Tutorial Template (working outline)

Code Tutorial

## Project Short Title | NASA DEVELOP | Term Year

## Project Long Title

**Tutorial**

Author 1 (Project Lead)

Author 2

Author 3

Author 4

***Advisors:***

Advisor 1, Affiliation (Science Advisor)

Advisor 2, Affiliation (Science Advisor)

***Previous Contributors:***

Contributor 1

Contributor 2

***Fellow:***  
Node Fellow (Node Name)

### Table of Contents

[Overview](#Bookmark1)

[Set up and Requirements](#Setup_Requirements)

*1. Program 01 installation/set up/usage notes*

*2. Program 02 installation/set up/ usage notes*

[Methods](#Methods)

1. Method Section

*1.1 Method subsection*

*1.2 Method subsection*

2. Method Section

*2.1 Method subsection*

*2.2 Method subsection*

[Conclusion](#Conclusion)

[Acknowledgements](#Acknowledgements)

[Citations](#Citations)

[Licensing](#Licensing)

[Examples](#_Examples)

**Section Ideas**

1. **Overview**

Brief paragraph explaining goal of the tutorial.

1. **Setup & Requirements**

Basic requirements and setup to get started. For instance, in GEE, explaining that your Gmail account needs to be registered if you are a first time GEE user, etc. Should be subdivided by programs/applications to include all that are utilized for the creation of the final product.

**Methods**

Here is where the majority of this deliverable lives. This should be an ordered list of instructions, e.g., actionable steps, necessary for replicating the methods that created the handoff material, and/or to perform a desired analysis. See [examples](#_Examples) of tutorial steps and code snippets at the end of this document.

**Brief Conclusion**

Suggestions on next steps if user finishes tutorial and may want to explore further. Conclusion should reiterate the overarching processes contained in the document and how they are applied to decision-making practices. Be careful here **not** to tell partners how to make decisions that are out of bounds for NASA. Try to focus on how the type of information garnered from this tutorial is of value to the organization. Word count limit for future work may be best. Not sure how long... 200-500 words?

1. **Acknowledgements**

All necessary acknowledgements of software, code, licensing, and contracting.

1. **Citations**

Maybe needs to correspond in-text, but a place for participants to put open-source code.

1. **Licensing**

Include licensing information as part of our dos/don'ts for code.

Keep these very broad so that teams can tailor organization to their specific project and specific tutorial type (i.e., code vs GIS)

**Tutorial Checklist**

**Dos and Don'ts for Including Code/Tutorial Best Practices**

Code tutorials explain how use to code or scripting tools that are already in the public domain. The idea here is that if the code is already in public, teams can quote that in snippets and interlace those snippets with narrative to create a hybrid kind of document that might help get partners closer to being able to code their own solutions. See [examples](#_Examples) at the end of this document.

|  |  |
| --- | --- |
| **DONT** | **DO** |
| **Don’t** include unreleased code. No code produced by the team can be shared outside of the DEVELOP program, including through tutorials. | **Do** include open-source code. There are two types of publicly released code to recognize:  (1) open-source code with a license  (2) public code without a license.  Teams should use open-source code with a license. |
| **Don’t** include images, screenshots, or direct examples of modified open-source code. This includes changing input parameters or study area directly in the code. | **Do** *describe* any changes or modifications to code in narrative form, such as changes in input parameters and study area. |
| **Don’t** describe the methods how you would in the tech paper. | **Do** write steps with an instructional tone. |
| **Don’t** combine or “mash up” various lines of code into one code chunk. Each code chunk or individual line of code should come from one source. | **Do** cite code included in the tutorial. Any code in the tutorial needs to be referenced clearly indicating the origin of the code. |

**Formatting**

1. We may need to include steps per page limit. That way some teams won’t cram info on pages while others put one step per page. Consistency here will be huge.
2. Strongly recommend use of **images/diagrams** to help illustrate steps better. Can include arrows, boxes, etc.
3. Maybe for GIS tutorials only, but if large images aren’t used, having icons in-text showing buttons the team instruct the user to click or select could be very helpful rather than the user having to hover over dozens of icons in ArcGIS Pro or Map for instance for each step.
4. Same font requirements (**Century Gothic** or **Garamond**) for non-code text.
5. Highlight key steps/words by app area color and in **bold font**. From the fall tutorials, lots of teams would put key words in quotes, but that doesn’t stand out to well, and was used inconsistently.
6. TBD – code snippets. Should they be screenshots? Can we format code in-text easily almost like a markdown file.
7. Paragraphs are okay to introduce sections, but all steps should be in numbered format. Complete sentences are not required in bullet points but are necessary for paragraphs.

