Google colab Getting Started…

By Dr Dawn Carmichael

Colaboratory Notebooks, are computational notebooks, which allow the user to create documents which include code. A useful analogy to help understand what a notebook is, is to consider a word document with the typical objects such as images and tables. Imagine that, the word document could also have sections where computer code could be executed. These dynamic sections could contain code that ran based on data which could change over time giving up to the minute content e.g. bar charts showing sales. These sections could also contain code which would allow the user to interact with the code e.g. inputting sales time period. Now imagine that instead of a word document you were creating a web-based document which is accessed via a browser. You have now imagined a computational notebook.

Another tool for creating computational notebooks is Jupyter, which is very similar but we are not using mainly because it requires installation on the machine that you are using. Colab notebooks are cloud based and can be accessed anywhere via a browser.

Computational notebooks, such as colab, are used extensively in data science because you can use python programming language in the code sections.

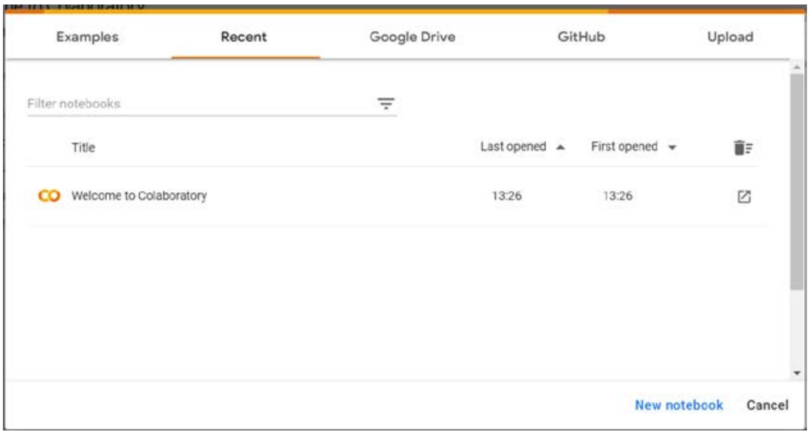
First things

When you use colab to create a notebook it will save the file to your google drive. If you don't have google drive, go to https://www.google.co.uk/intl/en-GB/drive/ .

> **What is an ipynb File?**
> Each .ipynb file is one notebook, so each time you create a new notebook, a new .ipynb file will be created. An .ipynb file is a file that describes the contents of your notebook in a format called JSON. Each cell and its contents, including image attachments that have been converted into strings of text, is listed there along with some metadata.

Important - computational notebooks have the file extension .ipynb

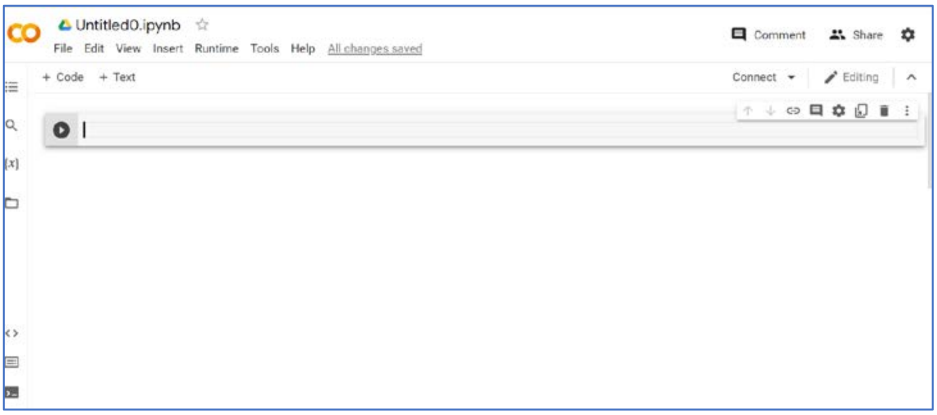
One way to get started with Colab is to navigate to the starting page - https://colab.research.google.com/. There you will see the introduction screen, select 'new notebook' towards the bottom right of the screen.



