**How to Create a Lab Guide**

Table of Contents

[Create the Lab Guide 3](#_Toc47104918)

[Write Your Lab Guide Content in Word (or another program) 3](#_Toc47104919)

[Access the ‘My Content’ Tool in TILE Composer 3](#_Toc47104920)

[Copy Your Lab Guide Content into a New ‘Post’ 4](#_Toc47104921)

[Map the Lab Guide to the Lab Activity 8](#_Toc47104922)

[Media Tips 10](#_Toc47104923)

[Lab Guide Templates 12](#_Toc47104924)

[Template 1 – Simple Text 12](#_Toc47104925)

[Template 2 – Text w/Media 13](#_Toc47104926)

[Template 3 – Content Toggles 16](#_Toc47104927)

[Template 4 –Tabs w/Content Toggles 20](#_Toc47104928)

Lab Guides are used in Virtual Lab activities to provide instructions and reference material for the learners, providing them with the information and guidance they’ll need to complete the lab. You can create your own lab guide for a Virtual Lab activity using the “My Content” feature in TILE Composer. Follow the instructions below to build your own guide.

A screenshot of a computer

Description automatically generated

## Create the Lab Guide

### Write Your Lab Guide Content in Word (or another program)

1. Select one of these links to choose a “Starter Content” template for your lab guide, then use it to write and format your lab guide content:

* [Starter Template #1 – Simple Text](#_Template_1_–)
* [Starter Template #2 – Text w/Media](#_Template_2_–)
* [Starter Template #3 – Content Toggles](#_Template_3_–_1)
* [Starter Template #4 – Tabs w/Content Toggles](#_Template_4_–Tabs_1)

2. When you are finished writing your lab guide, select and copy all of the content to your device’s clipboard.

### Access the ‘My Content’ Tool in TILE Composer

1. Login to TILE (<https://dcm.toolwire.com/alai/portal/index.jsp>)

A screenshot of a cell phone

Description automatically generated

2. Access TILE Composer by selecting **Composer** at the top of the TILE interface. “Composer” is the Admin area of TILE.

A picture containing drawing

Description automatically generated

(NOTE: If you do not see the ‘Composer’ option, you may need to switch to a TILE profile that has administrative rights. Select the **Profiles** option in the side menu, then select the appropriate TILE profile.)

A picture containing drawing

Description automatically generated

3. TILE Composer displays. Next, select the **My Content** option in the side menu.A screenshot of a cell phone

Description automatically generated

4. The **My Content** interface displays. You can use this tool to create lab guides, as well as to upload images and other media. (NOTE: This tool is attached to your TILE user profile, which means the content you add here is *only visible to you* until you decide to share it with others.)

A screenshot of a cell phone

Description automatically generated

### Copy Your Lab Guide Content into a New ‘Post’

1. Click **+ New** at the top of the My Content interface, and then select the **Post** option.

A screenshot of a cell phone

Description automatically generated

2. The ‘Write Post’ interface displays. Enter a **Title** for your new lab guide.

A screenshot of a cell phone

Description automatically generated

3. Select the ‘block’ located below the lab guide title. Then **Paste** the lab guide content that you copied earlier in your Word (or other) program.

A picture containing bird, tree, flower

Description automatically generated

A screenshot of a cell phone

Description automatically generated

4. Select **Save Draft**.

A screenshot of a cell phone

Description automatically generated

5. Review the lab guide content you pasted into the post and make updates as needed:

* Grammar and typos
* Text style (bold, italics, etc.) and color

A screenshot of a cell phone

Description automatically generated

6. If you have any code snippets in your content, just select the text for the code and then select the **Inline Code** option in the content properties.

A screenshot of a cell phone

Description automatically generated

Alternatively, you can add a new block (by clicking the **+ button**, then selecting **Code**). Then copy/paste your code text into the new code field.

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

7. If you copy/pasted any images or other types of media, you will be prompted to upload or provide a link to these files. (Refer to the “Media Tips” section further down in this document for more guidance on adding media to your lab guide.)

A screenshot of a cell phone

Description automatically generated

8. Save the post again. Then select the **Preview** button to see what your lab guide looks like in a browser.

A screenshot of a cell phone

Description automatically generated

9. Continue to adjust your content and formatting, as needed, until the lab guide is finished. Then select the **Publish** button at the top of the post.