**Content**

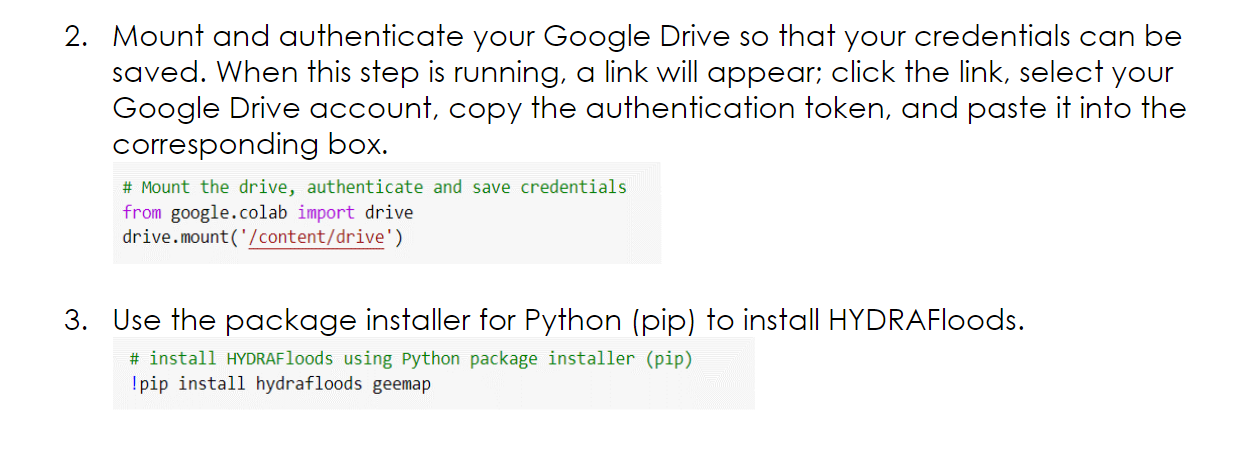
1. **Table of Contents is** **required**. Here are the guidelines for how to use the Table of Contents. See [examples](#_Examples) at the end of this document.
   1. Make sure to use the correct headings styles throughout the document. The styles are: Body Text, Heading 1, Heading 2, and Heading 3.

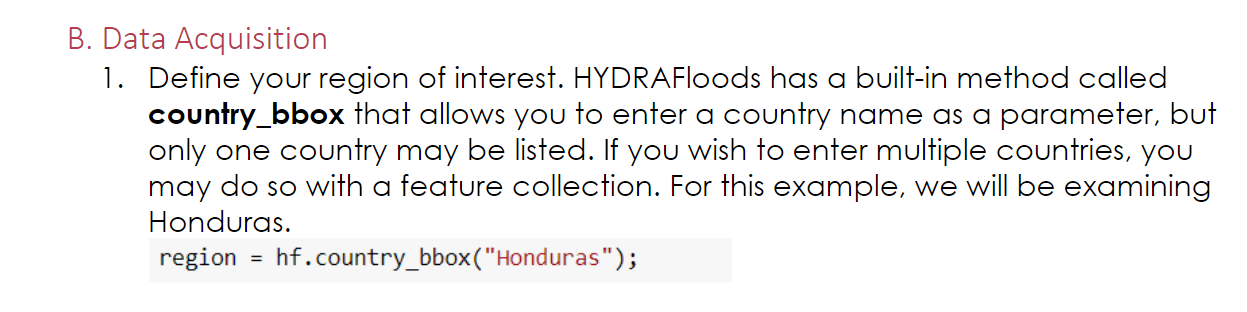
* **Body Text** – Used for most of the text throughout the document. Format when necessary (resize, bold, underline, etc)
* **Heading 1** – Used for the Main headings (Overview, Set Up & Requirements, Method, etc)
* **Heading 2** – Used for Section headings (Program, Data Acquisition, etc)
* **Heading 3** – Used for Sub Section headings (Study Area Shapefile, etc)
* To add/change a heading, highlight the text, go to the **Home** tab, go to **Styles**, and choose the one that you need.
  1. Once you’ve added new content to your document, you’re then able to Update the Table of Contents.
     + 1. Right click anywhere within the Table of Contents and select “**Update Field**”
       2. Click “**Update entire table**” and click “**Ok**”
       3. Everything is now updated, including the page numbers

**NOTE**: DO NOT edit the Table of Contents manually. As long as the right headings are used, it should appear correctly when updated.

1. All websites, software packages, etc., should be properly introduced at the first mention with how to access it (including links)
2. **Don’t assume your audience knows everything** – be detailed yet concise!
   1. Tailor this to the partner skill level as much as possible. Are they GIS experts or are they volunteers who have never done mapping/GEE before?
3. Include actionable steps – background info is better located in the tech paper. Keep this deliverable goal oriented.
4. Section at the beginning of tutorial describing purpose of document, how and why should be used, etc.
5. Brief conclusion section detailing how the methods described are applicable to decision making.

## Examples

**Tutorial Steps and Code Snippets**



**Table of Contents**

Graphical user interface, application

Description automatically generatedText

Description automatically generated

**Heading Styles**

**Examples of Headings being used**

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**Select “Update entire table” and “Ok”**

**Right-Click and select “Update Field”**