "=", "+")])
ipw.VBox([output, row0, row1, row2, row3, row4])
```

0

()	del	AC
7	8	9	/
4	5	6	*
1	2	3	-
0	.	=	+

In []:



The Real reason everyone is here....

Pass requirements

To pass this unit, students must:

- receive an overall mark of 50% or greater; and
- have submitted ALL Assessments.

Extended Learning Portfolio (ELP)

- Designed to take 3-5 Hours
- 24 Hour Window
 - Starts at published time ENDS next day
 - Check Official timetable!**
- Download → complete → Submit
 - Download when want, must submit by end time
 - Download DOES NOT change time due
- 5 ‘main’ questions
- Need to create 1-2 GitHub repositories
- Need to write 1 or more notebooks
- Copy: Body of Journals → Word document → Turnitin

*Apply what you know in a new
and novel way*

Question Ideas

- Reflective Writing (Journals as a doc)
 - Most important ULO, Why
 - Best activity/notebook, Why
 - Supported using example of your work
- Design a program to...(Step 1-4 of methodology)
- Implementation the following... (I/O, testing, interface, github)
- Fix/test/debug notebook/cell (logical, syntax, debugging, exceptions)
- Explain/Complete some code or algorithm
 - Add comments (inline/function)
 - Improve variable/function names

ELP Tips

- Focus - Application of Knowledge
- Logic important Syntax less so
 - Explain/Demonstrate in psuedocode
 - Correct Syntax bonus marks
- Answer the Question
 - Show me what you know
 - No Answer - round figure
 - Worst Case - rewrite the question

*If you can't do something, then
the next best thing is to
document what you want to do.
Not as good as doing it, but
better than not mentioning it.*

Revision Strategy

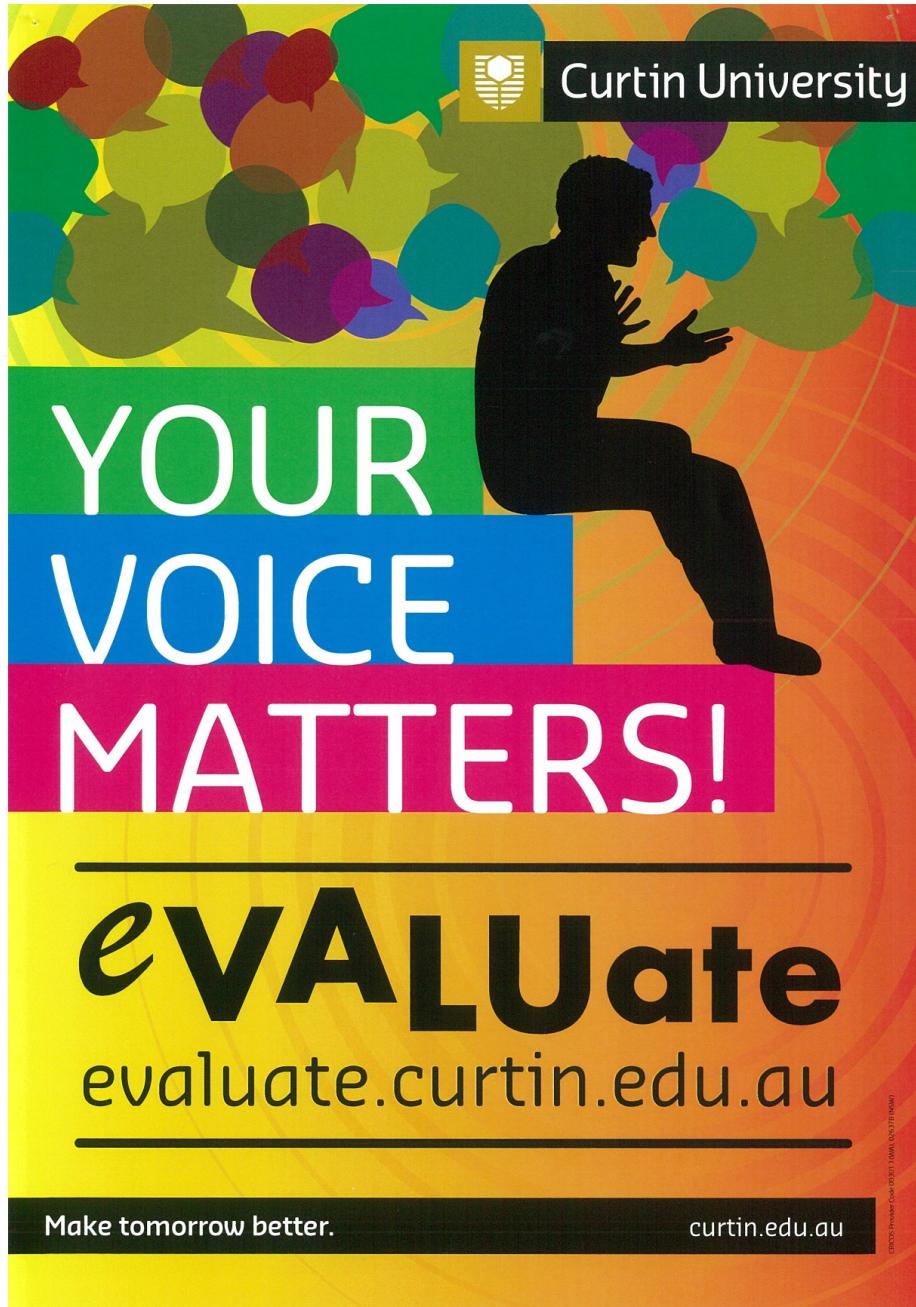
- Blackboard (Lecture notes, Videos, notebooks)
- Review all notebooks
 - Delete answer and try again
- Online Judge
- YouTube
 - Socratica Python Playlist
 - See Blackboard

Important Notebooks

- Design - Calculator Walkthrough
- Implementation - (Good) Calculator Walkthrough
- Testing (assert/doctest)
- Pay Calculator (Forms and ipywidgits)
- Input validation
- Simple Dashboard

Can You...

- Provide an overview of ULO
- Provide an overview of Lectures
- Discuss Extended Learning Portfolio
 - Questions
 - Study Strategies



ISYSS2001 – Revision

Curtin University is a trademark of Curtin University of Technology
CRICOS Provider Code 00301J

01 / 2015



Curtin University

Why eVALUate?



Curtin University

eVALUate is Curtin's way of getting feedback on your experience in a unit and with your teachers.

This feedback helps us to improve your learning experience in individual units and the overall course.

