

Applicant Visit Day

Getting to grips with code!

Prof Yannis Korkontzelos

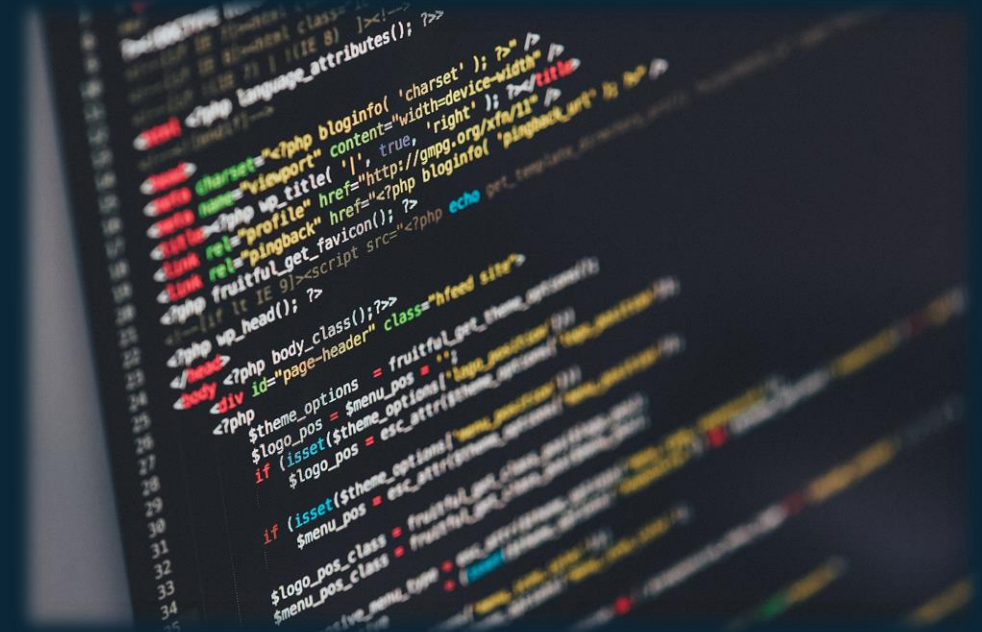
Material preparation: Dr. Robert Lyon

Welcome

- Welcome to the virtual applicant visit day!
- This session aims to **introduce you to computer programming**, giving you an idea of the type of work you'll do as a first year undergraduate studying here.
- The session aims to:
 - Help you understand the **importance of coding**.
 - Help you understand how coding **relates to our degree programmes**.
 - Introduce some basic **coding terminology**.
 - Give you the opportunity to **solve some coding challenges** for yourself.
- We have only 1 hour, but I encourage you to take up **independent study** after the session, whether you join us here or not. It pays to get a head start!

Programming in a Nutshell

- Programming involves the writing of **instructions** as code, that orchestrate the actions of a computer and it's **hardware components**.
- The instructions are written with the aim of accomplishing a specific goal or completing a specific task.
- Ultimately, the code we write controls a computers' **CPU**, to carry out the tasks.
- There are a few different ways we can “**talk**” to a computer...



Talking to a Computer

Machine code:

```
01001000 01100101 01101100
01101100 01101111 00100000
01000011 01001001 01010011
00110001 00110001 00110001
00110010
```

Assembly language:

```
MOV AL, 1h;
MOV CL, 2h;
MOV DL, 3h;
```

Natural Language:

Hi computer, please could you
tell students on the visit day
that coding is actually fun but
also a powerful skill to learn?



```
01100001 01101100 01101001
01100011 01100101 00100000
01110011 01110101 01100011
01101011 01110011 00001010
```

Python – a high-level language

- Python is a **interpreted, high-level** programming language.
- The Python code we write it is interpreted line by line and executed by a computer.
- First we write human readable Python source code files. These end with the `.py` extension.
- The a program know as the Python interpreter takes these files and converts them in to machine readable **byte code** files ending with the `.pyc` extension.
- This output is read by the CPU so it can carry out tasks – this is what we'll do today.