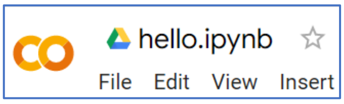
Title	Last opened ▲	First opened ▼
Welcome to Colaboratory	13:26	13:26

Alternatively, you can go to google drive, click on the new button, then navigate via 'more' to 'Google colaboratory'.

When you have navigated to the new notebook you will then see the Colab notebook interface as shown below.



Click on 'untitled0.ipynb' and change the file name to hello.



Cells & Kernels

There are two fairly prominent terms that you should note which are probably new to you in this context; cells and kernels. Both are key to understanding computational notebooks and to what makes it more than just a word processor. Fortunately, these concepts are not difficult to understand.

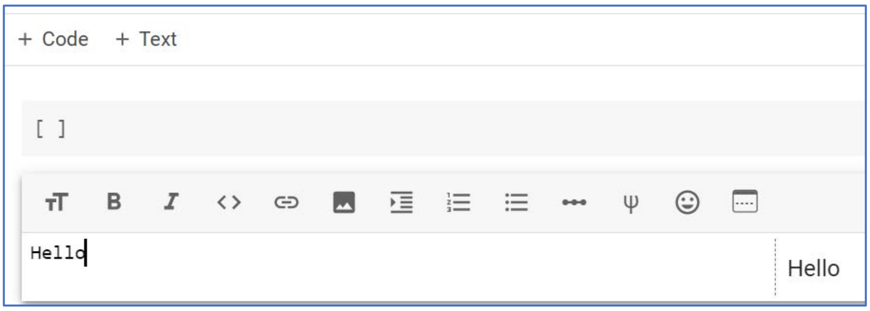
A **kernel** is a "computational engine" that executes the code contained in a notebook document. Typically, the python language is used for code. A **cell** is a container for text to be displayed in the notebook or code to be executed by the notebook's kernel.

Cells form the body of a notebook. There are two main types that we will cover:

> A text **markdown** cell contains text formatted using 'Markdown' and displays its output in-place.
>
> A **code** cell contains code to be executed in the kernel. When the code is run, the notebook displays the output below the code cell that generated it.

Let's test out the difference with a classic hello world example.

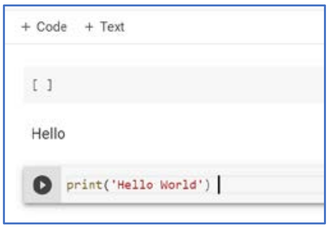
Click on '+ text' and then type in 'hello'.



```
Hello
```

Hello

Next click on '+ code' and type in print('Hello World')

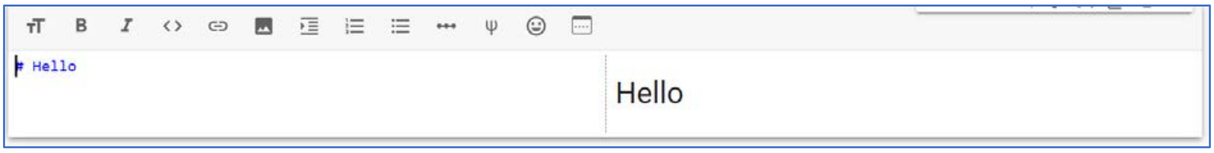


Hello

```
print('Hello World')
```

Now you have to **run the notebook to execute the code**, go to the 'runtime' menu along the top and select 'run all'. It might take a few minutes but eventually the interface will show green ticks to the left of your work which means that you have created a computational notebook. Alternatively, you can click on the arrow to the left of your input which will execute only that cell.

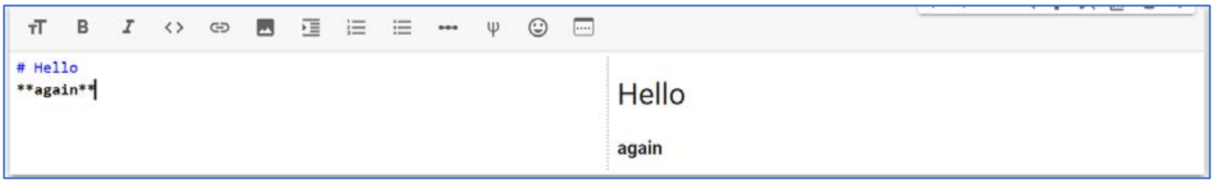
Now let's add some markdown to the text. Click at the beginning of the 'hello' in the text cell and type in one # followed by a space. The result of the markdown will be shown on the right as follows.



```
# Hello
```

Hello

There are also tools on the toolbar such as Bold to help you format the text. However, be aware that not all interfaces for creating notebooks have this toolbar. Try out the tools, type in the word ‘again’ on the next line and click on the bold icon. Colab adds the markdown for you as follows.



