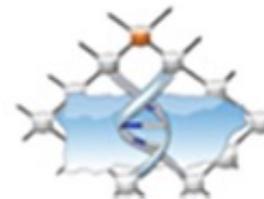




University of Puerto Rico

Puerto Rico - IDeA Networks of Biomedical Research Excellence



PRINBRE
IDeA Network of Biomedical Research Excellence

Python3 Part 1 – Crash Course in Python3 for Future STEM Coders

Prof. Judith S. Rodriguez-Martinez, MSc, PhD

Ph.D. Candidate

Penn State University – University Park Campus
jzr5814@psu.edu



June 1-6, 2025



PennState
Huck Institutes of
the Life Sciences



PennState
Huck Institutes of
the Life Sciences

- The following material is the result of PR-INBRE research and curriculum development effort to provide a set of educational materials for research training and curriculum changes in biology programs across the island to support PR-INBRE efforts to establish a Community of Practice in Bioinformatics that offers a fruitful environment to increase computational and bioinformatics skills among traditional researchers and students (undergraduate and graduate) in the island. They have been developed as a part of the NIH funded project **“Puerto Rico IDeA Network Biomedical Research Excellence (PRINBRE)”** (Award Number 5P20GM103475).
- Unless otherwise specified, all the information contained within is Copyrighted © by University of Puerto Rico. Permission is granted for use, modify, and reproduce these materials for research and teaching purposes. A copy of the modified material should be sent to help@hpcf.upr.edu.
- Most recent versions of these presentations can be found at <http://inbre.hpcf.upr.edu/>.





PennState
Huck Institutes of
the Life Sciences

Competencies

Record and write simple and common **Python scripts**
to deal with Bioinformatic needs for biological
data analyses using a Jupyter Notebook

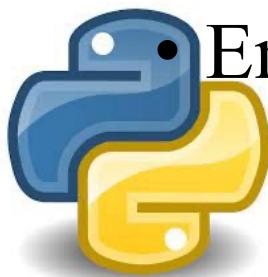




PennState
Huck Institutes of
the Life Sciences

Objectives

- Use **Google Colab** as an **environment** to practice and learn common **Python** lines of code.
 - Formulate simple calculations using **Python**
 - Identity the following **datatypes**: integer, float, and string
- Employ **variables** for different datatypes





PennState
Huck Institutes of
the Life Sciences

Objectives

- Manipulate integer, float, and string datatypes
- Use **bioinformatic tools** to find a protein-coding sequence
- **Evaluate a protein-coding sequence** of interest and calculate its GC content.



• **Debug** code



PennState
Huck Institutes of
the Life Sciences

Target Audience

- This training is addressed to beginners, highly motivated (eager to learn what is needed in order to be competitive without the tendency to self-limit when learning computational skills by saying that is difficult) wanting to become familiar with the **Python** programming language and become the Script Master of their bioinformatic analysis.

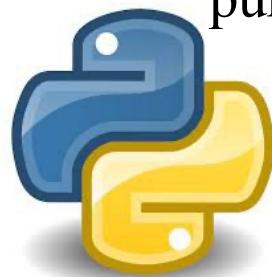




PennState
Huck Institutes of
the Life Sciences

The importance of Python

- R is great; however, you'll eventually run into the Python language
- No matter your field, python will be a necessity (chemistry, biology, engineering, ecology, mathematics, etc)
- Knowing Python gives you an edge when job searching and navigating graduate research
- The majority of bioinformatic tools are python-based and will lead you navigate under the hood code to successfully execute said program's purpose.





PennState
Huck Institutes of
the Life Sciences

Lesson 1.0.0

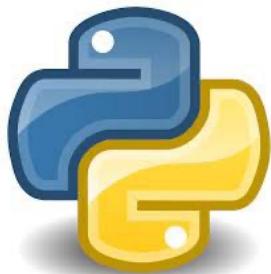
Connecting to

Google Colab to learn

common Python lines

of code

(Gmail account ready for colab.research.google.com)



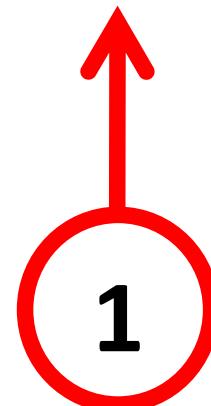


PennState
Huck Institutes of
the Life Sciences



Google Colab

<https://colab.research.google.com>



Activate the link



Activate colab.research.google.com

Screenshot of a web browser showing the Colab interface at colab.research.google.com.

The browser toolbar includes standard icons for back, forward, search, and refresh.

The main header bar shows the URL colab.research.google.com, a shield icon, and a user profile picture.

The top navigation bar includes "Welcome To Colaboratory" and a "CO" logo, followed by "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right are "Share", "Settings", and a user profile icon.

The left sidebar features a "Table of contents" section with the following categories:

- Getting started
- Data science
- Machine learning
- More Resources
 - Featured examples
- Section

Below the sidebar, the main content area displays the "Welcome to Colab!" page.

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.



What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

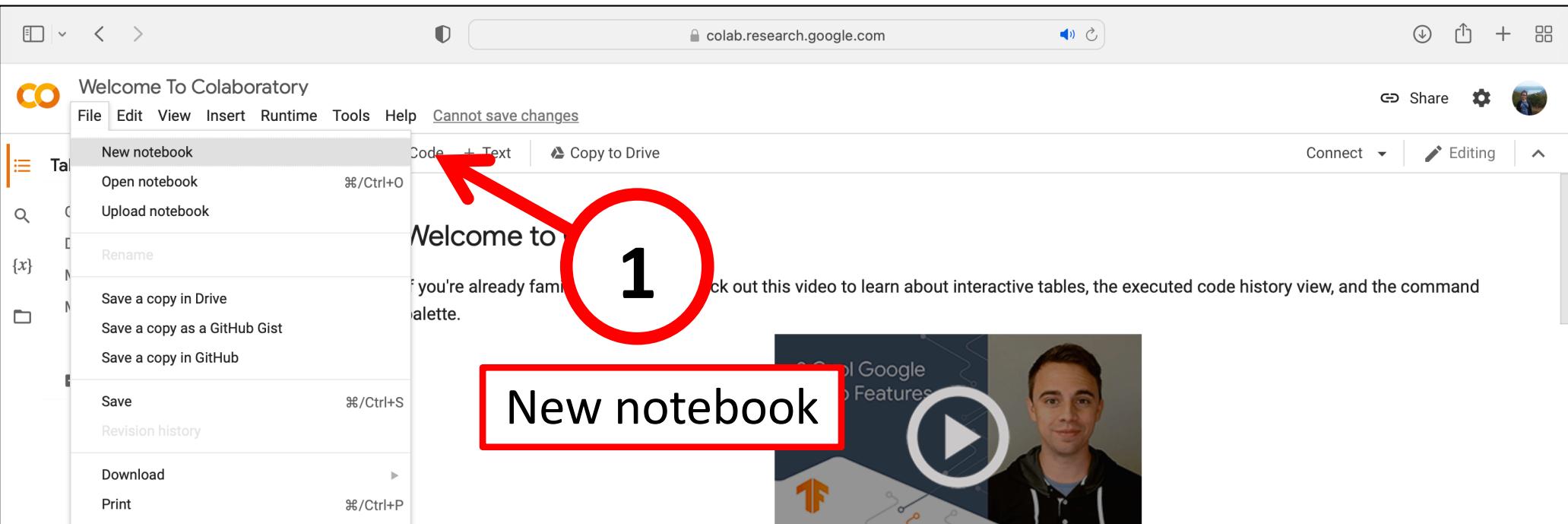
Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

