



**INMAS**

# Google Colab

- *Google Colab*
  - *Run Code or Selection*
  - *Autocomplete and Function Help*
  - *Sidebars: Table of Contents (TOC) and Files*
  - *Keyboard Shortcuts / Command Palette*
  - *Enable GPU/TPUs*

James Balamuta

Inmas Fall 2021 Statistical Methods Workshop



# Lecture Objectives

- **Understand** the trade-offs between Google Colab and a local computer.
- **Explain** course workflow on Google Colab

colab

Two common questions...

1. Do we have to use our own computer to do data science?

**No, use...**

2. Is it possible to do data science on old computers?

**Yes, but consider using....**

The screenshot shows the Google Colaboratory web interface. The browser address bar displays the URL `colab.research.google.com/notebooks/intro.ipynb#scrollTo=5fCEDCU_qrC0`. The page title is "Welcome To Colaboratory". The interface includes a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A "Table of contents" sidebar on the left lists sections: "Getting started", "Data science", "Machine learning", "More Resources", "Machine Learning Examples", and "Section". The main content area features a "What is Colaboratory?" section with a list of features: "Zero configuration required", "Free access to GPUs", and "Easy sharing". Below this is a "Getting started" section explaining that the document is an interactive Colab notebook. It includes a code cell with the following Python code:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

The output of the code cell is the number 86400. The text below the code explains how to execute and edit code cells.

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

# colab

vs.

# Local ()

- Online
- Use on any computer
- Free GPU
- Environment already setup
- Files stored in Cloud

- Offline
- Your computer
- Need to buy a GPU
- Need to setup environment
- Files stored on HD