A close up of a sign

Description automatically generated

An “Are you ready to publish?” message appears. Leave the settings as they are and click **Publish** again.

A screenshot of text

Description automatically generated

10. A message displays, confirming that your new lab guide is now published and available. Select the **Copy Link** button to copy the URL for your new lab guide. You will use this link in the next section of instructions to “connect” your lab guide to the virtual lab guide activity.

A screenshot of a cell phone

Description automatically generated

## Map the Lab Guide to the Lab Activity

*Navigate to the Lab Resource in TILE Composer*

1. Select **Resources** in the side menu.

A screenshot of a cell phone

Description automatically generated

2. The Resources page opens. Use the breadcrumb links to make sure you are in the right course/skill. Then select the **Lab Resource** you want to attach your new lab guide to.

A screenshot of a cell phone

Description automatically generated

3. The Settings Panel for the lab resource opens. (Make sure to click the ‘Lock & Edit’ button at the top of the Composer interface, if you haven’t already.) ***Paste the link you copied for the lab guide post in My Content*** into the **URL/Filename** field. Then click **Save**.

A screenshot of a cell phone

Description automatically generated

A picture containing drawing

Description automatically generated

4. Your lab guide is now attached to the Virtual Lab activity. To make sure the lab guide is displaying properly, select **Portal** at the top of the screen, then navigate to the Virtual Lab and launch it.

A close up of a sign

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## Media Tips

*1. Upload an Image to a Placeholder Copy/Pasted from Word*

Use the **Upload**, **Media Library**, or **Insert from URL** options and follow the resulting prompts:

A screenshot of a cell phone

Description automatically generated

*2. Add a New Media Element to Lab Guide in My Content*

a. Add a new ‘Block’ to the post by clicking the **+add** option at the top of the interface, or in-line with the lab guide content:

*A picture containing bird

Description automatically generated*

b. The ‘Block Settings’ panel opens. Select the **Common Blocks** category, then the media type you want to add. Then follow the resulting prompts to **upload** the media file or **insert from URL**.

*A screenshot of a cell phone

Description automatically generated*

c. Alternatively, you can also select the **Embeds** category to embed media from popular web-based sources.

*A screenshot of a cell phone

Description automatically generated*

## Adding Content Toggles and Tabs

If you have a lot of content for your lab guide, you may want to use Content Toggles and/or Tabs to make it easier for the learner to navigate. (NOTE: These elements are included in Lab Guide Templates 3 and 4.)

*Add a Content Toggle*

1. Select the **+Add block** button as you would for any other lab guide element.

*A picture containing bird, flower

Description automatically generated*

2. Type ‘content’ into the search box, then select the **Content Toggle** option.

*A screenshot of a cell phone

Description automatically generated*

3. A content panel is added. Add a title to the new panel heading. (When the user clicks this heading, the panel will expand/collapse.)

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

4. You can customize the color of the panel container using the control menu on the right side of the screen.

A screenshot of a social media post

Description automatically generated

5. Next, write/copy in the content for the panel. To add a new panel below the current one, just click the **+** button at the bottom of the panel. (You can click the **+** button at the top to add a new panel above.)

A screenshot of a social media post

Description automatically generated

6. You can control the expand/collapse behavior of the panels by changing the settings in the **Initial State** panel at the right side of the screen.

A screenshot of a cell phone

Description automatically generated

*Add a Tab*

1. Select the **+Add block** button as you would for any other lab guide element. Then type ‘tabbed’ into the search box, then select the **Tabbed Content** option.

*A screenshot of a cell phone

Description automatically generated*

2. The new Tab element is added. Click directly on a tab to give it a title. Click the **+** tab button to add a new tab.

*A screenshot of a cell phone

Description automatically generated*

*A screenshot of a cell phone

Description automatically generated*

3. Use the **Tab Colors** controls at the right side of the screen to adjust the color of the tabs.

*A screenshot of a cell phone

Description automatically generated*

4. To add content inside of a tab, just click the **+Add block** button as you normally would. You can add any type of content within a tab, including a Content Toggle (or even another Tab element).