10

Open a New notebook



The screenshot shows the Google Colab interface. A large red circle highlights the "New notebook" button in the top navigation bar. A red arrow points from this circle to a larger red circle containing the number "1" on the left side of the screen. Below the "New notebook" button, there is a video thumbnail featuring a man's face and the text "Google Features".

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help Cannot save changes

New notebook

Code + Text

Copy to Drive

Connect

Editing

1

New notebook

Welcome to Colab!

If you're already familiar with Jupyter notebooks, check out this video to learn about interactive tables, the executed code history view, and the command palette.

What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

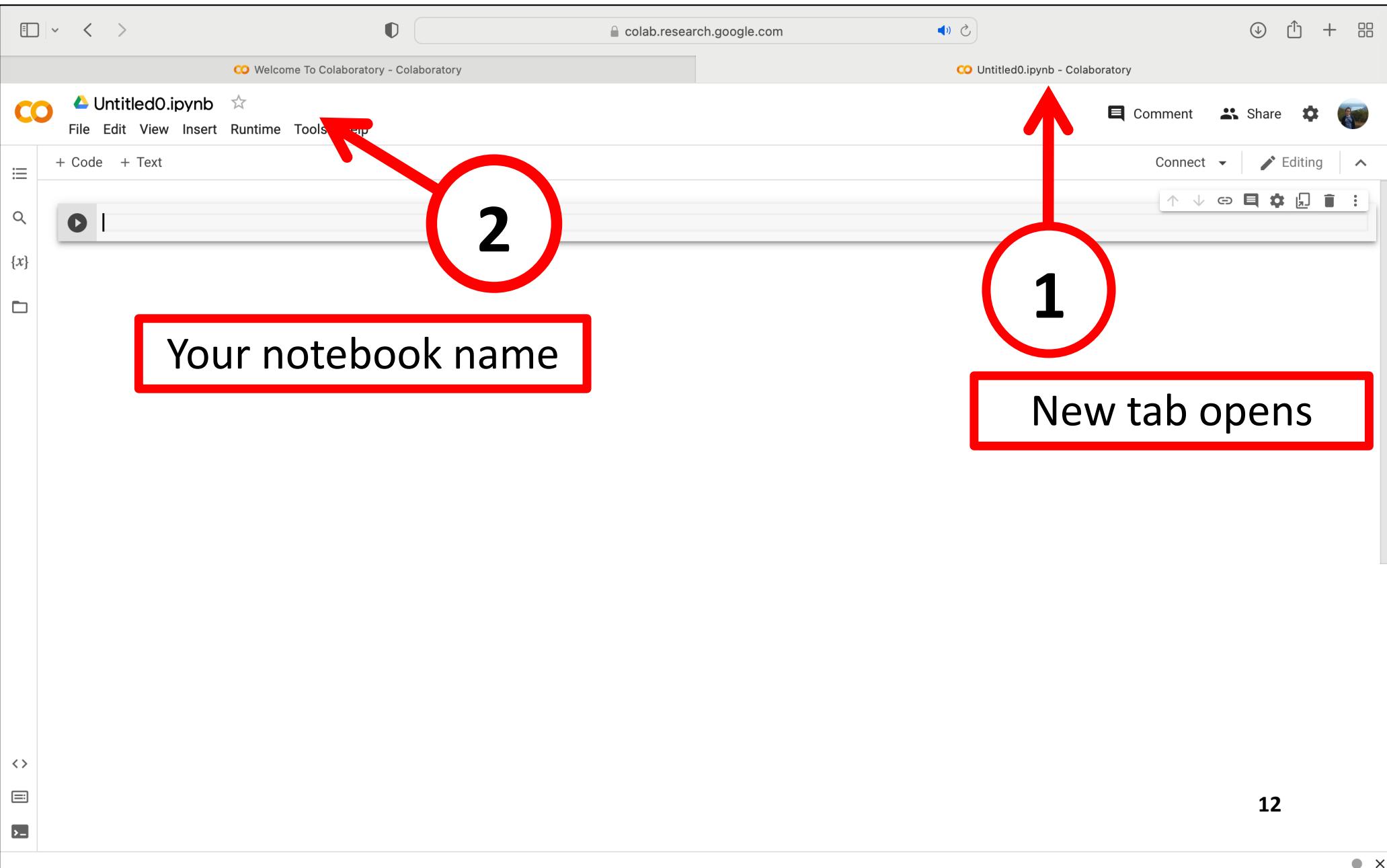
Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

11

Welcome to your new notebook!

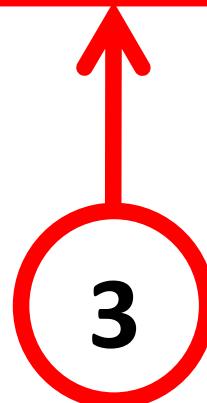




PennState
Huck Institutes of
the Life Sciences



[https://github.com/bioinfwithjudith/PR-
INBRE Python Workshop](https://github.com/bioinfwithjudith/PR-INBRE-Python-Workshop)



In a new tab, activate the link



This is a GitHub repository!

The screenshot shows a GitHub repository page for 'PR-INBRE_Python_Workshop'. The repository is public and has 1 branch and 0 tags. It was last updated 12 minutes ago by bioinfwithjudith. The repository contains files: PR-INBRE_2023, PR-INBRE_2024, README.md, and Student_PART_1_Python3_Workshop.ipynb. The README file is currently selected. The page also includes sections for About, Releases, Packages, and Languages.

About

This workshop was developed for PR-INBRE.

Files

File	Last Commit
PR-INBRE_2023	2025 update
PR-INBRE_2024	2025 update
README.md	2025 update
Student_PART_1_Python3_Workshop.ipynb	updated files for students

PR-INBRE Python Workshop 2025

This workshop is a crash course designed for beginner python coders in the STEM field. The development of sequencing methods and decreasing costs leads to the need of scientists knowing how to code. Not only is it important for scientists to know how to develop and execute code but scientists may run into looking at another scientists code and being able to decipher what's going on for their own research benefit.