# Overview



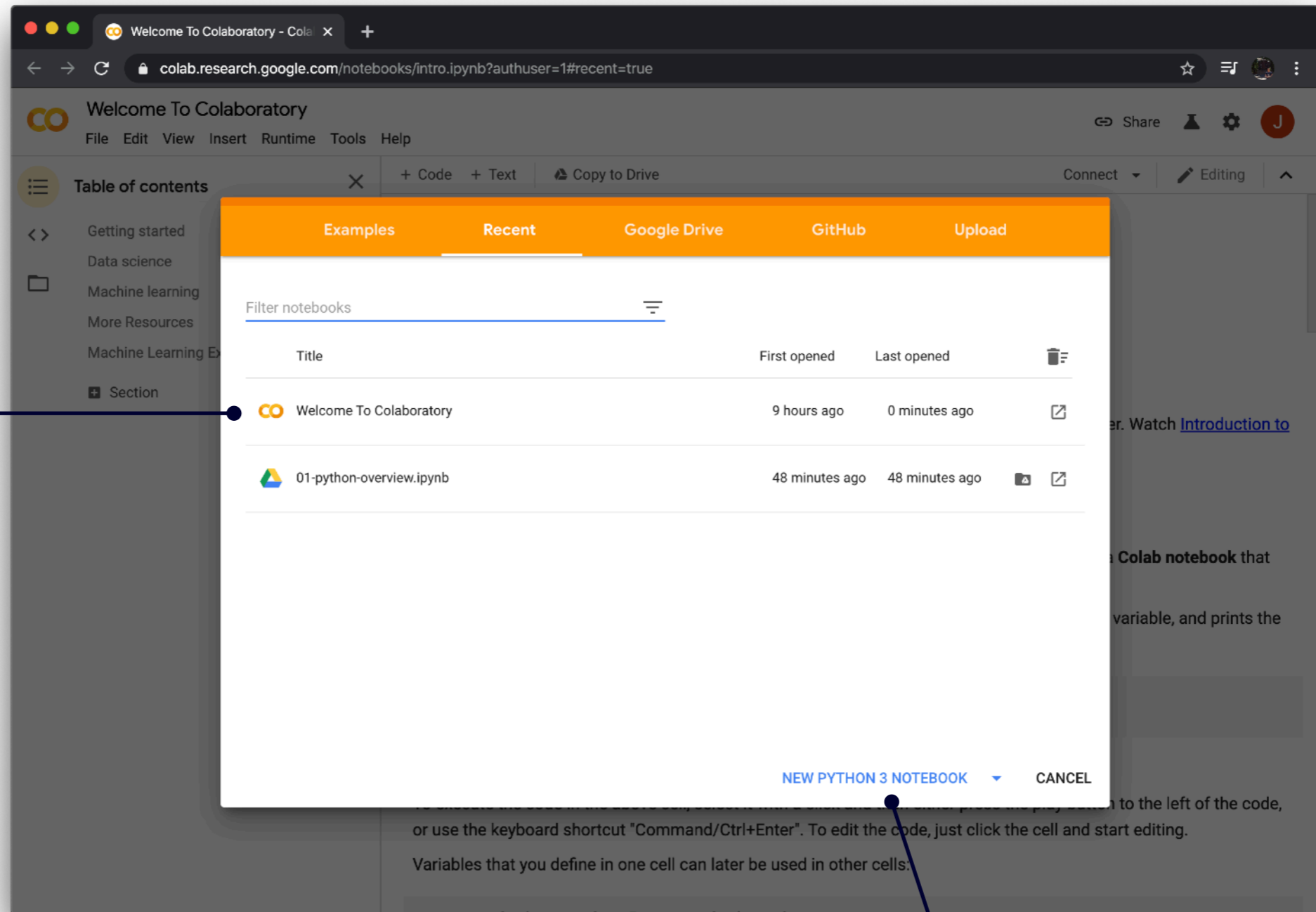
Make sure you're using

colab

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



# File Explorer



**Open**  
Notebook

**Create**  
Notebook

# Run Code

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

**Click Play** —●  
**[⌘/CTRL + ENTER]**

```
▶ 1 seconds_in_a_day = 24 * 60 * 60  
2 seconds_in_a_day
```

**Executed Code** —●

```
[1] 1 seconds_in_a_day = 24 * 60 * 60  
2 seconds_in_a_day
```

**Result** —●

```
👤 86400
```

# Run Code Region

**Open**  
Runtime  
Menu

**Run  
selection**

Runtime Tools Help All changes saved

|                       |                           |
|-----------------------|---------------------------|
| Run all               | ⌘/Ctrl+F9                 |
| Run before            | ⌘/Ctrl+F8                 |
| Run the focused cell  | ⌘/Ctrl+Enter              |
| <b>Run selection</b>  | <b>⌘/Ctrl+Shift+Enter</b> |
| Run after             | ⌘/Ctrl+F10                |
| <hr/>                 |                           |
| Interrupt execution   | ⌘/Ctrl+M I                |
| Restart runtime       | ⌘/Ctrl+M .                |
| Restart and run all   |                           |
| Factory reset runtime |                           |
| <hr/>                 |                           |
| Change runtime type   |                           |

+ Code + Text

```
[41] 1 30 + (10 - 20)  
      20
```

To execute a specific region of code, select it by clicking and dragging over the region and then either go to Runtime menu at the top to select **Run Selection**, or use the keyboard shortcut **CMD/CTRL + Shift + Enter**. This shortcut only works on Google Colab!