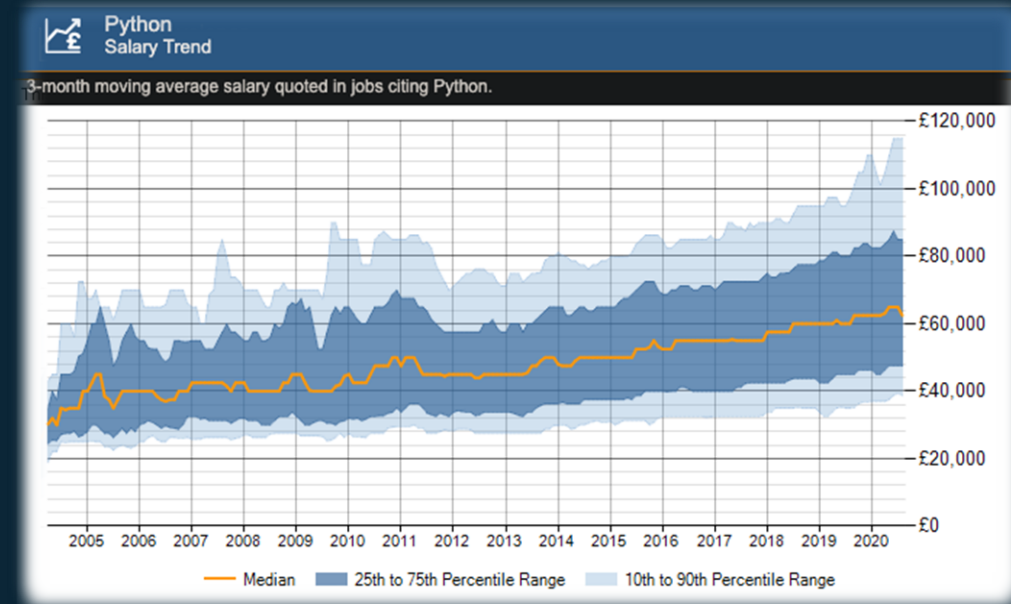
Why is programming important?

- We live in an increasingly **digitised** world - from the mobile phones in our pockets, the computers we use and even home appliances (.e.g. smart fridges!).
- Such technologies are **only made possible** through software – software that,
 - Controls these systems.
 - Allows them to interact with each other.
 - Allows us to interact with them.
- Software is comprised of **computer code** written by programmers. Thus, without code and programmers, all these amazing systems would simply not exist.
- The importance of software is also **increasing over time**.
- We therefore have a choice – do we want to just be **users** of the software (with no control), or, do we want to help **shape** the software all around us....
- Given that our world is becoming **increasingly automated** due to software, I would suggest that it's advantageous to be skilled in software creation – it will make us more **employable**.

A good career path?

Rank	Name	Country	Net worth (USD)	Age	Main source of wealth	Ref(s)
1	Jeff Bezos	USA	\$113.0 billion ▲	55	Amazon	[4]
2	Bill Gates	USA	\$107.1 billion ▲	64	Microsoft	[5]
3	Bernard Arnault	FRA	\$106.6 billion ▲	70	LVMH	[6]
4	Warren Buffett	USA	\$86.9 billion ▲	89	Berkshire Hathaway	[7]
5	Mark Zuckerberg	USA	\$74.9 billion ▲	35	Facebook	[8]
6	Larry Ellison	USA	\$69.2 billion ▲	75	Oracle	[9]
7	Amancio Ortega	ESP	\$69.1 billion ▼	83	Zara	[10]
8	Larry Page	USA	\$61.2 billion ▲	46	Google	[11]
9	Carlos Slim	MEX	\$60.4 billion ▼	79	America Movil	[12]
10	Mukesh Ambani	IND	\$60 billion ▼	62	Reliance Limited	[13]
11	Sergey Brin	USA	\$57.5 billion ▼	46	Google	[14]
12	Françoise Bettencourt Meyers	FRA	\$55.7 billion ▼	66	L'Oréal	[3]
13	Steve Ballmer	USA	\$55.6 billion ▲	63	Microsoft	[15]
14	Michael Bloomberg	USA	\$54.6 billion ▲	76	Bloomberg	[16]
15	Jim Walton	USA	\$53.1 billion ▼	71	Walmart	[17]
16	Alice Walton	USA	\$52.8 billion ▼	70	Walmart	[18]
17	S. Robson Walton	USA	\$52.8 billion ▼	75	Walmart	[19]
18	Charles Koch	USA	\$42.9 billion ▼	84	Koch Industries	[20]
19	Julia Koch	USA	\$42.9 billion ▼	57	Koch Industries	[21]
20	Jack Ma	CHN	\$41.4 billion ▲	55	Alibaba Group	[22]