Part 1 - The use of a Python Interpreter and running your first set of commands.

About

This workshop was developed for PR-INBRE.

Readme

Activity

Stars 0

Watching 1

Forks 0

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

Jupyter Notebook 100.0%

Download Jupyter Notebook

The screenshot shows a GitHub repository page for 'PR-INBRE_Python_Workshop'. The repository is public and has 1 branch and 0 tags. The main file listed is 'Student_PART_1_Python3_Workshop.ipynb'. A red arrow points from a large red circle containing the number '4' to this file. The page includes sections for About, Releases, Packages, and Languages.

About

This workshop was developed for PR-INBRE.

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

Jupyter Notebook 100.0%

PR-INBRE Python Workshop

This workshop is a crash course designed for beginner python coders in the STEM field. The development of sequencing methods and decreasing costs leads to many opportunities for learning how to code. Not only is it important for scientists to know how to develop and run their own code, but it is also important for them to be able to understand other scientists code and being able to decipher what's going on.

Part 1 - The use of a Python Interpreter and running your first set of commands.

Click

Download Jupyter Notebook

The screenshot shows a GitHub repository page for 'bioinfwithjudith / PR-INBRE_Python_Workshop'. The 'Code' tab is selected. On the right, a Jupyter notebook file named 'Student_PART_1_Python3_Workshop.ipynb' is displayed. The notebook content includes:

- WELCOME TO THE FIRST DAY OF OUR PYTHON CRASH COURSE!**
- Lesson 1.1.0 Testing snippets of code using the Python Interpreter**
- Example 1**: How many seconds(s) are in 10 milliseconds (ms)?
In []: (empty input field)
- Problem 1**: How many 5µL are in mL?
In []: (empty input field)
- Problem 2**: How many kg are in 0.0034g?
In []: (empty input field)

At the top right of the notebook view, there is a 'Download raw file' button. A large red arrow points to this button, and it is enclosed in a red circle with the number 5. To the right of the notebook content, there is a red box containing the text 'Download notebook'.

Upload existing notebook

The screenshot shows the Google Colab interface. A red arrow points from the top center towards the 'File' menu. Inside the 'File' menu, the 'Upload notebook' option is highlighted with a red circle containing the number '6'. Below the menu, the main content area displays a 'Welcome to Colab!' message and a section titled 'Upload notebook' which is also highlighted with a red border.

Welcome To Colab

File Edit View Insert Runtime Tools Help

New notebook in Drive

Open notebook ⌘/Ctrl+O

Upload notebook

Rename

Save a copy in Drive

Save a copy as a GitHub Gist

Save a copy in GitHub

Save ⌘/Ctrl+S

Revision history

Download ⌘/Ctrl+P

Print

Welcome to Colab!

Explore the Gemini API

Gemini API generates multimodal, so you can seamlessly across text, images, code, and audio.

6

Upload notebook

Cover Gemini's advanced capabilities

- Play with Gemini [multimodal outputs](#), mixing text and images in an iterative way.
- Discover the [multimodal Live API](#) (demo [here](#)).
- Learn how to [analyze images and detect items in your pictures](#) using Gemini (bonus, there's a [3D version](#) as well!).
- Unlock the power of [Gemini thinking model](#), capable of solving complex task with its inner thoughts.

Explore complex use cases

- Use [Gemini grounding capabilities](#) to create a report on a company based on what the model can find on internet.
- Extract [invoices and form data from PDF](#) in a structured way.
- Create [illustrations based on a whole book](#) using Gemini large context window and Imagen.

To learn more, check out the [Gemini cookbook](#) or visit the [Gemini API documentation](#).

Colab now has AI features powered by [Gemini](#). The video below provides information on how to use these features, whether you're new to

{ } Variables Terminal

Remember this for the next few days!

The screenshot shows a Google Colab notebook interface. The title bar indicates the file is titled "Student_PART_1_Python3_Workshop.ipynb". The left sidebar contains a "Table of contents" section with several items listed under "WELCOME TO THE FIRST DAY OF OUR PYTHON CRASH COURSE!":

- Lesson 1.1.0 Testing snippets of code using the Python Interpreter
 - Example 1
 - Problem 1
 - Problem 2
 - Problem 3
- Lesson 1.2.0. Python datatypes (Integers and floats and strings...oh my!)
 - Identify datatypes of Python objects using type() functions.
 - Run this classic string!
- Assign a variable to this string and print variable
- Lesson 1.3.0 Evaluate the GC content of a DNA sequence
 - Get to know the CRTAM sequence
- Create a variable for your favorite protein-coding gene sequence.

The main content area displays the following text:

WELCOME TO THE FIRST DAY OF OUR PYTHON CRASH COURSE!

- > **Lesson 1.1.0 Testing snippets of code using the Python Interpreter**
[] ↴ 8 cells hidden
- > **Lesson 1.2.0. Python datatypes (Integers and floats and strings...oh my!)**
[] ↴ 8 cells hidden
- > **Lesson 1.3.0 Evaluate the GC content of a DNA sequence**
[] ↴ 10 cells hidden
- > **Lesson 1.4.0 Running into errors**
[] ↴ 1 cell hidden
- > **Proficiency Assessment**

At the bottom of the screen, there are tabs for "Variables" and "Terminal".



PennState
Huck Institutes of
the Life Sciences

Lesson 1.1.0

Test snippets of code
using the Python
interpreter



Python is your new calculator!

The image shows a screenshot of a Python Colab notebook interface. On the left, there's a sidebar with icons for file operations, code, text, and search. A large Python logo is at the bottom left. The main area has a title bar with 'Welcome To Colaboratory - Colaboratory'. Below the title bar, there's a menu bar with 'File', 'Edit', 'View', 'Code', and 'Text'. A red box highlights a list of arithmetic operators:

Addition	+
Subtraction	-
Multiplication	*
Division	/
Modulus	%
Floor division	//
Exponent	**

At the bottom right, the number '20' is visible.

Let's solve metric conversion problems!

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** CO OX-Python3-Workshop.ipynb
- Menu Bar:** File Edit View Insert Runtime Tools Help
- Toolbar:** + Code + Text, search icon, and a dropdown menu {x}.
- Table of Contents:** Metric Conversion Problems (expanded) and Example 1 (expanded).
- Content Area:** A question: "How many seconds(s) are in 10 milliseconds (ms)?"
- Execution Area:** A code cell with a play button icon and a red circle containing the number 1.
- Buttons:** A red box labeled "Type equation" with an arrow pointing to the play button, and a red circle containing the number 2 with an arrow pointing to a "Run" button.
- Python Logo:** A blue and yellow Python logo icon is located at the bottom left.

Red annotations are present: a red circle with the number 1 points to the play button; a red box with the text "Type equation" has an arrow pointing to the play button; a red circle with the number 2 points to the "Run" button.