```
▶ 1 30 + (10 - 20)  
  -10
```

**Result from  
Selection**

**Highlight & Run Selection  
[⌘/CNTRL + Shift + Enter]**

# Autocomplete

Just Hit **[Tab]**!

## ▼ Autocomplete

When using built-in functions, there is autocomplete available for typing long function names.

Begin by typing the function name into the code cell and, then, press **[Tab]** to autocomplete it.

```
1 import math
2
3 math.fa
```

Autocomplete suggestions:

- fabs
- factorial

**Type**  
function name

**Navigate**  
by using up and  
down arrows

```
1 import math
2
3 math.factorial
```

**Press Tab**  
to autocomplete

# Function Help Documentation

+ Code + Text

▼ Help Documentation

When in doubt about how to use a function, pull up the help documentation by using ?  
`function_name`.

Let's see what the help documentation says about the `min()` function.

```
[18] 1 ?min
```

The help document will open on the right side of Google Colab. Each help file will be given its own tab. Make sure to close tabs after reading through the help documentation.

**Open** Help Documentation

**Result from Selection**

**Close** Help Panel

RAM   | Disk  | Editing | ^

Help X

**Docstring:**  
`min(iterable, *[, default=obj, key=func]) -> value`  
`min(arg1, arg2, *args, *[, key=func]) -> value`

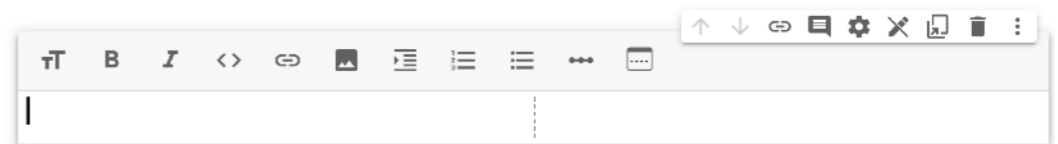
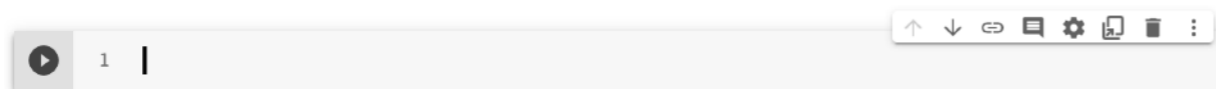
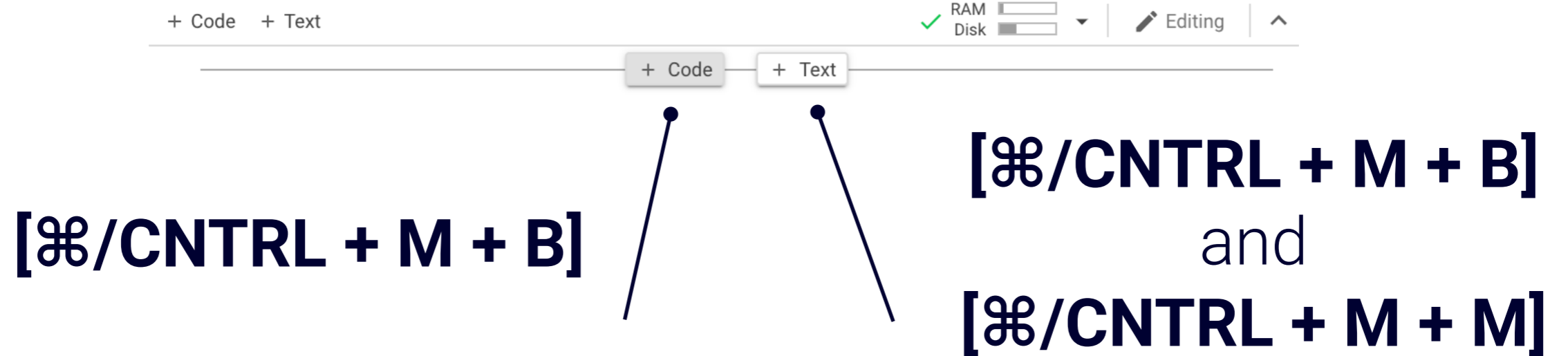
With a single iterable argument, return its smallest item. The default keyword-only argument specifies an object to return if the provided iterable is empty.