A close up of a logo

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## Lab Guide Templates

### Template 1 – Simple Text

*Instructions:* Update the content below with the content for your lab. (You can write the content directly in this Word document, or write it in a separate Word file or another program.) When you are finished writing your lab guide, select and copy all of the content to your device’s clipboard.

Heading

Overview text…

Follow these step-by-step instructions:

1. Instructions for step 1...
2. Instructions for step 2…

* Detailed instructions about this step…
* More detailed instructions about this step…

1. Instructions for step 3…

NOTE: Special information about this step that needs to be emphasized…

1. Instructions for step 4…
2. Instructions for final step of the lab guide…

[[Go back up to Lab Guide Instructions](#__Write_Your_Lab)]

**Sample Content for this Lab Guide Type**

Here’s some sample content for this type of Lab Guide:

Linux Command Line

Follow these lab instructions:

1. Start the Praxis Cloud Desktop to access the Linux Operating System (OS).
2. Click on your home directory in the Graphical User Interface (GUI).

* What directories are inside this directory?
* Look inside the directories. What files are in there? Don’t be shy.

1. Open the terminal so you can access the Linux Command Line Interface (CLI).
2. Instead of clicking icons representing files and apps in a GUI, one types text commands in the CLI. This may be awkward at first, but soon you will see the power in the CLI’s flexibility. Type some Linux commands in the CLI.

NOTE: Linux is case sensitive so uppercase and lowercase letters are different.

1. **Is ->** This command is a Linux program (aka app or binary) that lists files and directories. **Note:** In Linux, you usually start in your home directory named after your username. For example ‘\home\student’ is the home directory for the user ‘student’.
2. **pwd ->** This command shows you at the present working directory. Note that the root of the filesystem begins with ‘\’ so one can ‘cd\’.

NOTE: When we show single quotes around a command line, do not type the quotes in the CLI.

### Template 2 – Text w/Media

*Instructions:* Update the content below with the content for your lab. (You can write the content directly in this Word document, or write it in a separate Word file or another program.) When you are finished writing your lab guide, select and copy all of the content to your device’s clipboard.

Heading

Overview text…

**Step-by-step instructions:**

1. Instruction for step 1, including an image...

A close up of a logo

Description automatically generated

1. Instruction for step 2…

* Detailed instructions about this step…
* More detailed instructions about this step…

1. Instruction for step 3…

NOTE: Special information about this step that needs to be emphasized…

1. Instruction for step 4, including a reference table…

|  |  |  |
| --- | --- | --- |
| Heading 1 | Heading 2 | Heading 3 |
| Cell 1 text… | Cell 2 text… | Cell 3 text… |

1. Instruction for step 5, including a code snippet…
2. Instruction for step 6, including a video…

[video filename or link to video hosted online]

1. Instruction for final step of the lab guide…

[[Go back up to Lab Guide Instructions](#__Write_Your_Lab)]

**Sample Content for this Lab Guide Type**

Here’s some sample content for this type of Lab Guide:

Remote Data Monitoring

**Overview**

Due to the drawbacks of the manual tracking of devices, we need to design a cloud-based system which can perform below tasks automatically:

* Collect device data (Temperature, Humidity and Door Status)
* Process it for threshold violation and generate alert records if required
* Save telemetry data and alerts data in cloud database
* Expose telemetry data for external system usage
* Create a dashboard web app to visualize telemetry and alerts data

Complete the Requirements in order.

**Problem Description:**

1. You need to create a cloud-based system which will keep track of various parameters of the devices. The parameters are Temperature, Humidity and Door Status. These parameters should be sent by each device to the cloud after a specific interval (ex. after each 20 seconds). On cloud, process only temperature data against predefined thresholds from blob storage.
2. Rest of the parameters data should be directly added to Table Storage without threshold checking. Alert records should be generated if any threshold is violated for temperature value. All the telemetry & alerts data should be saved in table storage.