As of 28 November 2019

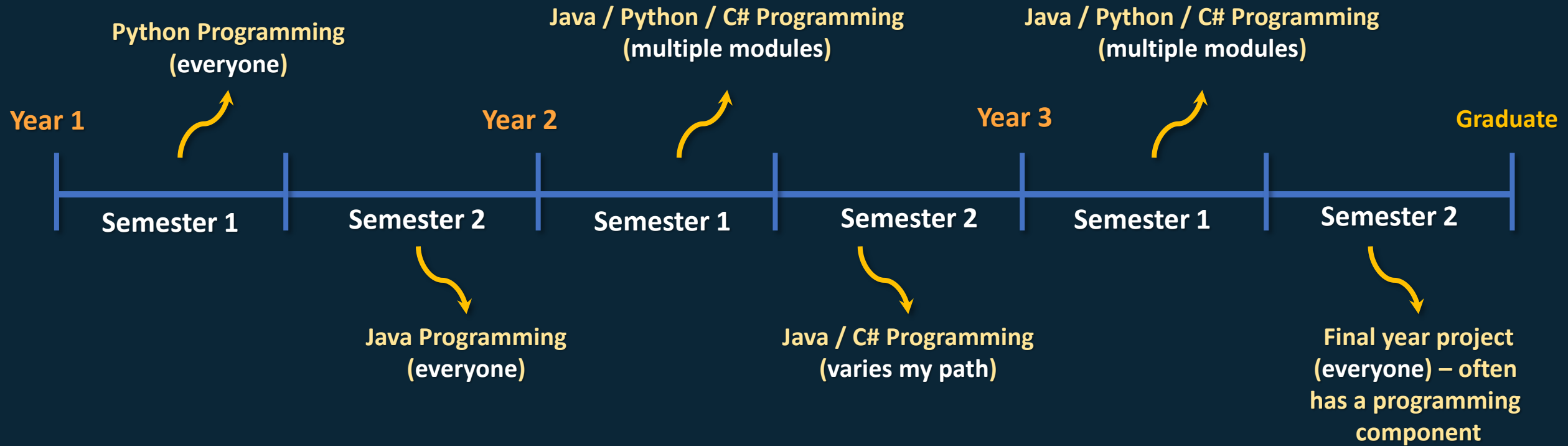


As of Sept. 2020, the median salary for a Python developer is £65,200.

The prospects are amazing – if you work hard.

Of the worlds 20 richest people, 8 are in software engineering.

Our Degree Paths



**Your module choices influence the programming
you do – but everyone does some!**

Programmes needing Programming Skills!

- BSc (Hons) Computer Engineering, Science, Science and Mathematics
- BSc (Hons) Computing
- BSc (Hons) Computing (Games Programming)
- BSc (Hons) Computing (Networking, Security & Forensics)
- BSc (Hons) Data Science
- BSc (Hons) Robotics & Artificial Intelligence
- BSc (Hons) Software Engineering
- MComp Computing
- BSc (Hons) Information Technology for Business

Desktop software

Algorithms

Mobile applications

Modelling

Web-applications

Games

Security software

Security testing

Data analyses

Intelligent systems

Automation

Sensors

Management

Control systems

E-Commerce

Review

- You now know what programming is – a process that produces code that we can use to **control a computer**.
- We know how programming relates to our degree programmes.
- We know why coding is **important**.

Next we turn our attention to giving you the chance to write some code – we'll first need to get access to a development environment that we can use to **write / run** code.