With two or more arguments, return the smallest argument.

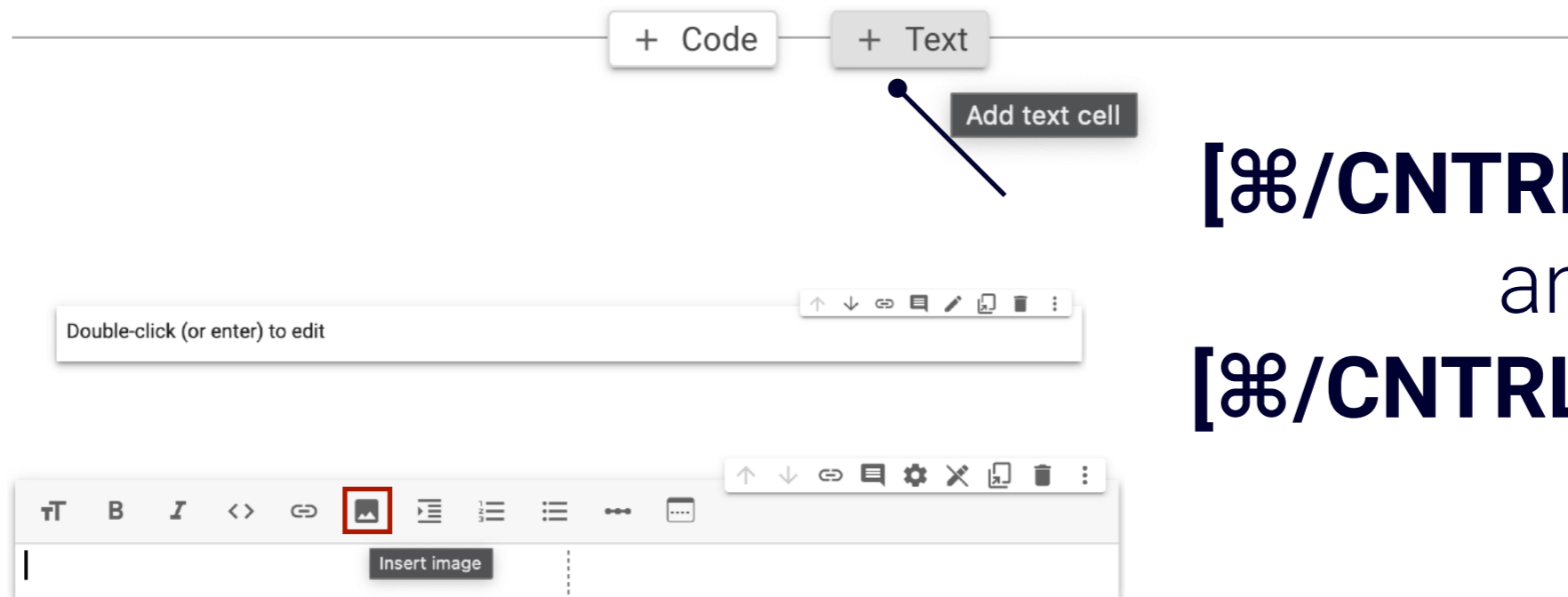
**Type:** builtin\_function\_or\_method

**Help Panel** showing function's docstring

# Add Code or Text Cells

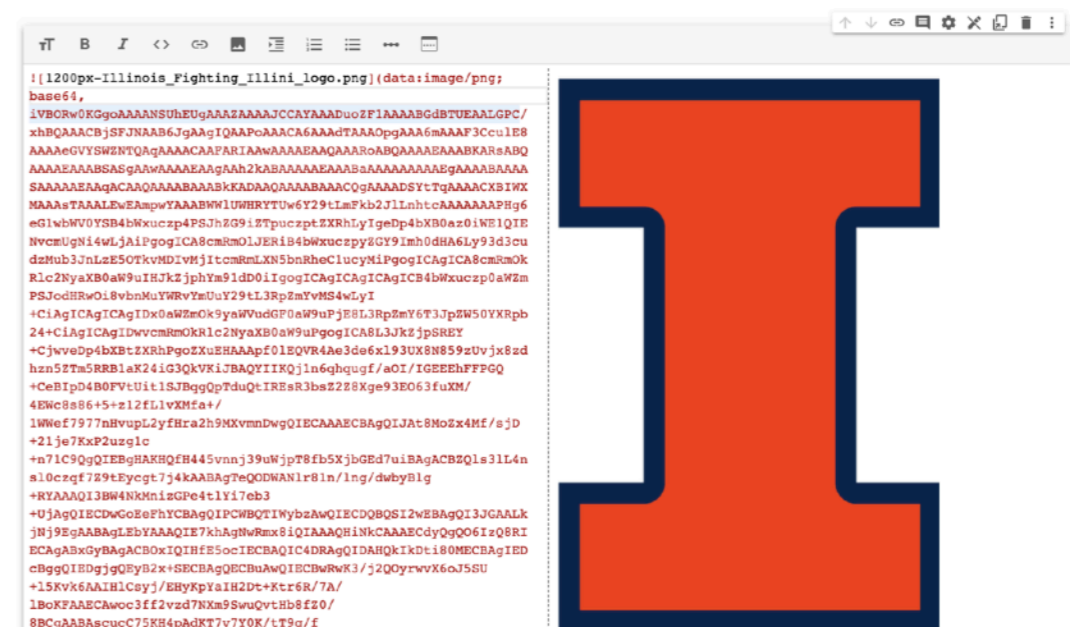


# Add External Picture



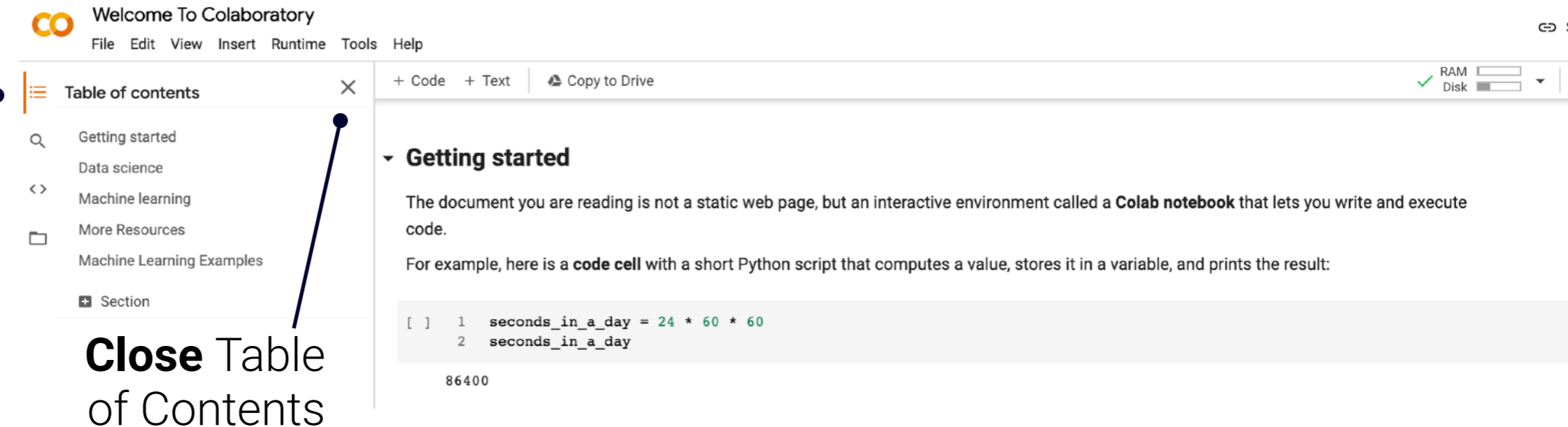
**[⌘/CTRL + M + B]**  
and  
**[⌘/CTRL + M + M]**

| Name                                       | Date Modified     | Size |
|--|-------------------|------|
| > research                                 | Today at 12:45 PM | --   |
| > 1200px-Illinois_Fighting_Illini_logo.png | Today at 12:44 PM | 8 KB |
| > screenshots                              | Today at 12:44 PM | --   |