**Design Consideration:**

A screenshot of a cell phone

Description automatically generated

**Code Skeleton:**

*Device simulator will send following JSON to the cloud:*

{

"DeviceId": "<GUID>",

"CaptureTime": "<EPOCH/Unix Time>",

"DeviceLat": "<Device latitude>", //Range from -90 to 90

"DeviceLong": "<Device longitude>", //Range from - 180 to 180

"Temperature": "<Device Temperature>", //Range from -15 to 10

"TemperatureUnit": "<Unit>", //Celsius

"Humidity": "<Device humidity>", //Range from 50% to 80%

"HumidityUnit": "<Device humidity>", //%

"DoorStatus": "<Device door status (0/1)>" //Possible values true/false

}

*Actual simulator output json:*

{

"TemperatureUnit":"Celsius",

"HumidityUnit":"%",

"DeviceId":"d4aea4d2-2c97-4e2c-9d12-036e9a0c52c0",

"CaptureTime":1522250154,

"DeviceLat":0.0,

"DeviceLong":0.0,

"Temperature":4.0248263856,

"Humidity":52.0,

"DoorStatus":false

}

### Template 3 – Content Toggles

*Instructions:* This template includes ‘content toggles,’ which are expandable/collapsible sections you can organize your content into. Update the content below with the content for your lab. (You can write the content directly in this Word document, or write it in a separate Word file or another program.) When you are finished writing your lab guide, select and copy the content – section by section – to your device’s clipboard.

Heading

Toggle 1 (expanded by default)

**Subheading A**

Content…

Follow these step-by-step instructions:

1. Instructions for step 1...
2. Instructions for step 2…

Toggle 2

**Subheading B**

Content…

Follow these step-by-step instructions:

1. Instructions for step 1...
2. Instructions for step 2…

Toggle 3

**Subheading C**

Content…

Follow these step-by-step instructions:

1. Instructions for step 1...
2. Instructions for step 2…

[[Go back up to Lab Guide Instructions](#__Write_Your_Lab)]

**Sample Content for this Lab Guide Type**

Here’s some sample content for this type of Lab Guide:

Coronavirus Genomics Lab

Toggle 1 (expanded by default)

**Overview**

Genomes are DNA (or RNA) sequences that encode the instructions (genes) required to make an organism. Humans, aardvarks, clovers, bacteria, viruses … everything alive … has a genome. In the 21st century, emerging pathogen genomes can now be fully sequenced and assembled in a few days.

Once we know the pathogen’s genome sequence, we can develop diagnostics tests, therapeutics, vaccines, and study the pathogen’s biology. Never in history has this biotechnology been so rapid! For example, here is a viral pathogen called the coronavirus that was sequenced in 2019:

A picture containing cake, decorated, fruit, table

Description automatically generated

Toggle 2

**Task A**

**Obtain the coronavirus genome sequence in two formats: FASTA and GENBANK.**  
*Step 1:* Download the coronavirus genome sequence in FASTA format from NCBI ([MN908947.3](https://www.ncbi.nlm.nih.gov/nuccore/MN908947.3?report=fasta)) and store in a text file called MN908947.fasta. *Step 2:* Obtain the coronavirus genome sequence in GENBANK format from NCBI ([MN908947.3](https://www.ncbi.nlm.nih.gov/nuccore/MN908947.3?report=genbank)) by copying the GENBANK sequence starting with the word ‘LOCUS’ and ending with the double slash ‘//’ and pasting into a text file called *MN908947.genbank*.

Toggle 3

**Task B**

**Cut out the all of the coronavirus gene subsequences from the full genome and save in a file.**  
*Step 1:* In a Jupyter notebook, write Python code that reads the GENBANK formatted coronavirus genome DNA sequence file from task A, parses the coordinates of all CDS gene subsequences, and stores those values in a Python dictionary.

***Pseudocode:***

Open and load the GENBANK-format genome file

Loop through each line of the file {

For each gene {

Extract the gene name and CDS coordinates and store in a dictionary where the key is the gene\_name and the values are the CDS coordinates

}***DataByte:***

For example, here is the first gene ‘orf1ab’ from the GENBANK file with two sets CDS coordinates (266..13468,13468..21555). CDS coordinates might have a gap so two subsequences would need to be joined to get the full CDS subsequence:

**gene 266..21555**

**/gene=”orf1ab”**

**CDS join(266..13468,13468..21555)**

}*Step 2:*In a Jupyter notebook, write Python code that reads the FASTA formatted coronavirus genome DNA sequence file from task A, cuts out the actual sequences from the FASTA file using the dictionary from Task B (Step 1), and writes a text file called that contains with all of the coronavirus CDS DNA sequences in a single concatenated DNA file.