Coding Activities

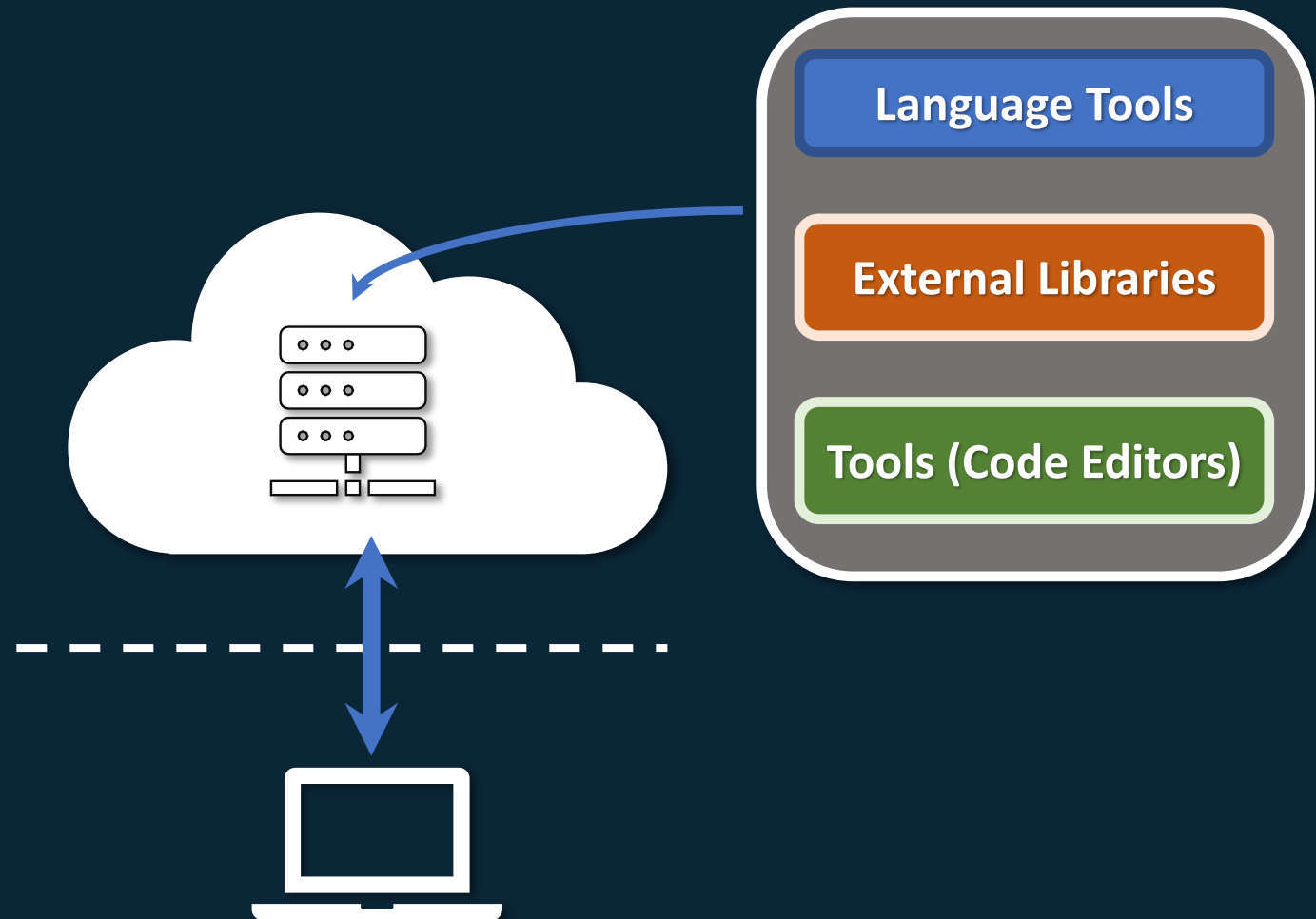
Hopefully, you'll find this fun and interesting!

Software Development Environment

- To develop code we need a couple of **tools**.
 1. A tool we can use to **write code** – a little like using Microsoft Word to write essays.
 2. A tool that **converts** the code files we write, into machine code.
 3. A tool that allows us to **run** the machine code.
- We typically think about these tools together as a single **development toolkit**.
- Such toolkits operate in a **development environment** – i.e. a software environment set up on a computer that we can use.
- We don't have the time to setup our own environments – instead, we're going to use a **pre-built, on-line development environment** that allows us to write Python code.
- This tool is called **Google Collaboratory**.

Google Collaboratory

- Google Collaboratory has it's own code editor, and tools that can be used to run code.
- You connect to this via the Google Chrome web browser.
- You may be interested to know that Google Collaboratory is a **cloud environment**.
- It is also a great environment for newcomers!



Activity 1.

Connect to the Google Collaboratory.

You need:

- A google account.
- The google Chrome web-browser installed.

The next slide has more details...

is great for learning.

Where to find the Colab?

Google Colab can be found at the following URL:
<https://colab.research.google.com/>

Before proceeding, please create a google account if you don't already have one. Once you've done that, connect to the Colab via the link above.