# View Table of Contents

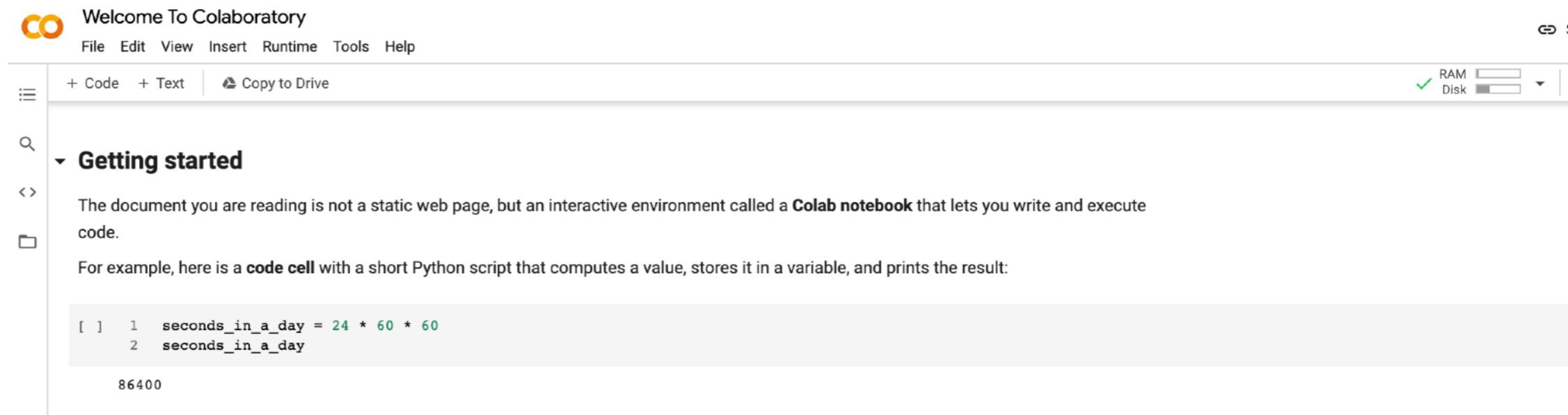
**Open** Table of Contents



The screenshot shows the Colaboratory interface with the 'Table of contents' sidebar open on the left. The sidebar contains a search icon, a list of sections: 'Getting started', 'Data science', 'Machine learning', 'More Resources', and 'Machine Learning Examples', and a 'Section' button with a plus icon. A blue arrow points from the text 'Open Table of Contents' to the sidebar's search icon. Another blue arrow points from the text 'Close Table of Contents' to the close button (an 'X' icon) at the top right of the sidebar. The main content area shows the 'Getting started' section, which includes introductory text and a code cell with a Python script that outputs the number 86400.

**Close** Table of Contents

**Closed** sidebar gives more space!

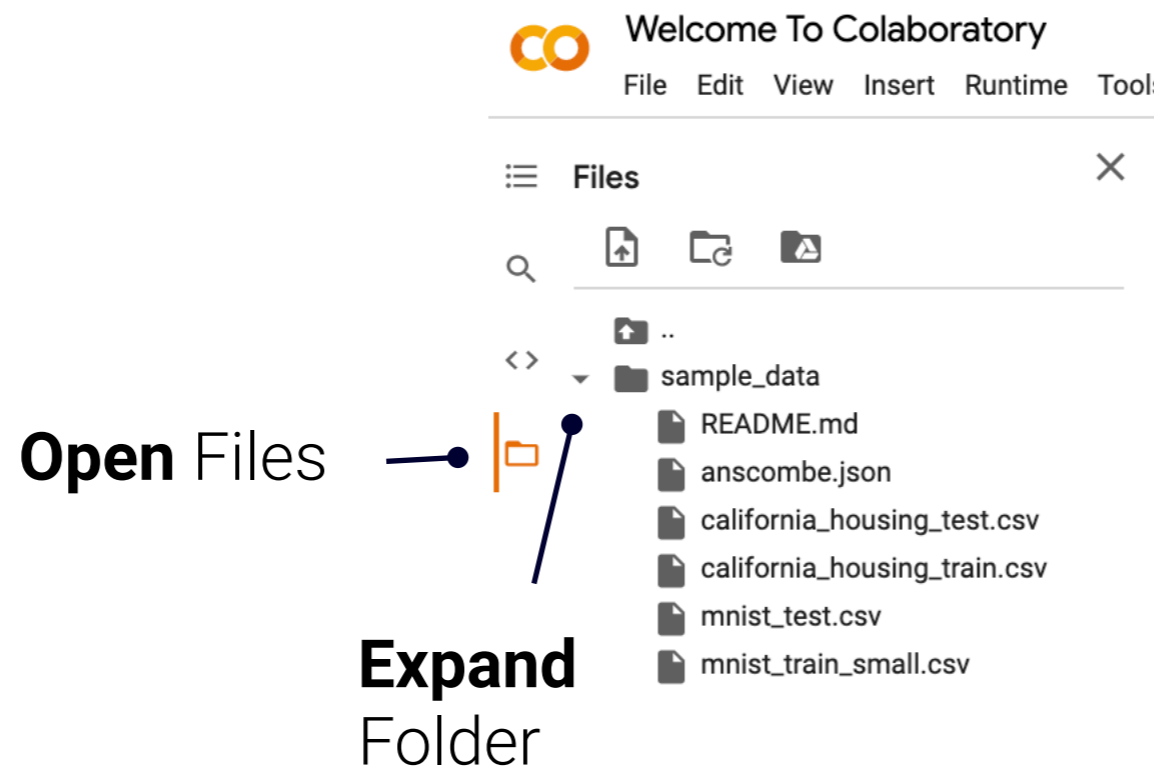


The screenshot shows the same Colaboratory interface as above, but with the 'Table of contents' sidebar closed. The sidebar's search icon, list of sections, and 'Section' button are no longer visible. The main content area now has more space, and the 'Getting started' section is more prominent. The code cell and its output (86400) are still visible.