Let's solve metric conversion problems!

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** CO OX-Python3-Workshop.ipynb
- Menu Bar:** File Edit View Insert Runtime Tools Help All changes saved
- Toolbar:** + Code + Text
- Table of Contents:**
 - Metric Conversion Problems
 - Example 1
- Text Area:** How many seconds(s) are in 10 milliseconds (ms)?
- Code Cell:** 10 * (1/0.001)
- Output Cell:** 10000.0
- Note Callout:** A red callout box on the right contains the text: "Note: Python interprets the numbers as integers or floats".
- Python Logo:** A large Python logo is visible on the left side of the slide.

A red circle with the number 3 is overlaid on the output cell, and a red arrow points from it to the word "Answer" which is enclosed in a red box.

Try it on your own!

▼ Problem 1

How many $5\mu\text{L}$ are in mL?

ANSWER



Try it on your own!

▼ Problem 1

How many 5 μ L are in mL?

```
[ ] 5 * (0.001/1)
```

0.005



Try it on your own!

▼ Problem 2

How many kg are in 0.0034 g?

ANSWER



Try it on your own!

▼ Problem 2

How many kg are in 0.0034 g?

```
[ ] 0.0034 * (0.001/1)
```

3.4e-06



Try it on your own!

▼ Problem 3

How many moles are there in 50g of water(H₂O)?

Tip: Use variables

ANSWER



Try it on your own!

▼ Problem 3

How many moles are there in 50g of water(H₂O)?

Tip: Use variables

Note: Facilitate
your programming
employing
variables!



#Given:

mass_of_h2o = 50

molecular_mass_of_h2o = (1)*2 + 16

#Solution

```
result = mass_of_h2o/molecular_mass_of_h2o  
result
```

2.7777777777777777





PennState
Huck Institutes of
the Life Sciences

Lesson 1.2.0

Python datatypes



Integers and floats and strings... oh my!

Tip: Knowing the function `type(object)` of objects you are working with during coding can help you get around programming obstacles!

Datatypes

<code>int()</code>	12
<code>float()</code>	12.0
<code>str()</code>	"hola"
<code>list()</code>	[1,2,3, "hola"]
<code>dictionary()</code>	{"blue":"azul"}



Integers and floats and strings... oh my!

- ▼ Identify datatypes of Python objects using type() functions.

Answer



Integers and floats and strings... oh my!

- ▼ Identify datatypes of Python objects using type() functions.

```
[15] type(mass_of_h2o)  
int
```



Integers and floats and strings... oh my!

- ▼ What is the datatype of molecular_mass_of_h2o?

Answer



Integers and floats and strings... oh my!

- ▼ What is the datatype of molecular_mass_of_h2o?

```
[16] type(molecular_mass_of_h2o)  
int
```



Int and Float Types can be manipulated

int(object)
float(object)

How many seconds(s) are in 10 milliseconds (ms)?

```
✓ [23] 10 * (1/0.001)  
0s
```

10000.0

```
✓ [30] seconds = 10 * (1/0.001)  
    int(seconds)  
0s
```

10000

▼ Identify datatypes of Python objects using type() functions.

```
✓ [31] mass_of_h2o  
0s
```

50

```
✓ [27] type(mass_of_h2o)  
0s
```

int

```
✓ [29] float(mass_of_h2o)  
0s
```

50.0



Integers and floats and strings... oh my!

✓ [33] 'hola puerto rico'

0s

```
'hola puerto rico'
```

- ▼ Assign a variable to this string and print variable

Hint: Search for python's print() function

Answer



Integers and floats and strings... oh my!

✓ [33] 'hola puerto rico'

0s

```
'hola puerto rico'
```

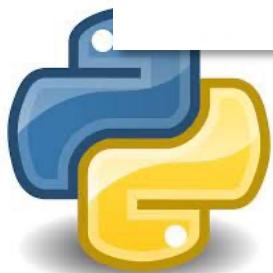
- ▼ Assign a variable to this string and print variable

Hint: Search for python's print() function

✓ [34] greeting = 'hello puerto rico'
print(greeting)

0s

```
hello puerto rico
```



There are a multitude of **functions** to evaluate and manipulate your **strings**

`len(str)`

`str.count("")`

`str.replace("", "")`

`str.capitalize()`

`str.split("")`

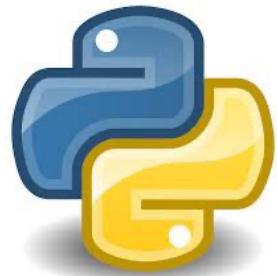
`"".join(str)`

Note: These are my favorite and most used **string functions!**

Indexing a String

Strings have a property called indexes, which are positions in the string

```
[4] sequence="ATGTGGTGG"
```

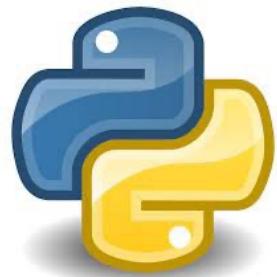


Indexing a String

Strings have a property called indexes, which are positions in the string

```
[4] sequence="ATGTGGTGG"  
sequence
```

```
' ATGTGGTGG '
```



Indexing a String

Strings have a property called indexes, which are positions in the string

```
[4] sequence="ATGTGGTGG"  
sequence
```

```
' ATGTGGTGG '  
012345678
```



Indexing a String

Strings have a property called indexes, which are positions in the string

```
[4] sequence="ATGTGGTGG"  
sequence
```

' ATGTGGTGG '
012345678
↑

The index of the
first position is 0



Indexing a String

The index of the first position is 0

'ATGTGGTGG'
012345678

[5] sequence[0]

Call the character at index 0

'A'



Indexing a String

The index of the last position is can be 8 or -1

' ATGTGGTGG '
012345678

[7] sequence[8]

Call the last character using 8

' G '

[8] sequence[-1]

Call the last character using -1

' G '

Note: Using -1 is useful when last index is unknown!

Indexing a String

Index a range of the sequence

' ATGTGGTGG '
012345678

[9] sequence[2:8]

Index is included and starts new string

Index is not included in new string

' GTGGTG '





PennState
Huck Institutes of
the Life Sciences

Lesson 1.3.0

Evaluate the GC content of a DNA sequence.