***Pseudocode:***

Open and load the FASTA-format genome file

Loop through all of the genes in your coordinate dictionary from step 1{

Write the gene name to an output file starting with the ‘>’ FASTA delimiter

Cut the CDS DNA subsequence of the genome from the FASTA file (join CDS substrings if necessary) and append to the output file

}***Notes:***

1. All the gene sequences are now in the same file that can be parsed using lines that start with the FASTA ‘>’ delimiter.
2. A string of triplet codons in RNA is called the protein codon sequence (CDS) – See BioBytes. If you look in the GENBANK file above the actual sequence
3. The CDS DNA sequences are in the GenBank file, so you can check the accuracy of your sequence extraction.

Toggle 4

**Task C**

**Transcribe all of the genes from DNA into RNA.**

In a Jupyter notebook, write Python code that transcribes all DNA CDS sequences into single concatenated RNA sequence file in FASTA format.

***Pseudocode:***

Open and load the CDS FASTA DNA file from Task B

For each sequence {

Write the line that starts with ‘>’ to an output file

Use a dictionary to convert all ‘T’ bases to ‘U’ and make all bases uppercase

Write the transcribed RNA sequence to the output file

}***DataByte:***

For example, the beginning of the first gene ‘orf1ab’ would be transcribed from

**>orf1lab (DNA)**

**ATGGAGAGCCTTGTC…***into*

**>orf1lab (RNA)**

**AUGGAGAGCCUUGUC…**

Toggle 5

**Task D**

**Translate all of the genes from RNA into PROTEIN.**

In a Jupyter notebook, write Python code that translates all DNA CDS sequences into single concatenated PROTEIN sequence file in FASTA format.

***Pseudocode:***

Open and load the FASTA RNA file from Task C

For each sequence {

Write the line that starts with ‘> GeneName’ to an output file

Use a Python dictionary to translate all three character codons to amino acid codes

Write the transcribed RNA sequence to the output file

}***DataByte:***

For example, the beginning of the first gene ‘orf1ab’ would be translated from

**>orf1lab (RNA)**

**AUGGAGAGCCUUGUC…***into*

**>orf1lab (PROTEIN)**

**MESLV…*Notes:***

1. The protein sequences are in the GenBank file, so you can check the accuracy of your sequence extraction.
2. Check out the CodeBytes section for help with the genetic code dictionary.
3. Stop codons do not code for an amino acid. The penultimate codon would be the last amino acid.

### Template 4 –Tabs w/Content Toggles

*Instructions:* This template includes ‘content toggles’ and tabs - which provide another level of navigation for lab guides with in-depth content. Update the content below with the content for your lab. (You can write the content directly in this Word document, or write it in a separate Word file or another program.) When you are finished writing your lab guide, select and copy the content – section by section – to your device’s clipboard.

Heading

Tab 1 (selected by default)

Tab Label A

Toggle 1 (expanded by default)

**Subheading A**

Content…

Follow these step-by-step instructions:

1. Instructions for step 1...
2. Instructions for step 2…

Toggle 2

**Subheading B**

Content…

Follow these step-by-step instructions:

1. Instructions for step 1...
2. Instructions for step 2…

Tab 2

Tab Label B

Toggle 1 (expanded by default)

**Subheading A**

Content…

Follow these step-by-step instructions:

1. Instructions for step 1...
2. Instructions for step 2…

Toggle 2

**Subheading B**

Content…

Follow these step-by-step instructions:

1. Instructions for step 1...
2. Instructions for step 2…

[[Go back up to Lab Guide Instructions](#__Write_Your_Lab)]

**Sample Content for this Lab Guide Type**

Here’s some sample content for this type of Lab Guide:

Coronavirus Genomics Lab

Tab 1 (selected by default)

Overview

Toggle 1 (expanded by default)

**Coronavirus Genomics Lab Overview**

**Background:** Genomes are DNA (or RNA) sequences that encode the instructions (genes) required to make an organism. Humans, aardvarks, clovers, bacteria, viruses … everything alive … has a genome. In the 21st century, emerging pathogen genomes can now be fully sequenced and assembled in a few days.