# See uploaded files

All files are placed at /content



# Hands on Keyboard

## Comparison of Mouse and Keyboard Efficiency

**Richard C. Omanson**  
Infogix

**Craig S. Miller**  
DePaul University

**Elizabeth Young**  
DePaul University

**David Schwantes**  
DePaul University

In many corporate settings, users are required to quickly execute commands. Three methods of issuing commands were compared: 1) selecting a menu item with a mouse (Menu-Mouse condition); 2) selecting a menu item with a keyboard shortcut (Menu-Keyboard condition); and 3) selecting a toolbar item with a mouse (Toolbar-Mouse condition). Users performed one of the three methods across 90 trials and had their speed assessed in blocks of 30 trials. Overall, the Toolbar-Mouse method was the fastest, while the Menu-Keyboard condition showed the most improvement. A GOMS-based model is presented that accounts for differences among methods. **This work confirms the use of toolbars for common commands, but also suggests that for heavily-used interfaces, keyboard shortcuts can be as efficient as toolbars and have the advantage of providing fast access to all commands.**

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1087.6916&rep=rep1&type=pdf>

# Keyboard Shortcuts

Open with ⌘/CTRL + M + H

Welcome To Colaboratory - Co x +

colab.research.google.com/notebooks/intro.ipynb

Welcome To Colaboratory

File Edit View

Table of contents

Getting started

Data science

Machine learning

More Resources

Machine Learning

Section

### Keyboard preferences

Editor key bindings  
default

Enter key accepts suggestions

#### Shortcuts

To add or change a shortcut, click the key combination and then type the new keys. Note that ⌘/Ctrl+M can be used as a prefix for multi-key-event shortcuts.

|                               |                                |                |   |
|-------------------------------|--------------------------------|----------------|---|
| Ctrl+Alt+M                    | Add a comment                  | Set shortcut   | Open notebook settings                        |
| Set shortcut                  | Add a form                     | ⌘/Ctrl+Alt+N   | Open scratch code cell                        |
| Set shortcut                  | Add a form field               | Set shortcut   | Open settings                                 |
| Set shortcut                  | Add code cell                  | ⌘/Ctrl+M P     | Previous cell                                 |
| Set shortcut                  | Add section header cell        | ⌘/Ctrl+P       | Print notebook                                |
| Set shortcut                  | Add text cell                  | ⌘/Ctrl+Shift+Y | Redo cell action                              |
| Ctrl+Space, Option+Esc or Tab | Autocomplete ?                 | Shift+⌘/Ctrl+H | Replace all in current cell                   |
| Set shortcut                  | Clear all outputs              | ⌘/Ctrl+M .     | Restart runtime                               |
| Set shortcut                  | Clear selected outputs         | Set shortcut   | Restart runtime and run all cells in notebook |
| ⌘/Ctrl+] ]                    | Collapse all/selected sections | ⌘/Ctrl+F9      | Run all cells in notebook                     |
| ⌘/Ctrl+/ /                    | Comment current line           | Alt+Enter      | Run cell and insert new cell                  |
| Set shortcut                  | Comments sidebar               | Shift+Enter    | Run cell and select next cell                 |
| Set shortcut                  | Connect to a local runtime     |                |   |