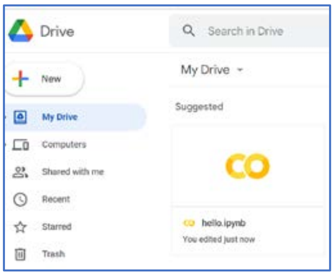
```
# Hello
**again**
```

Hello

again

Saving your work

The notebook is being automatically saved so the next important step is being able to retrieve your work. Close down the browser window containing your notebook and then go to google drive - https://drive.google.com/drive/my-drive . You should see your most recent file which in this case is ‘hello.ipynb’.



If you double click on the file name it will take you back to the colab interface and show your file.

Going Local

In order to get a local copy of the ‘hello.ipynb’ go to the file menu and choose ‘download’ and then ‘download ipynb’.

This is what you should do to keep track of your work and to have the .ipynb file to submit for the assessment.

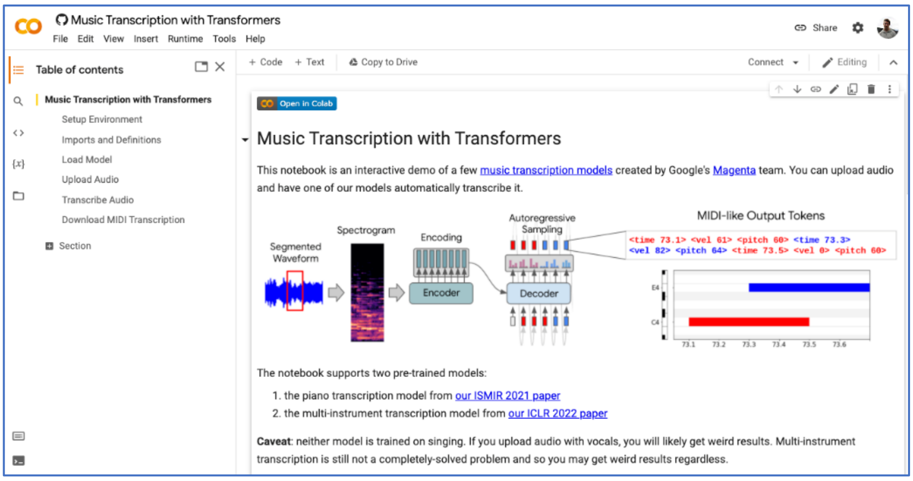
You can upload local .ipynb files via the file menu ‘Open Notebook’

Why isn’t it working?

Every now and again you might get an error in the syntax which you should be able to resolve by checking the example you were given or asking for help in the labs. It is also possible that the Kernel is stuck on an error you made previously and won’t work properly again until you do a restart of the runtime. To do this go to the runtime menu and select ‘restart and run all’. If the problem persists, try creating a new workbook and copying over you code, if it still doesn’t work it is probably a syntax error.

Where are we going with this?

A notebook can have titles, subtitles, text, images, charts and graphs. A bit like the example below.



Music Transcription with Transformers

This notebook is an interactive demo of a few music transcription models created by Google's Magenta team. You can upload audio and have one of our models automatically transcribe it.

```
<time 73.1> <vel 61> <pitch 60> <time 73.3>
<vel 82> <pitch 64> <time 73.5> <vel 0> <pitch 60>
```

The notebook supports two pre-trained models:

1. the piano transcription model from our ISMIR 2021 paper
2. the multi-instrument transcription model from our ICLR 2022 paper

Caveat: neither model is trained on singing. If you upload audio with vocals, you will likely get weird results. Multi-instrument transcription is still not a completely-solved problem and so you may get weird results regardless.