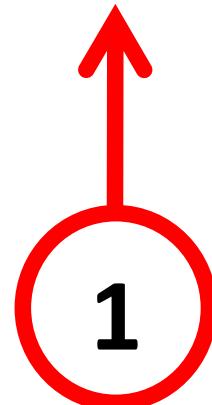


PennState
Huck Institutes of
the Life Sciences



Connect to Uniprot

<https://www.uniprot.org>



Activate link



Activate uniprot.org

UniProt BLAST Align Peptide search ID mapping SPARQL Release 2023_01 | Statistics 📄 🛒 📧 Help

Find your protein

UniProtKB ▾ Advanced | List Search Examples: Insulin, APP, Human, P05067, organism_id:9606 Feedback

UniProt is the world's leading high-quality, comprehensive and freely accessible resource of protein sequence and functional information. [Cite UniProt™](#)

Proteins
UniProt Knowledgebase

Species
Proteomes

Protein Clusters
UniRef

Sequence Archive
UniParc



CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

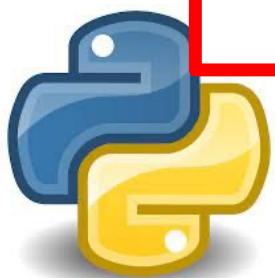
Find your protein

UniProtKB ▾ human crtam

Examples: Insulin, AFP_Human, P05067, organism_id:9606

2

Search CRTAM



CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

UniProt BLAST Align Peptide search ID mapping SPARQL UniProtKB ▾ human crtam Advanced | List Search

status
Reviewed (Swiss-Prot) (5)
Unreviewed (TrEMBL) (11)

popular organisms
Human (5)
Mouse (2)
Zebrafish (1)

economy
ter by
Proteins (11)
D structure (3)
Alternative products (isoforms) (1)
Alternative splicing (5)
Beta strand (3)
Binary interaction (5)

UniProtKB 16 results

BLAST Align Map IDs Download Add View: Cards Table Share

Select how you would like to view your results

Cards Table

3

Entry Name Protein Names Gene Names Organism Length

A2VEY9 APP_DROME Palmitoyltransferase app - Drosophila melanogaster (Fruit fly) - EC:2.3.1.225 - Gene: app - 693 amino acids - Evidence at protein level - Annotation score: 204508 / 5288c [Baker's yeast] 1 domain - 1 active site - 4 isoforms - 7 reviewed variants

P53933 APP1_YEAST Palmitoyltransferase APP1 - Saccharomyces cerevisiae (strain ATCC 204508 / S288c) [Baker's yeast] 1 domain - 1 active site - 4 isoforms - 7 reviewed variants

Q9P7EB APP1_SCHPO Protein app1 - Schizosaccharomyces pombe (strain 972 / ATCC 24842) [Fission yeast] 1 domain - 1 active site - 4 isoforms - 7 reviewed variants

Q9BY24 APPB2_HUMAN Amyloid protein-binding protein 2 [Human] APPBP2 1 domain - 1 active site - 4 isoforms - 7 reviewed variants

O44750 XPP_CAEEL Xaa-Pro aminopeptidase app-1 [C. elegans] app-1 1 domain - 1 active site - 4 isoforms - 7 reviewed variants

P05067 A4_HUMAN Amyloid-beta precursor protein [Human] APP 1 domain - 1 active site - 4 isoforms - 7 reviewed variants

P51972 PA2B1_AGRKPI Basic phosphatase A2 APP-D49 1 domain - 1 active site - 4 isoforms - 7 reviewed variants

Q13564 ULA1_HUMAN NEDD8-activating enzyme E1 regulatory subunit 1 NAE1 1 domain - 1 active site - 4 isoforms - 7 reviewed variants

View results

Gene Names	Organism	Length
CRTAM	Homo sapiens (Human)	393 AA
CADM1, IGSF4, GSF4A, NECL2, SYNCAM, TSLC1	Homo sapiens (Human)	444 AA
SCRIB, CRIB1, KIAA0147, LAP4, SCRIB1, VARTUL	Homo sapiens (Human)	1,066 AA
Scrib, Kiaa0147, Lap4, Scrib1	Mus musculus (Mouse)	1,066 AA
Cadm1, Igsf4, Necl2, Ra175, Syncam, SynCam1, Tslc1	Mus musculus (Mouse)	453 AA
CRTAM	Pan paniscus (Pan)	393 AA

CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

UniProt BLAST Align Peptide search ID mapping SPARQL UniProtKB ▾ human crtam Advanced | List Search Help

Status
Reviewed (Swiss-Prot) (5)
Unreviewed (TrEMBL) (11)

Popular organisms
Human (5)
Mouse (2)
Zebrafish (1)

Taxonomy
Filter by taxonomy

Proteins with
3D structure (3)
Alternative products (isoforms) (5)

UniProtKB 16 results

Entry ▲ Entry Name ▲ Protein Names ▲ Gene Names ▲ Organism ▲ Length ▲

Entry	Entry Name	Protein Names	Gene Names	Organism	Length
O95727	CRTAM_HUMAN	Cytotoxic and regulatory T-cell molecule [...]	CRTAM	Homo sapiens (Human)	393 AA
Q14160	CADM1_HUMAN	Cell adhesion molecule 1 [...]	CADM1, IGSF4, IGSF4A, NECL2, SYNCAM, TSLC1	Homo sapiens (Human)	442 AA
Q14160	SCRIB_HUMAN	Protein scribble homolog [...]	SCRIB, CRIB1, KIAA0147, LAP4, SCRIB1, VARTUL	Homo sapiens (Human)	1,630 AA
Q8R5M8	SCRIB_MOUSE	Protein scribble homolog [...]	Scrib, Kiaa0147, Lap4, Scrib1	Mus musculus (Mouse)	1,612 AA
Q8R5M8	CADM1_MOUSE	Cell adhesion molecule 1 [...]	Cadm1, Igsf4, Necl2,	Mus musculus	456 AA

Activate link to Entry



CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

UniProt BLAST Align Peptide search ID mapping SPARQL UniProtKB Advanced | List Search Help

Function 095727 · CRTAM_HUMAN

Names & Taxonomy Proteinⁱ Cytotoxic and regulatory T-cell molecule Amino acids 393
Subcellular Location Geneⁱ CRTAM Protein existenceⁱ Evidence at protein level
Disease & Variants Statusⁱ UniProtKB reviewed (Swiss-Prot) Annotation scoreⁱ 5/5
PTM/Processing Organismⁱ Homo sapiens (Human)

Expression Entry Feature viewer Publications External links History

Interaction BLAST Align Download Add Add a publication Entry feedback

Structure

Family & Domains Functionⁱ

Sequence & Isoform Mediates heterophilic cell-cell adhesion which regulates the activation, differentiation and tissue retention of

Similar Proteins Interaction with CADM1 promotes natural killer (NK) cell cytotoxicity and IFNG/interferon-gamma secretion by NK cells (By similarity).
Cell-mediated rejection of tumors expressing CADM1 in vivo (PubMed:[15811952](#)).
In vitro as well as NK

5

Scroll



CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

PTM/Processing

Expression

Interaction

Structure

Family & Domains

Sequence & Isoform

Similar Proteins

Sequence databases

CCDS | CCDS76489.1 ↗ [O95727-2]
CCDS8437.1 ↗ [O95727-1]

RefSeq | NP_001291711.1 ↗ NM_0[O95727-2]
NP_062550.2 ↗ NM_01960

SEQUENCE	PROTEIN	MOLECULE TYP
AF001622 (EMBL ↗ GenBank ↗ DDBJ ↗)	AAC80267.1 (EMBL ↗ GenBank ↗ DDBJ ↗)	mRNA
AB209830 (EMBL ↗ GenBank ↗ DDBJ ↗)	BAD93067.1 (EMBL ↗ GenBank ↗ DDBJ ↗)	mRNA
BC070266 (EMBL ↗ GenBank ↗ DDBJ ↗)	AAH70266.1 (EMBL ↗ GenBank ↗ DDBJ ↗)	mRNA

6

Activate link to EMBL



CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

The ENA Advanced Search API is changing on 2023-05-02! Details [here](#).