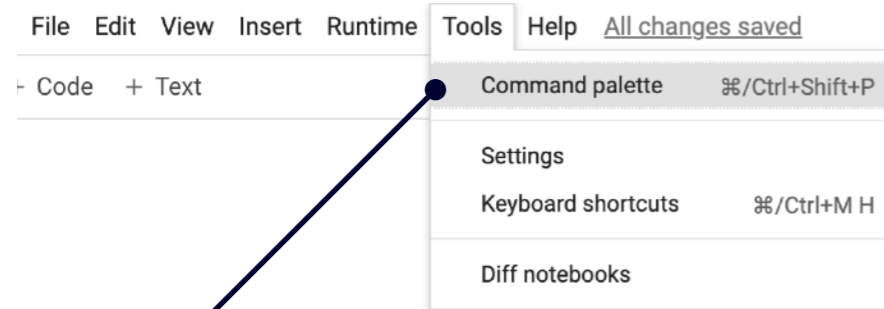
RESTORE DEFAULTS CANCEL SAVE

or use the keyboard

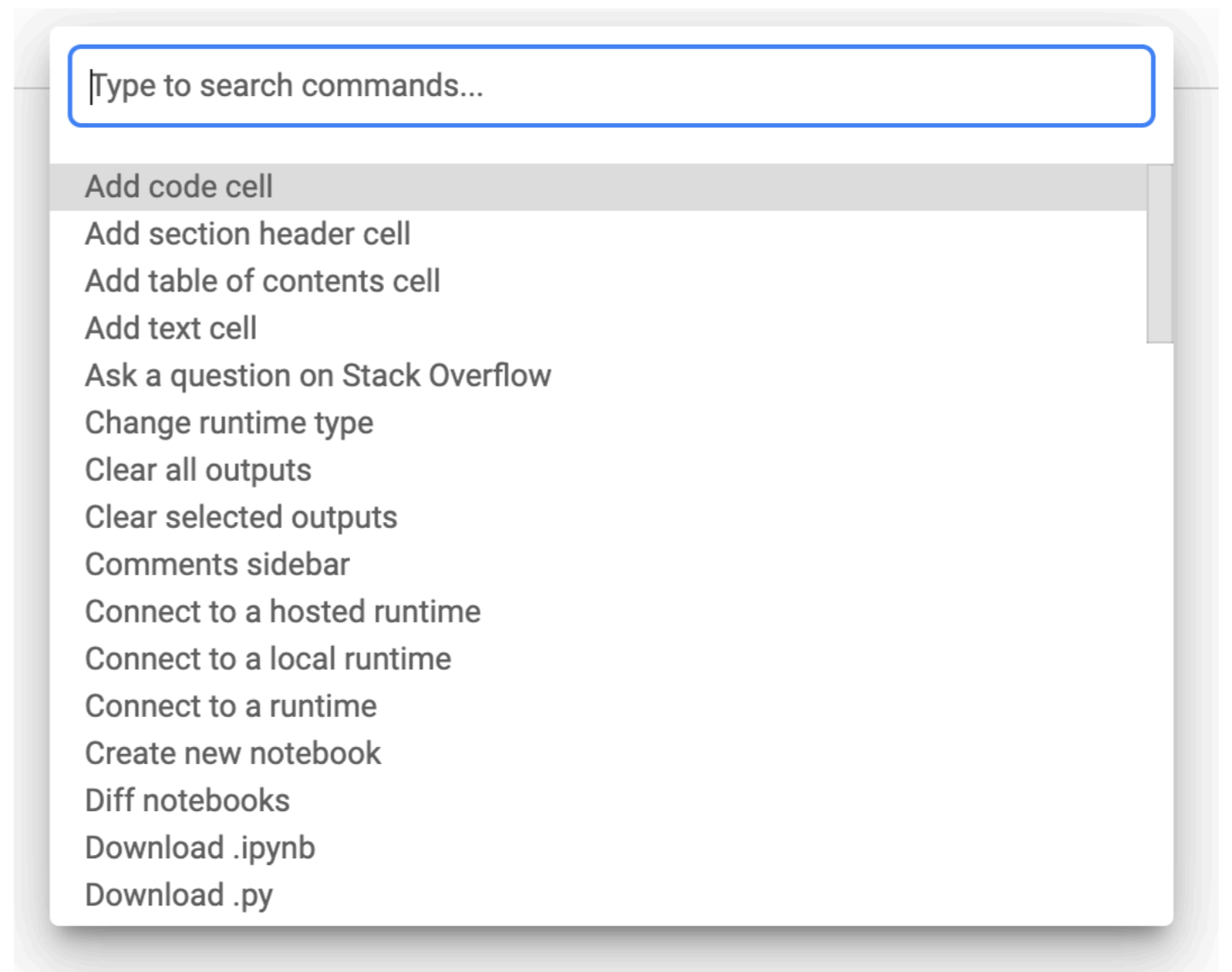
\* Colab shortcuts prefixes all Jupyter shortcuts with ⌘/CTRL + M