1. Once signed into the Colab, you likely be presented with this default page:

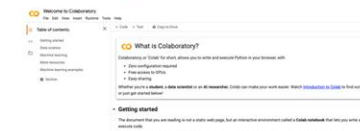


Figure 3. The default Google Colab page.

2. This default page is great, but we want to create our own Colab notebooks. Notebooks are files that contain the code and text we enter in to the Colab environment. The notebooks are stored inside Google Drive and not on your local computer. Though you can download and upload notebook files. To create our own notebook, head to the menu bar at the top of the screen. Click:

File -> New notebook.

CIS1111 – Programming: Concepts to Construction
Tutorial One

Edge Hill University

3. This will create an empty notebook file called something like "Untitled.pynb". Why not rename the notebook – click inside the box that holds the name of the notebook. Rename it to something more descriptive, e.g. CIS1111 First Notebook.ipynb. Don't alter the extension of the file (.ipynb) because that extension identifies the file type, and without it, the Colab won't open it.

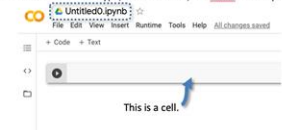


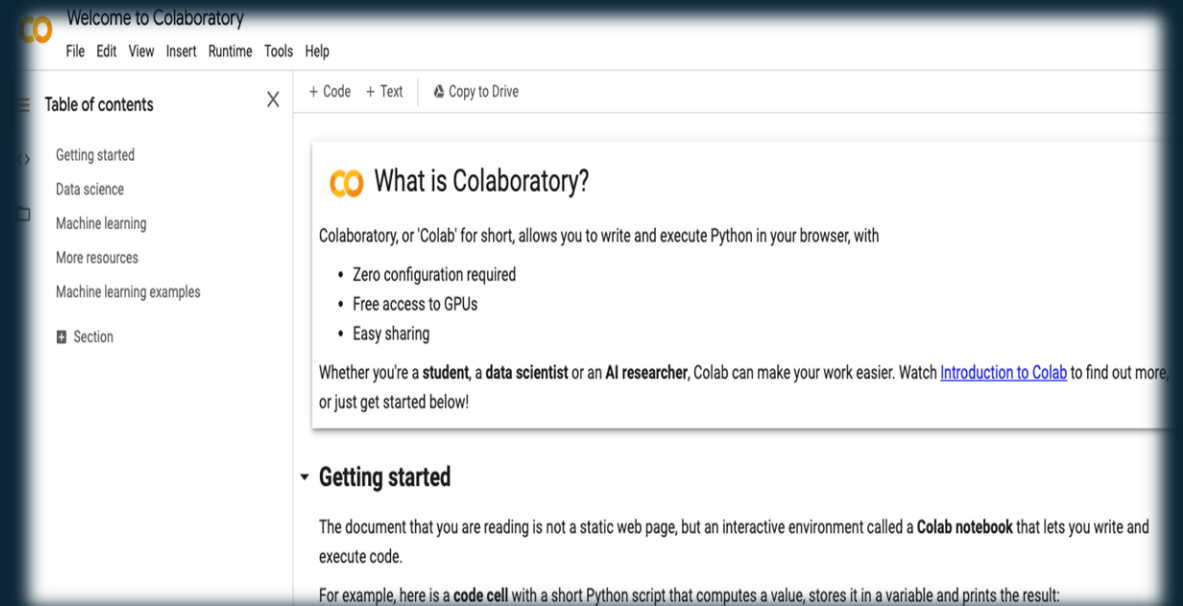
Figure 4. The area where you can rename the notebook, circled using a dashed blue line. I also show the first cell created in the notebook.

4. Notebooks are comprised of cells. Cells can contain code, or they can contain text.

Instructions are in the
word document provided.