Once we know the pathogen’s genome sequence, we can develop diagnostics tests, therapeutics, vaccines, and study the pathogen’s biology. Never in history has this biotechnology been so rapid! For example, here is a viral pathogen called the coronavirus that was sequenced in 2019:

A picture containing cake, decorated, fruit, table

Description automatically generated

Toggle 2

**SuperLab Objectives**

In this lab, you will download a coronavirus genome and write Python code that:

1. Extracts the coronavirus genes from the full genome file as DNA sequences
2. Transcribes the DNA sequences into RNA sequences
3. Translates the RNA sequences into protein sequences

Tab 2

CodeBytes

Toggle 1 (expanded by default)

**What is a CodeByte?**

Here are useful Python code pieces that will help you in this lab. Remember that programming is a language are there are many code paths to achieve the same goal so try out other solutions!

Toggle 2

**Open and Read a File in Python**

**CodeByte: How to open a file, read all the lines, and close a file a Linux filesystem:**  
infile = open(“/home/student/directory/filename”, “r”)  
lines = infile.readlines()  
infile.close()

Toggle 3

**Open and Write to a File in Python**

**CodeByte: How to open a file, write data to the file, and close a file a Linux filesystem:**  
outfile = open(“/home/student/directory/filename”, “, “w+”)  
for line in lines:

outfile.write(line)outfile.close()

Toggle 4

**Remove a Newline Character**

**CodeByte: Text files end in a newline character that is often not displayed. One can remove a newline character from a line of like this:**  
line = line.rstrip(‘\n’)

Toggle 5

**Python Single-Letter Replacement Dictionary**

**CodeByte: Transcription dictionary (Plus change lowercase to UPPERCASE):**  
dna2rna = {“A”:”A”, “T”:”U”, “G”:”G”, “C”:”C”, “N”:”N”, “a”:”A”, “t”:”U”, “g”:”G”, “c”:”C”, “n”:”N”}

Toggle 6

**Python Multiple-Letter Replacement Dictionary**

**Codebyte: The standard genetic code in a Python dictionary:**  
Below is the full genetic code in the form of a Python dictionary. Note that some amino acids are coded by multiple codons. Also, ‘AUG’ typically is the first codon in an RNA so most proteins begin with M as the first amino acid. Three other codons do not code for amino acids but signal the end of translation. These codons are called stop codons and include ‘UGA’, ‘UAA’, and ‘UAG’.

rna2protein = {“UUU”:”F”, “UUC”:”F”, “UUA”:”L”, “UUG”:”L”,

“UCU”:”S”, “UCC”:”S”, “UCA”:”S”, “UCG”:”S”,

“UAU”:”Y”, “UAC”:”Y”, “UAA”:””, “UAG”:””,

“UGU”:”C”, “UGC”:”C”, “UGA”:””, “UGG”:”W”,

“CUU”:”L”, “CUC”:”L”, “CUA”:”L”, “CUG”:”L”,

“CCU”:”P”, “CCC”:”P”, “CCA”:”P”, “CCG”:”P”,

“CAU”:”H”, “CAC”:”H”, “CAA”:”Q”, “CAG”:”Q”,

“CGU”:”R”, “CGC”:”R”, “CGA”:”R”, “CGG”:”R”,

“AUU”:”I”, “AUC”:”I”, “AUA”:”I”, “AUG”:”M”,

“ACU”:”T”, “ACC”:”T”, “ACA”:”T”, “ACG”:”T”,

“AAU”:”N”, “AAC”:”N”, “AAA”:”K”, “AAG”:”K”,

“AGU”:”S”, “AGC”:”S”, “AGA”:”R”, “AGG”:”R”,

“GUU”:”V”, “GUC”:”V”, “GUA”:”V”, “GUG”:”V”,

“GCU”:”A”, “GCC”:”A”, “GCA”:”A”, “GCG”:”A”,

“GAU”:”D”, “GAC”:”D”, “GAA”:”E”, “GAG”:”E”,

“GGU”:”G”, “GGC”:”G”, “GGA”:”G”, “GGG”:”G”}

Tab 3

Lab

Toggle 1 (expanded by default)

**Task A**

**Obtain the coronavirus genome sequence in two formats: FASTA and GENBANK.**  
*Step 1:* DDownload the coronavirus genome sequence in FASTA format from NCBI ([MN908947.3](https://www.ncbi.nlm.nih.gov/nuccore/MN908947.3?report=fasta)) and store in a text file called MN908947.fasta. *Step 2:* Obtain the coronavirus genome sequence in GENBANK format from NCBI ([MN908947.3](https://www.ncbi.nlm.nih.gov/nuccore/MN908947.3?report=genbank)) by copying the GENBANK sequence starting with the word ‘LOCUS’ and ending with the double slash ‘//’ and pasting into a text file called *MN908947.genbank*.