Sequence: AF001622.1

Homo sapiens class-I MHC-restricted T cell associated molecule (CRTAM) mRNA, complete cds.

Organism:	Homo sapiens (human)
Accession:	AF001622
Mol Type:	mRNA
Topology:	LINEAR;linear
Base Count:	2425
Dataclass:	STD
Tax Division:	HUM
Chromosome:	11
Md5 Checksum:	f28e47cb1576f4d6aaaf50b83f6cacbd8
Map:	11q22-q23

View: EMBL FASTA
Download: EMBL FASTA
Navigation: Show
Publications: Show
Sequence Versions: View

6

Activate link to FASTA

CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

>ENA|AF001622|AF001622.1 Homo sapiens class-I MHC-restricted T cell associated molecule (CRTAM) mRNA, complete cds.
ATGGTGGAGAGCTCTCAGCTGGCATGGTCCCTGCAAGAGGCCTCTGACT
AACCACACAGAAACCATCACCGTGGAGGAAGGCCAGACGCTCACTCTAAAGTGTGTCACT
TCTCTGAGGAAGAACCTCCTCCCTCAGTGCTGACCCCTCAGGGTTCACCATTTTTTA
AATGAGTATCCTGCTTAAAAAATTCCAATACCACTAGCTTCTCATCACTCGGCCAATCAG
CTCTCCATCACTGTGCCAACGTAAACCTGCAAGATGAAGGCGTGTACAAGTGCTTACAT
TACAGCGACTCTGTAAGCACAAAGGAAGTGAAGTGATTGTGCTGGCAACTCCTTCAAG
CCAATCTGGAAGCTCAGTTATCAGAAAGCAAATGGAGAAGAACATGTTGACTCATG
TGCTCCACCATGAGAACAGCCCCCTCCGAGATAACCTGGCTACTTGGGAATAGCATG
GAAGTGTCCGGTGGAACGCTCATGAATTGAAACTGATGGAAAGAAATGTAATACTACC
AGCACTCTCATAATCCACACTTATGGCAAAAATTCAACGGTGGACTGCATTATCCGACAC
AGAGGCCTGCAAGGGAGAAAAGTAGTACGACCCCTCCGGTTGAAGATTGGTTACTGAT
GAAGAGACAGCTCAGATGCTCTGGAGAGAAAACTCTCATCCTCAAGAACCCACAGCAG
CCCACCACTACTGTCAGTAACGGAAGATTCTAGTACATCGGAGATTGACAAGGAAGAG
AAAGAACAAACCACTCAAGATCCTGACTTGACCACCGAAGCAAATCCTCAGTATTTAGGA
CTGGCAAGAAAGAAAAGTGGCATCCTGCTGCTCACGCTGGTGTCTCCTCATTTCTATA
CTCTTCATCATAGTCAGCTCTCATCATGAAGCTGAGGAAAGCACATGTGATATGGAAAG
AGAGAAAAGCAAGTTTCAGAACACACACTAGAAAGTTACAGATCAAGGTCAAATAATGAA
GAAACATCATCTGAAGAGAAAAATGCCAATCTCCCACCCATGCGTTGATGAACATAC
ATCACAAAGTTGACTCAGAACGAAAAACAAAGAGGAAGGAAAATGTACAACATTCAAAA
TTAGAAGAAAAGCACATCCAAGTACCAAGAGAGTATTGTGTAGTGCTCTGCAATGGAAC
ATGTGATTTCAGGGTGCGCAGTGTACCTCAGTGGACCAGCCTGGGGAGGAGCTTA
ATTGCTGAGACATTAATAATGACCTCTTAGTGCATGCAAGATGGTGTCTCGGATAATG
ATCTGCCCCGGAGCTAGGGCAGCAACATGAGGACCAAACCATGACATAAAAGCTTGAGT
TTAAAAAAGAAAAGCAAAAAATAATTATGCCCTGACACTACTTCAGAGCAGGAGGATTCT
ACGAAGCCTTGGGATCAGGGTCAGTGTGAGCAGCTAACATCCTACCTCAAATGGAACAG
GATTTTTGATGCTTGTCTAATGGAACTGTTAAAAATTTTTTCTTTTAATAT
TTTCTCTGGTCACAAAATAAGAAATTGGGATGCAAAGTACCTAAAGATCTGATCC
TAAGAAGTTACTTCTGGCCAGGGCGGCTCATGCTGTAACTCTAGCAGTTGGAG
GCTGAGGTAGGCAGATCACTTGAGGTCAAGGAGTTGGAGAACAGCCTGGCCACACTG
AACCCCGTCTACTAAAAATGCAAAAATTAGCCAGGGTAGTGGTGCACCTGTAGTC
TCAGATACTTGGGAGGCTGAGGGTGGAGAACGCTTGAACCTGGGAGGTGGAGATTG
TGAGTCAGATCTCACCCTGAACTCCAGGCTGGCGACAGAGGGAGACTCTGCTCAAA

7

Copy Sequence



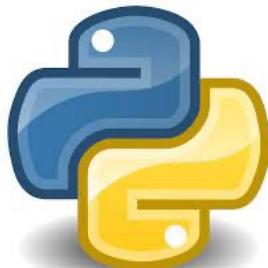
CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.