Activity 1. Accessing the Collaboratory

1. Firstly open google Chrome – if you don't have Chrome, you'll need to install it via: https://www.google.com/intl/en_uk/chrome/.
2. Create a Google account if you don't have one, then sign-in to the Collaboratory environment via: <https://colab.research.google.com/>.
3. Once everyone is signed in, you should see a user interface that looks a little like this:
4. When everyone is ready, **let me know.**

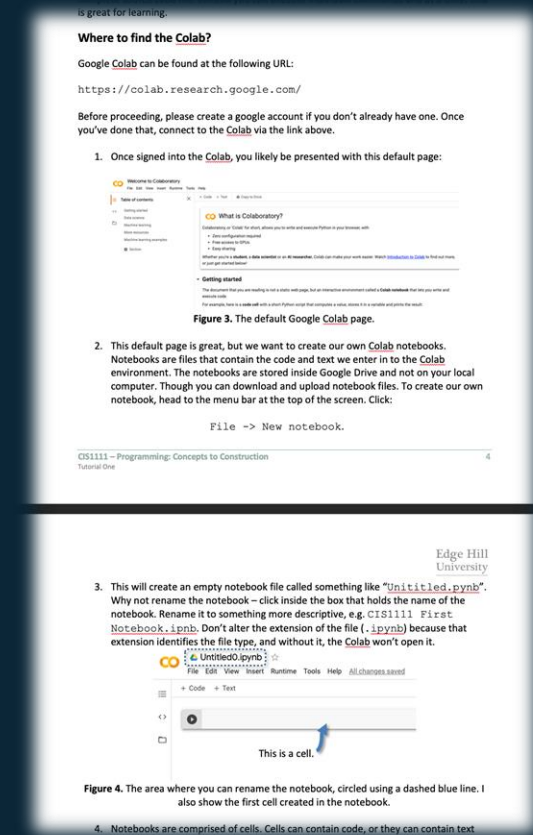


Activity 2.

Creating your own notebook.

You need:

- The instructions document to guide you, just in case you get stuck.

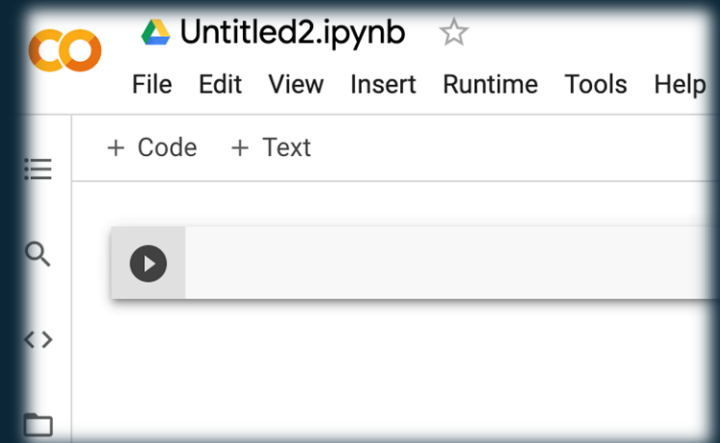


Instructions are in the
word document provided.

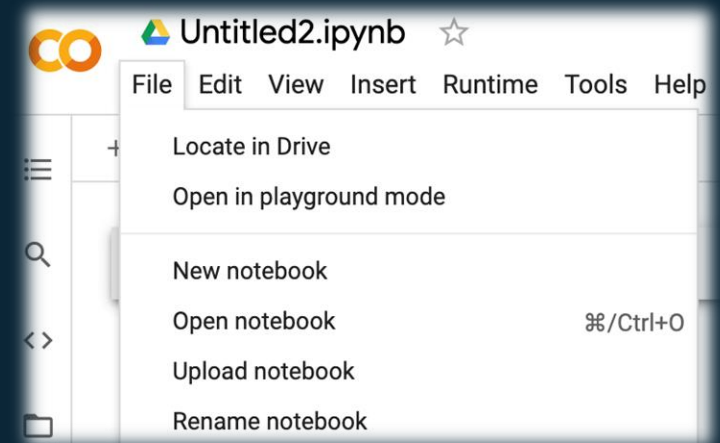
Activity 2. Creating a notebook

- A **notebook** is a coding environment within the Collaboratory.
- It's actually a special environment, because it allows you to **write code or normal text** within it.
- The notebook itself is represented within a **single file** – stored in the cloud (in Google Drive).
- Notebook files end with the `.ipynb` file extension.
- Next, I want you to create your own “**notebook**”.

1. Click the `File` menu at the top of the Collaboratory interface.
2. Select the `New Notebook` option.
3. Let me know when you've done this.



Creating a new notebook – click `File` -> `New Notebook`.