1 Basics; mark-down and code

By Dr Dawn Carmichael

Take a moment to examine the interface.

Check out the menus to get a feel for it, take a few moments to scroll down the list of commands in the command palette, which is the small button with the keyboard icon on the left-hand toolbar (or Ctrl + Shift + P).

Markdown Classics

For the first example create a text cell and then use the single # to denote the highest-level heading. Lower level headings are marked with increasing numbers of #s.

Headings

Go into your hello world example and type out

```
# Top Heading
## Second Heading
### Third Heading
```

Then press ctrl enter or click on the run button on the left and note the results.

Bold

You can mark text as bold as follows;

```
**bold my text** my normal text
```

You can also use HTML for example with bold as follows;

```
<b>bold my text</b> my normal text
```

Newline

Use the following markdown to add a newline

```
One\
Two
```

Links

You can add links as follows;

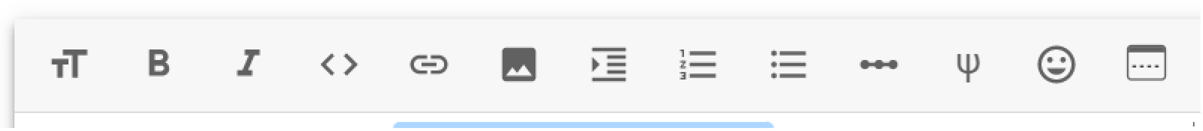
```
[search here](https://www.google.com)
```

Or the HTML form

```
<a href="https://www.google.com">search here</a>
```

Images

To display an image in a text cell, choose 'Insert image' button from Edit menu immediately above the input point. It is just after the link button which you can use instead of markdown.



Challenge

Create heading and sub-heading sections for a notebook suitable for 'Cloud Computing with Colab', making use of the markdown above. Find a suitable image for your logo and add a link.

Python Starters

Remember you can add cells, by using the '+ code' and '+ text' buttons and you also remove cells via the edit menu.

Variables

Python has the same features as other languages check this out by taking a moment to do a simple addition using variables. Add the following code to a code cell and then run it.

```
x=5
y=10
z=x+y

print(z)
```

You should take a few minutes to familiarise yourself with this code by altering the values and the arithmetic operator.

You can also set string variables. I am not going to give you the code in this worksheet, instead go to Google search and type in 'colab string variables'. You will probably get a number of results, usually the best ones start with colab.research.google.com which will probably take you to a live example.

Lists

A list is a data type which is basically a group of data similar to arrays. Before you run the example list code, decide what you think it is going to show.

```
thislist = ["apple", "banana", "cherry"]

print(thislist)

print(thislist[1])
```

Were you correct?

Consider this example list code, what will it display?

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[2:5])
```

Items in lists can be changed.

```
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"
print(thislist)
```

They can also be inserted into.

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "orange")
print(thislist)
```

They can also be appended.

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

You can delete a named value.

```
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

Or you can delete by index number.

```
thislist = ["apple", "banana", "cherry"]
del thislist[0]
print(thislist)
```

You can also step/progress through a list using a for loop

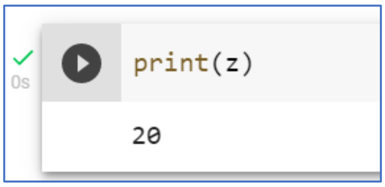
```
listnum = [1, 2, 3]
print(listnum)

for x in listnum:
    print(x)
```

Challenge

Change the name of the list and the list members for one of the examples above.

Cool Stuff about Notebooks

You can create a variable or list with values in one cell and use it in another cell further down the notebook.
Assuming that you are continuing to work in the same notebook and have already assigned values to x, y and z, create a new code cell and type in print(z) and run it. You should see the result from the calculation that you created further up the notebook.



```
print(z)
```

```
20
```

Challenge - notebook

The most important take-away from these exercises is not to attempt to memorise the markdown and code but to start thinking about how it could be useful.

Working in small groups discuss how a colab notebook could be used in an organisation such as a streaming service. In essence you are thinking about where a report containing executable code might be useful.