Toggle 2

**Task B**

**Cut out the all of the coronavirus gene subsequences from the full genome and save in a file.**  
*Step 1:* In a Jupyter notebook, write Python code that reads the GENBANK formatted coronavirus genome DNA sequence file from task A, parses the coordinates of all CDS gene subsequences, and stores those values in a Python dictionary.

***Pseudocode:***

Open and load the GENBANK-format genome file

Loop through each line of the file {

For each gene {

Extract the gene name and CDS coordinates and store in a dictionary where the key is the gene\_name and the values are the CDS coordinates

}***DataByte:***

For example, here is the first gene ‘orf1ab’ from the GENBANK file with two sets CDS coordinates (266..13468,13468..21555). CDS coordinates might have a gap so two subsequences would need to be joined to get the full CDS subsequence:

**gene 266..21555**

**/gene=”orf1ab”**

**CDS join(266..13468,13468..21555)**

}*Step 2:*In a Jupyter notebook, write Python code that reads the FASTA formatted coronavirus genome DNA sequence file from task A, cuts out the actual sequences from the FASTA file using the dictionary from Task B (Step 1), and writes a text file called that contains with all of the coronavirus CDS DNA sequences in a single concatenated DNA file.

***Pseudocode:***

Open and load the FASTA-format genome file

Loop through all of the genes in your coordinate dictionary from step 1{

Write the gene name to an output file starting with the ‘>’ FASTA delimiter

Cut the CDS DNA subsequence of the genome from the FASTA file (join CDS substrings if necessary) and append to the output file

}***Notes:***

1. All the gene sequences are now in the same file that can be parsed using lines that start with the FASTA ‘>’ delimiter.
2. A string of triplet codons in RNA is called the protein codon sequence (CDS) – See BioBytes. If you look in the GENBANK file above the actual sequence
3. The CDS DNA sequences are in the GenBank file, so you can check the accuracy of your sequence extraction.

Toggle 3

**Task C**

**Transcribe all of the genes from DNA into RNA.**

In a Jupyter notebook, write Python code that transcribes all DNA CDS sequences into single concatenated RNA sequence file in FASTA format.

***Pseudocode:***

Open and load the CDS FASTA DNA file from Task B

For each sequence {

Write the line that starts with ‘>’ to an output file

Use a dictionary to convert all ‘T’ bases to ‘U’ and make all bases uppercase

Write the transcribed RNA sequence to the output file

}***DataByte:***

For example, the beginning of the first gene ‘orf1ab’ would be transcribed from

**>orf1lab (DNA)**

**ATGGAGAGCCTTGTC…***into*

**>orf1lab (RNA)**

**AUGGAGAGCCUUGUC…**

Toggle 4

**Task D**

**Translate all of the genes from RNA into PROTEIN.**

In a Jupyter notebook, write Python code that translates all DNA CDS sequences into single concatenated PROTEIN sequence file in FASTA format.

***Pseudocode:***

Open and load the FASTA RNA file from Task C

For each sequence {

Write the line that starts with ‘> GeneName’ to an output file

Use a Python dictionary to translate all three character codons to amino acid codes

Write the transcribed RNA sequence to the output file

}***DataByte:***

For example, the beginning of the first gene ‘orf1ab’ would be translated from

**>orf1lab (RNA)**

**AUGGAGAGCCUUGUC…***into*

**>orf1lab (PROTEIN)**

**MESLV…*Notes:***

1. The protein sequences are in the GenBank file, so you can check the accuracy of your sequence extraction.
2. Check out the CodeBytes section for help with the genetic code dictionary.
3. Stop codons do not code for an amino acid. The penultimate codon would be the last amino acid.