# Command Palette

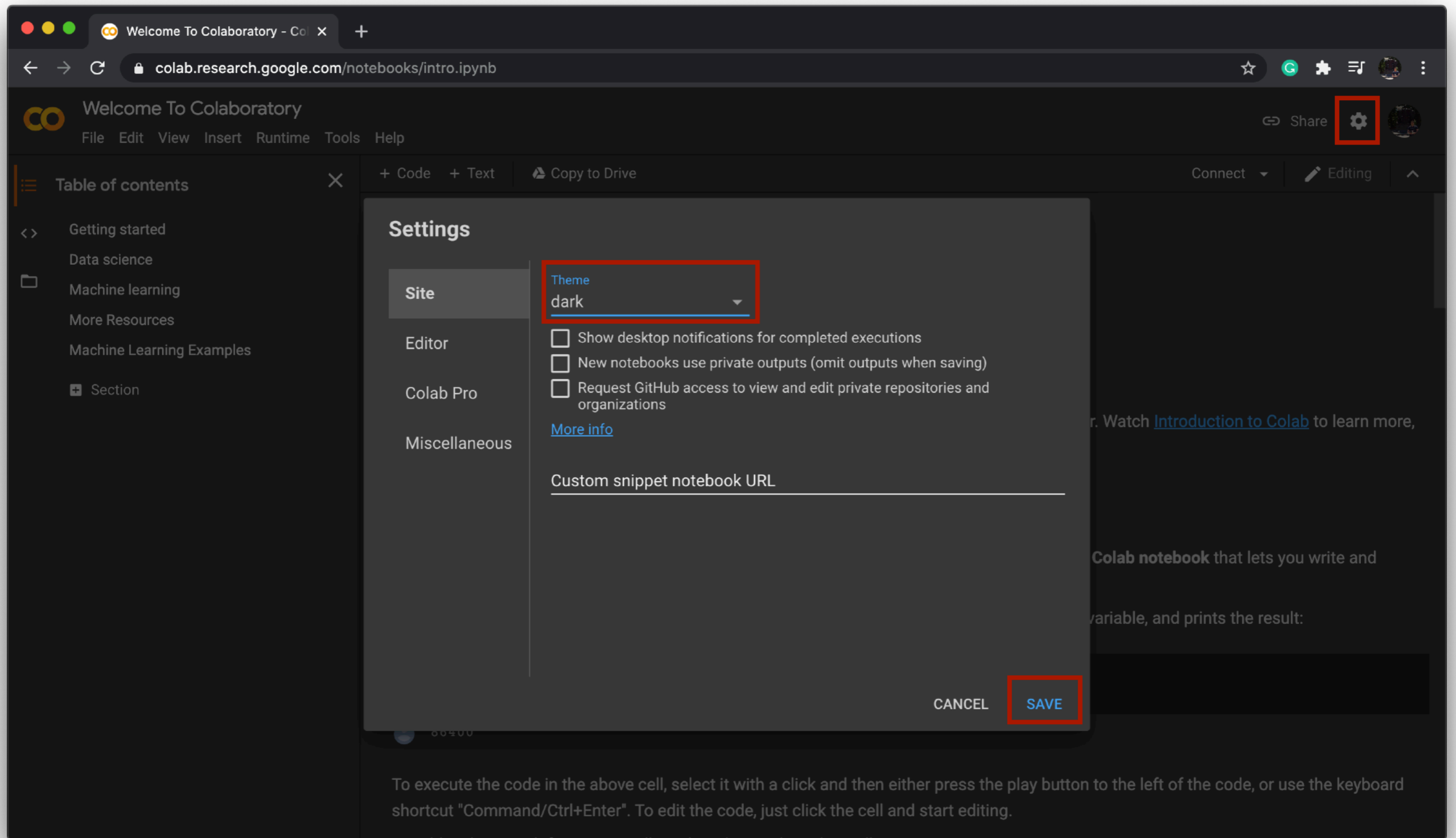
## English-version of Keyboard Shortcuts



**[⌘/CTRL + SHIFT + P]**



# Night/Day Themes



The image shows a screenshot of the Google Colaboratory interface. The browser address bar displays the URL `colab.research.google.com/notebooks/intro.ipynb`. The page title is "Welcome To Colaboratory". The main menu includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A "Share" button and a settings gear icon are visible in the top right corner. The settings dialog is open, showing the "Settings" panel with a sidebar containing "Site", "Editor", "Colab Pro", and "Miscellaneous". The "Theme" dropdown menu is expanded, showing "dark" as the selected option. Below the theme settings, there are three checkboxes: "Show desktop notifications for completed executions", "New notebooks use private outputs (omit outputs when saving)", and "Request GitHub access to view and edit private repositories and organizations". A "More info" link is also present. At the bottom of the dialog, there is a "Custom snippet notebook URL" input field and two buttons: "CANCEL" and "SAVE".

Settings

Theme  
dark

Show desktop notifications for completed executions

New notebooks use private outputs (omit outputs when saving)

Request GitHub access to view and edit private repositories and organizations

[More info](#)

Custom snippet notebook URL

CANCEL SAVE

# Enable GPU/TPU

Increase model training speed with large data

Runtime Tools Help All changes saved

|                      |                    |
|----------------------|--------------------|
| Run all              | ⌘/Ctrl+F9          |
| Run before           | ⌘/Ctrl+F8          |
| Run the focused cell | ⌘/Ctrl+Enter       |
| Run selection        | ⌘/Ctrl+Shift+Enter |
| Run after            | ⌘/Ctrl+F10         |

---

|                        |            |
|------------------------|------------|
| Interrupt execution    | ⌘/Ctrl+M I |
| Restart runtime...     | ⌘/Ctrl+M . |
| Restart and run all... |            |
| Factory reset runtime  |            |

---

**Change runtime type**

---

Manage sessions

View runtime logs

## Notebook settings

Runtime type

Python 3

Hardware accelerator

GPU

Omit code cell output when saving this notebook

CANCEL

SAVE

# Summary

- Provided an overview of Google Colab Features
- Emphasized how to add code cells and execute them.

This work is licensed under the  
Creative Commons  
Attribution-NonCommercial-  
ShareAlike 4.0 International  
License