8

```
sequence = """ATGTGGTGGAGAGTTCTCAGCTGCTGCATGGTCCCTGCAAGAGGCCTCTGACT  
AACCACACAGAAACCATACCGTGGAGGAAGGCCAGACGCTCACTCTAAAGTGTGTCACT  
TCTCTGAGGAAGAACTCCTCCCTCAGTGGCTGACCCCTCAGGGTTACCAATTNTTA  
AATGAGTATCCTGCTTAAAAAATTCCAATACCAGCTTCTTCATCACTCGCCAATCAG  
CTCTCCATCACTGTGCCTAACGTAACCTGCAAGATGAAGGCGTGTACAAGTGCTTACAT  
TACAGCGACTCTGTAAGCACAAAGGAAGTGAAAGTGATTGTGCTGGCAACTCCTTCAAG  
CCAATCCTGGAAGCTTCAGTTATCAGAAAGCAAATGGAGAAGAACATGTTGACTCATG  
TGCTCCACCATGAGAAGCAAGCCCCCTCCGAGATAACCTGGCTACTTGGGAATAGCATG  
GAAGTGTCCGGTGGAACGCTCCATGAATTGAAACTGATGGGAAGAAATGTAATACTACC  
AGCACTCTCATAATCCACACTTATGGCAAAAATTCAACGGTGGACTGCATTATCCGACAC  
AGAGGCCTGCAAGGGAGAAAAGTAGTAGCACCCTCCGGTTGAAGATTGGTTACTGAT  
TTTCAGATGCTCTGGAGAGAAACTCTATCCTCTCAAGACCCACAGCAG  
TGTCTCAGTAACGGAAGATTCTAGTACATCGGAGATTGACAAGGAAGAG  
CACTCAAGATCCTGACTTGACCACCGAAGCAAATCCTCAGTATTTCAG  
GAAAAGTGGCATCCTGCTGCTCACGCTGGTGTCCCTCCTCATTT  
AGTCCAGCTTCATCATGAAGCTGAGGAAAGCACATGTGATATC  
AGTTTCAGAACACACACTAGAAAGTTACAGATCAAGGTCAAATAA  
TGAAAGAGAAAATGCCAATCTCCACCTATGCCTGCATGAA  
ATCACAAAGTTGACTCAGAACGAAAAACAAAGAGGAAGGAAAATGTACAACATT  
TTAGAAGAAAAGCACATCCAAGTACCGAGAGTATTGTGAGTGCTCTGCAATC  
ATGTGATTCAGGGTTGCCGCAGTGTACCTCAGTGGACCAGCCTGGGGAAAGGAC  
ATTGCTGAGACATTAATAATGACCTCTAGTGCAATGCAAGATGGTGTCTCGGA  
ATCTGCCCGGAGCTAGGGCAGCAACATGAGGACCAACCATGCACATAAGCTGTAGT  
TTAAAAAAAGAAAAGCAAAAAATAATTATGCCTGACACTACTTCAGAGCAGGAGGATTCT  
ACGAAGCCTGGGGATCAGGGTCAGTGTGAGCAGCTAACATCCTACCTCAAATGGAACAG
```

Assign sequence to variable as a string



Note: Use triple quotations to fit the sequence in a block of code

Using the functions that you have learned, get to know the CRTAM DNA sequence.

Answer the following questions **using complete sentences**:

- What is the length of the CRTAM DNA sequence?
- How many Adenines are in your sequence?

Answer



Using the functions that you have learned, get to know the CRTAM DNA sequence.

Answer the following questions **using complete sentences**:

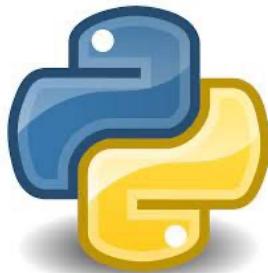
- What is the length of the CRTAM DNA sequence?
- How many Adenines are in your sequence?

```
[ ] length = len(sequence)
    print(f'The length of my DNA sequence is {length} nucleotides.')
```

The length of my DNA sequence is 2465 nucleotides.

```
[16] A_count = sequence.count('A')
    print(f'This sequence has {A_count} Adenines in total.')
```

This sequence has 772 Guanines in total.



Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$



Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

What are we missing?



Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

Return the total length of DNA sequence



Let's find the GC content of the CRTAM DNA sequence.

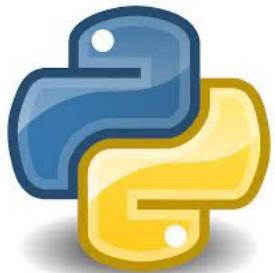
Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

```
[10] length = len(sequence)
      print(f'The length of my DNA sequence is {length} nucleotides.')
      The length of my DNA sequence is 2465 nucleotides.
```



Tip: Using string formatting eases mixing different datatypes together

Let's find the GC content of the CRTAM DNA sequence.

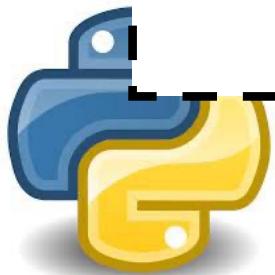
Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

Return total Guanines and Cytosines of the DNA sequence



Let's find the GC content of the CRTAM DNA sequence.

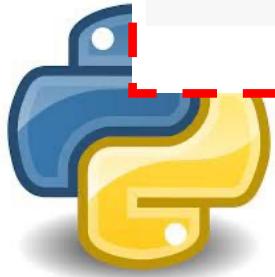
Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

```
[14] G_count = sequence.count('G')
     C_count = sequence.count('C')
     print(f'This sequence has {G_count} Guanines in total.')
     print(f'This sequence has {C_count} Cytosines in total.')
```



Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

Return GC Content



Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

```
[12] GC_content = (G_count + C_count)/length*100
      print(f'The GC content of my sequence is {GC_content}')
```

The GC content of my sequence is 42.799188640973625



Let's find the GC content of the CRTAM DNA sequence.

Overall Code

```
✓ 0s [13] length = len(sequence)
      print(f'The length of my DNA sequence is {length} nucleotides.')
```

The length of my DNA sequence is 2465 nucleotides.

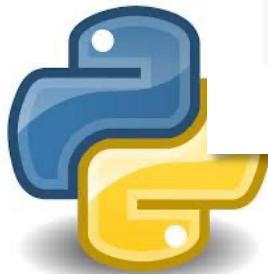
```
✓ 0s [14] G_count = sequence.count('G')
      C_count = sequence.count('C')
      print(f'This sequence has {G_count} Guanines in total.')
      print(f'This sequence has {C_count} Cytosines in total.')
```

This sequence has 537 Guanines in total.

This sequence has 518 Cytosines in total.

```
✓ 0s [15] GC_content = (G_count + C_count)/length*100
      print(f'The GC content of my sequence is {GC_content}')
```

The GC content of my sequence is 42.799188640973625





PennState
Huck Institutes of
the Life Sciences

Lesson 1.3.1

Define a function
that returns the GC
content of a DNA
sequence.



