



Curtin University

# Managing Code

ISYS2001, School of Marketing and Management

## **ELECTRONIC WARNING NOTICE FOR COPYRIGHT STATUTORY LICENCES**

### **WARNING**

This material has been reproduced and communicated to you by or on behalf of **Curtin University** in accordance with section 113P of the *Copyright Act 1968 (the Act)*

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.



I acknowledge the traditional custodians of the land on which I work and live, and recognise their continuing connection to land, water and community. I pay respect to elders past, present and emerging.



# Today

- Enterprise Software
- Features of Software Ecosystem
- Development → Production Workflow
- Deployment
- Future

# Enterprise Software Today

- Cloud Based
- Complex
- Independent of Operating System

Browser Based

# Features of Software Ecosystem

- Reliability
- Accessibility
- Reusability
- Interpretability
- Flexibility
- Agility

# Ecosystem Considerations

- How do you get the data?
- Where does the code run?
- How is your code organised?
- Where do your outputs go?



# Application

- Development
- Production
- Interface - Widgets + Code
- Deployment

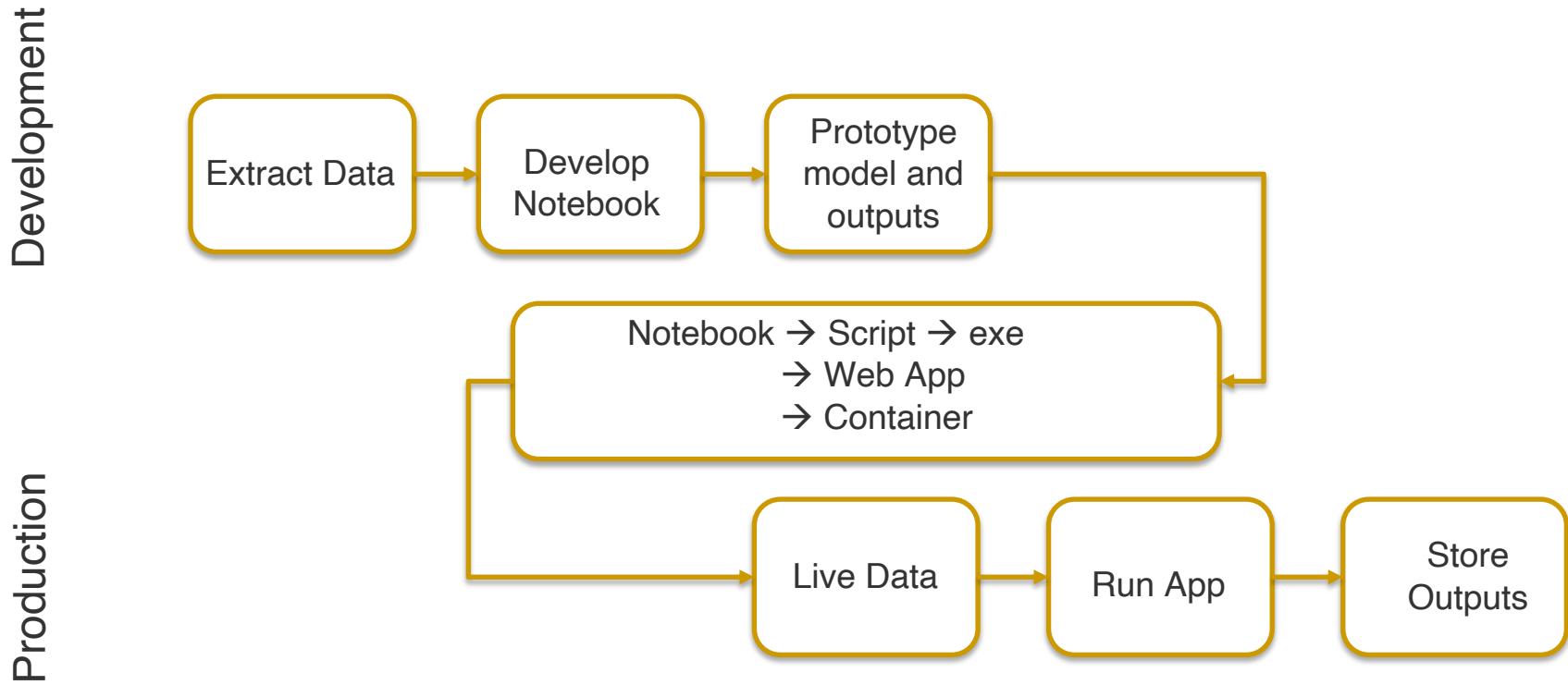


# Virtual environments

- Create virtual environment
- Activate virtual environment
- Install required packages
- Deactivate environment



# Typical Workflow



# Development

- Local Environment
  - Jupyter Lab, VSCode,
- Cloud Environment
  - AWS, Google Compute, Asure
- Notebook Specific
  - Paperspace, Colab

# Deployment

- Notebooks as Packages
  - NBDev
- Notebooks as (Web) Apps
  - Ipywidgets, Voila, Appmode, Anvill
- Deploy/Share
  - nbviewer, GitHub, Binder, Docker
  - nbconvert → Script → Package/Exe



# Binder

## Build and launch a repository

GitHub repository name or URL

1 GitHub ▾ 2 https://github.com/michael-borck/the\_calculator\_walkthrough.git

Git ref (branch, tag, or commit)

HEAD

Path to a notebook file (optional)

Path to a notebook file (optional)

File ▾

4 launch

Copy the URL below and share your Binder with others:

https://mybinder.org/v2/gh/michael-borck/the\_calculator\_walkthrough.git/HEAD



Expand to see the text below, paste it into your README to show a binder badge:  launch 



# Appmode

Jupyter example\_app Last Checkpoint: 2 minutes ago (unsaved changes) Logout Trusted Python 2

File Edit View Insert Cell Kernel Widgets Help Code Appmode

In [1]:

```
from future import division
import ipywidgets as ipw

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

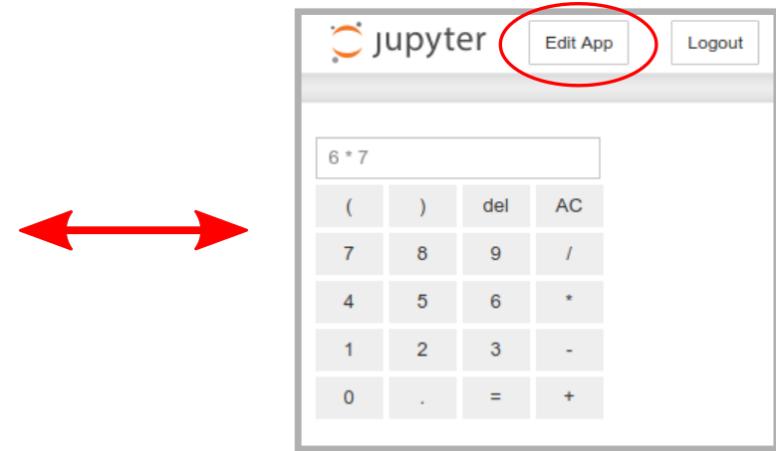
def on_click(btn):
    if btn.description == "=":
        try:
            output.value = str(eval(output.value))
        except:
            output.value = "ERROR"
    elif btn.description == "AC":
        output.value = ""
    elif btn.description == "del":
        output.value = output.value[:-1]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



# Future

- Notebooks as Applications
- Rise of Containers
- New Platforms
- New Capabilities



# Can you?

- State the direction of enterprise software today
- List the features of software ecosystem
- Describe a development to production workflow
- Suggest how to deploy and share code
- Outline a possible future for notebooks