Python Development Tech Doc

* [Headings, Subheadings, and Table of Contents (Microsoft Word)](https://guides.lib.uni.edu/c.php?g=1243498&p=9207893)

|  |
| --- |
| Contents  [General Resources 2](#_Toc211252261)  [Python Intro 2](#_Toc211252262)  [Variables 2](#_Toc211252263)  [Concatenation and F-strings. 3](#_Toc211252264)  [Python Language and Syntax 3](#_Toc211252265)  [Python Lab 10/13/25 3](#_Toc211252266)  [Python Operators 4](#_Toc211252267)  [Version Control 4](#_Toc211252268)  [Web Accessibility and Inclusive Design 5](#_Toc211252269) |

Make your own technical documentation to help you learn and have resources to help you on your projects.

|  |
| --- |
| **DOs and DON’Ts** |
| DO: Provide clear and simple explanations in your own words (be concise)   * Use short sentences, small paragraphs * Use lists and tables to organize information * Use standard industry terms but explain in simple language   **DON’T copy my explanations from the slides as yours.** |
| DO Use Code Snippets and Visuals to reduce lengthy explanations:   * Include code examples you created * Include comments to describe parts of your code * Include Screenshots, GIFs, Images   **DON’T use my slides as your image**  **DON’t copy my code examples** |
| DO Include Resources that give steps or help explain concepts   * You can include resources given in the lecture or find your own but   + Include website link with summary   + Include links to videos with summary of content |
| DO Use AI to help you learn   * Ask questions to understand concepts better, give examples for it to explain in simple terms, ask for AI to provide different examples * Ask AI for resources so you can cite and provide AI Tool Name and include the URLs to the main resources that helped you * Summarize in your own words and include * Use AI to create images that you describe   **DON’t**  use AI to create your technical documentation. |

# General Resources

Here you can list general resources that you use frequently for this module. Add other general resources to quickly get help on topics in this module.

* [Class lectures](https://docs.google.com/presentation/d/1kMf_h0IPAY0Fo-jnHCoXOBztffeUHAEPjRv-BpURJO8/edit)
* [Runestone Academy Python Book](https://runestone.academy/ns/books/published/thinkcspy/index.html?mode=browsing)
* [Add a resource that helps you with Bootstrap]
* [Add a resource that helps you with Git]
* [Add a resource that helps you with GitHub]

You will also add resources to more detailed examples in each section below.

# Python Intro

## Variables

Variables can come in multiple forms. String and Integer are 2 very common types.

Strings: Strings are variables that are comprised of text/anything that is not a number

|  |  |
| --- | --- |
| Strings  Strings are variables that are composed of text. | Integers  Integers are variables that are composed of whole numbers. |
| A black screen with text  AI-generated content may be incorrect. | A screen shot of a computer program  AI-generated content may be incorrect. |

### Concatenation and F-strings.

F-strings are a output format that allows you to attach strings of output with integers and floats.

### Python Language and Syntax

|  |  |
| --- | --- |
| Keyword: A reserved word with special meaning. EX: if, else, while, for, def, class, True, False, import, return | Syntax: Syntax refers to the rules that define how code must be written for the Python interpreter to understand it.  Unlike many other languages, Python relies on indentation (spacing) instead of braces {} to group statements. |
| Punctuation: Python uses punctuation characters like :, (), [], ,, and . as structure markers, not as line terminators.   * : indicates a block of code (e.g., after if, def, or for). * () are used in function calls and grouping expressions. * [] are used for lists or indexing. * , separates arguments or items. | Statements: A statement is a complete instruction that performs an action.  In Python, each line of code is typically one statement.  Statements execute in order, top to bottom, unless modified by loops or conditionals. |
|  |  |

### Python Lab 10/13/25

Create a age calculator in python that also outputs the users favorite hobby.

|  |
| --- |
| #Simple Python program to calculate age based on user input  #Variable for user name input  user\_name = input('Enter your name: ')  #variable for favorite hobby input  favorite\_hobby = input('Enter your favorite hobby: ')  #Variable for user birth year input  birth\_year = int(input('Enter your Birth Year: '))  #Variable for current year input  current\_year = int(input('Enter Current Year: ' ))  current\_age = current\_year - birth\_year  print(f'Hello {user\_name}, you are {current\_age} years old and enjoy {favorite\_hobby}.') |

## Python Operators

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Meaning** | **Example** | **Result** |
| + | Addition | 4 + 7 | 11 |
| - | Subtraction | 12 - 5 | 7 |
| \* | Multiplication | 6 \* 6 | 36 |
| / | Division (float) | 30 / 5 | 6 |
| % | Modulus (remainder) | 10 % 4 | 2 |
| // | Floor Division (integer quotient) | 18 // 5 | 3 |
| \*\* | Exponentiation | 3 \*\* 5 | 243 |

# Version Control

This is the link for help with Github desktop: <https://github.com/CS-2-Community-HUB/resources/blob/main/tools/git_github/github_desktop_guide.md>

Always save and commit and push.

Commit saves the version to my computer.

Push sends my edits to GitHub

A screenshot of a computer

AI-generated content may be incorrect.

# Web Accessibility and Inclusive Design

[Link to Accessibility website](https://accessibility.blog.gov.uk/2016/09/02/dos-and-donts-on-designing-for-accessibility/)

Autistic Spectrum:

* Use Simple colors
* Write in plain english
* Use bullets and simple sentences.
* Build Descriptive buttons
* Build simple and consistent layouts

Screen Readers

* Provide alternate text for images and transcripts for videos
* Structure content using HTML5
* Write descriptive Links and headings.

Low Vision

* Use good color contrasts and larger font sizes.
* Place buttons and notifications in context

Physical or Motor disabilities

* Provide large clickable actions
* Have keyboard or speech use.

Deaf or hard of hearing

* Use of subtitles
* Allow users to request a interpreter

Dyslexia

* Use images and diagrams to support text
* Keep content short, clear and simple
* Allow users to change contrast between background and text.