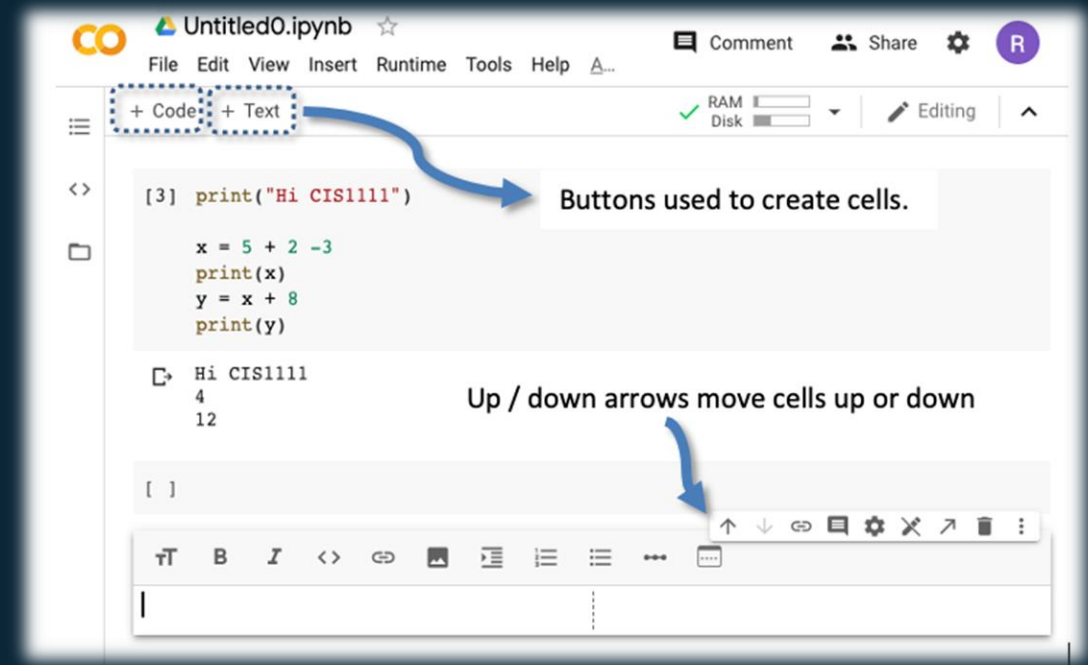


Demo 1.

Showing you around....

Sit back and watch as I talk you through the following concepts in the Google Collaboratory:

- A **cell** – a place where you can write code / text.
- How to “**run**” a cell and what that means.
- How to **write code** in a cell.



Remember: Instructions are in the word document provided.

What we've seen...

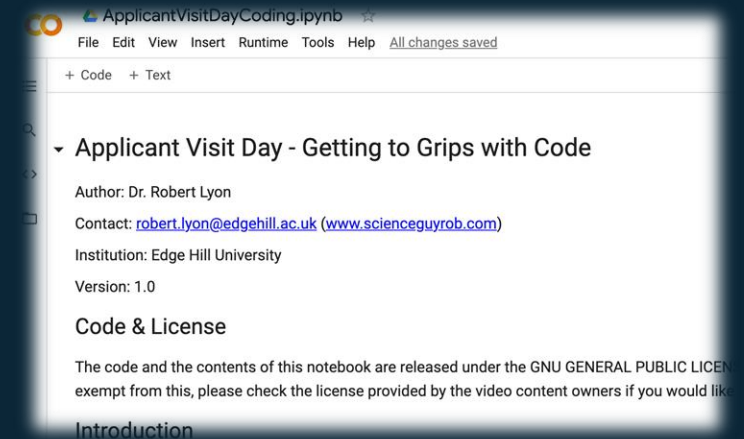
- We've now seen what a **text cell** is – how it can be used to provide information that helps explain our code.
- We've also seen a **code cell** and learned how to **run it** using the “**play**” button.
- We've also learned how we can **delete cells, move them** up or down, and how the environment is structured in general.
- Next, we get **you** working in this environment.
- To do that, we ask you to first **load a notebook** I made earlier....

Activity 3.

Loading the activity notebook.