Define a function that calculates GC content

```
[11] def encontrar_GC_content(a_sequence):
    length = len(a_sequence)
    G_count = a_sequence.count('G')
    C_count = a_sequence.count('C')
    GC_content = (G_count + C_count)/length*100
    return(GC_content)
```



Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
```



In the first line, a function is initiated by using **def**



Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
```



Name the function
as you see fit



Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence)
```



Identify if the function takes **input**. Here, our function just needs one **input**. The function needed as **input** is a DNA sequence in order to return the GC content.

DON'T FORGET PARENTHESES!



Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
```



End it with a **colon**



Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):  
    length = len(sequence)  
    G_count = sequence.count('G')  
    C_count = sequence.count('C')  
    GC_content = (G_count + C_count)/length*100
```



**Body of the function
defined **dictates**
what the function is
doing**



Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):  
    length = len(a_sequence)  
    G_count = a_sequence.count('G')  
    C_count = a_sequence.count('C')  
    GC_content = (G_count + C_count)/length*100  
    return(GC_content)
```



Using the **return** function will give you the result of the function when you run it

**DON'T FORGET
PARENTHESES!**



Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
    length = len(a_sequence)
    G_count = a_sequence.count('G')
    C_count = a_sequence.count('C')
    GC_content = (G_count + C_count)/length*100
    return(GC_content)
```

```
[22] find_GC_content(sequence)
```

42.799188640973625

great way to
control code when
running a repeated
block of code more
than once on
different
datatypes!





PennState
Huck Institutes of
the Life Sciences

Lesson 1.4.0

Running into errors





Uh oh!

Ran into an error!



PennState
Huck Institutes of
the Life Sciences

```
[44] length = len(sequence)
    G_count = sequence.count('G')
    C_count = sequence.count('C')
    GC_content = (G_count + C_count)/length

    print("My sequence is "+str(length)+" nucleotides long.")
    print("There are "+str(G_count)+" guanines and "+C_count+" cytocines.")
    print("The GC content of my sequence is "+str(GC_content)+".")
```

My sequence is 2508 nucleotides long.

TypeError

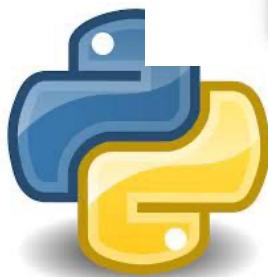
Traceback (most recent call last)

<ipython-input-44-b05e980949a1> in <module>

```
      5
      6 print("My sequence is "+str(length)+" nucleotides long.")
----> 7 print("There are "+str(G_count)+" guanines and "+C_count+" cytocines.")
      8 print("The GC content of my sequence is "+str(GC_content)+".")
```

TypeError: can only concatenate str (not "int") to str

SEARCH STACK OVERFLOW



Where is my **error?**



Debug and rerun program



PennState
Huck Institutes of
the Life Sciences

```
[44] length = len(sequence)
    G_count = sequence.count('G')
    C_count = sequence.count('C')
    GC_content = (G_count + C_count)/length

    print("My sequence is "+str(length)+" nucleotides long.")
    print("There are "+str(G_count)+" guanines and "+C_count+" cytocines.")
    print("The GC content of my sequence is "+str(GC_content)+".")
```

My sequence is 2508 nucleotides long.

TypeError

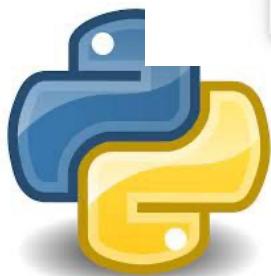
Traceback (most recent call last)

<ipython-input-44-b05e980949a1> in <module>

```
5
6 print("My sequence is "+str(length)+" nucleotides long.")
----> 7 print("There are "+str(G_count)+" guanines and "+C_count+" cytocines.")
8 print("The GC content of my sequence is "+str(GC_content)+".")
```

TypeError: can only concatenate str (not "int") to str

SEARCH STACK OVERFLOW



Where is my **error**?



Error Corrected

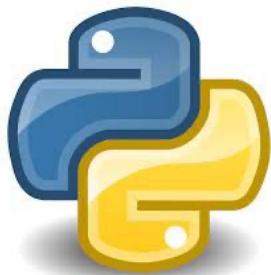


PennState
Huck Institutes of
the Life Sciences

```
[45] length = len(sequence)
G_count = sequence.count('G')
C_count = sequence.count('C')
GC_content = (G_count + C_count)/length

print("My sequence is "+str(length)+" nucleotides long.")
print("There are "+str(G_count)+" guanines and "+str(C_count)+" cytocines.")
print("The GC content of my sequence is "+str(GC_content)+".")
```

My sequence is 2508 nucleotides long.
There are 549 guanines and 528 cytocines.
The GC content of my sequence is 0.42942583732057416.

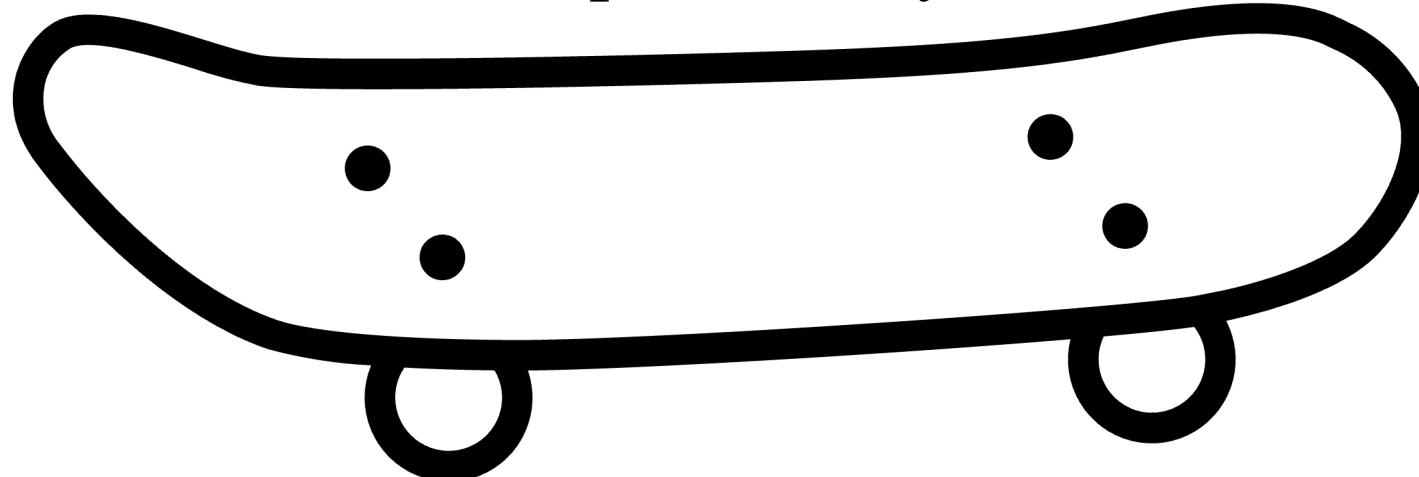




PennState
Huck Institutes of
the Life Sciences

Proficiency Assessment

Show off your new skills using Google COLAB with a proficiency assessment



Develop a block of code

- 1. Find a DNA sequence of your interest using Uniprot and cross referencing a database. Assign your sequence a variable.**

- 2. Define and return the following functions:**
 - 1. Total number of Guanines**
 - 2. Total number of Cytosines**
 - 3. Total number of Adenines**
 - 4. Total number of Uridines**
 - 5. A+T/G+C ratio**