We tend to only give feedback when something is really good or really bad. But we also want all the important feedback that sits in the middle. This helps us to improve and be better at what we do.

Be **professional** when completing the survey – remember that integrity and respect are important Curtin values. You can be *constructive* and *kind* at the same time.

#bekind

eVALUate now!

Giving constructive feedback...

You can comment on:

- Assessment and feedback
- Lecture and tutorial content
- Organisation and structure of the unit
- Teaching methods
- Use of technology
- Communication between you and your teacher

**BE
KIND**

Some tips on giving feedback:

- Be specific with your comments and provide relevant examples
- Avoid comments that are personal (dress, appearance, manner, etc.) - focus on the teaching and learning activities
- Try to describe the effect of the teacher's behaviour on your learning and offer alternatives so that we can improve
- Be respectful - do not write anything that is offensive, racist or sexist
- Avoid emotional language
- Comment on both the good and bad parts of the unit and/or teaching

Some examples...

BE
KIND

Instead of this...	Try this...
My tutor was boring and seemed disinterested.	My tutor could try and make the class more interesting by providing real-life examples and engaging the class more in discussion.
The feedback I got on my assignment didn't help me at all.	Specific feedback on how I could improve my writing would be really useful and help me in my further studies.
I can't understand a thing the lecturer is saying – it's like he speaks another language.	It would be helpful if the lecturer spoke slower and emphasised the main points.
My tutor was great!	My tutor was great because she explained everything clearly and was open to questions. She was also enthusiastic and clearly loves her job.
I enjoyed this unit.	I enjoyed this unit because the lecturer was organised and was passionate about the topic. I really liked the way he brought in guest lecturers from industry to give us some real life examples. The assignments were also really useful for my career development.
My tutor never responded to my emails – does he even work here?	Perhaps the tutor could be more organised in the way he responds to emails and student queries. A timely response would really help my understanding of the unit content.

***"You are braver than you believe,
stronger than you seem, and
smarter than you think."***

— Christopher Robin, Winnie the Pooh.

Thank You!