You need:

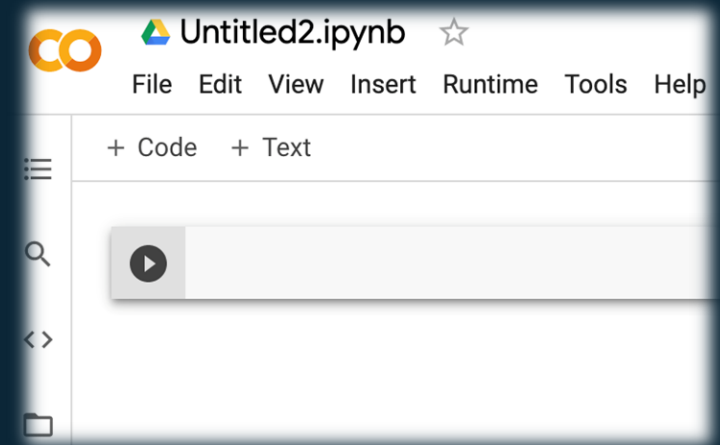
- The notebook file: `ApplicantVisitDay.ipynb` to be downloaded somewhere on your computer (e.g., your downloads folder).
- This file can be found in the **Blackboard environment** you're using.



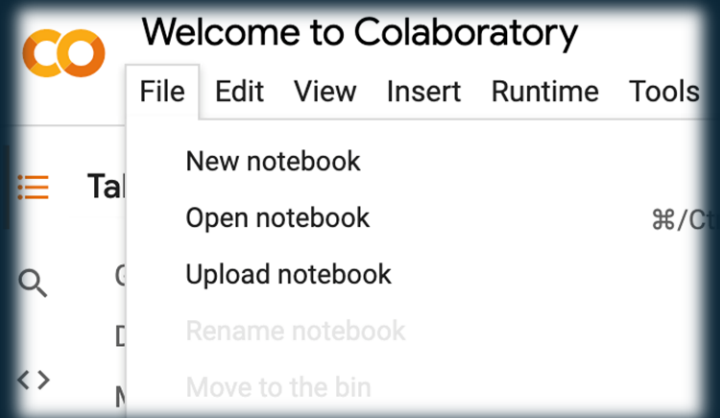
The notebook we're going to load.

Activity 3. Loading the notebook

1. Download `ApplicantVisitDay.ipynb` to some location on your personal computer. For instance, download it to your **downloads** folder.
2. Head to the Collaboratory environment.
3. Click the `File` menu at the top of the Collaboratory interface.
4. Select the `Open Notebook` option.
5. A file chooser dialog will open that allows you to choose a file to upload. Please upload the `ApplicantVisit.ipynb` file you saved to your computer in Step 1.
6. Let me know when you've done this.



Creating a new notebook – click `File` -> `Open Notebook`.



Next steps

- You should have loaded the `ApplicantVisitDay.ipynb` notebook successfully.
- The notebook itself contains instructions on what to do next – there are also some coding challenges and some learning material.
- It's your job to read through this, with my support, and to solve the challenges.
- We'll give you some time to do this, just like we would in class during a real lesson.
- Then with around 10 minutes to go, we'll come back together to finish up, and give you the opportunity to ask questions.
- So on your marks, get set.... GO!

Activity 4.

Tackling the notebook.

You need:

- To try and tackle the notebook activities for yourself – I'm here to help you and answer questions.
- If you've coded before and want a challenge... then let me know!



How fast can you get through the material :P

Reflecting on today

- You may not realise but today you've done the following:
 - Used an **industry standard software environment** for writing Python code – it is especially used in **Data Science** and for **Machine Learning** research.
 - Learned how to **run code** in this environment.
 - Learned how to **write some basic Python** statements in the Collaboratory environment.
 - Learned **why coding is so important** and how it relates to your (potential) degree programme.
 - Not bad for an hour? Maybe if you join us, you'll learn a whole lot more!

Follow-up Learning

- No matter where you decide to study, getting ahead start is beneficial.
- Here are some resources you can use to help prepare you and build your programming knowledge:
 - A **free** eBook called “Think Python”, by A. Downey, 2nd Edition, Green Tea Press.
 - Code academy – free programming tutorials available online:
<https://www.codecademy.com/>
 - The **Solo Learn** mobile application: <https://www.sololearn.com/>

Learning Aims

- By the end of this session I want you to understand / be aware of:
 - The basic principles computer programming – how we write code.
 - Why we write code (to control a CPU) and what that gives us.
 - How coding is used throughout our degree programmes.
 - How coding is beneficial to you and your career / employability.
 - A tool used to write code – Google Collaboratory.
 - How to write some basic computer programs in the Python programming language.

Applicant visit day – DONE.

Find out more details about our department on-line.

I hope you enjoyed today and now understand what it might be like to study with us!

Do you have any questions?