

INF206

Programming

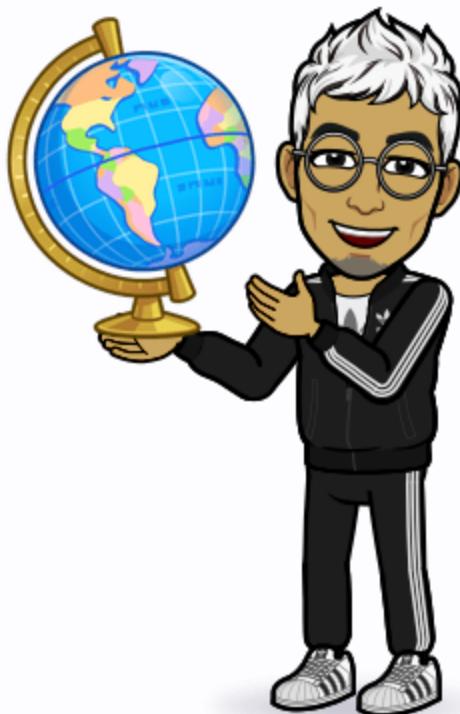
プログラミング

Week 1 | September 28, 2022

Welcome.



Hello!



Call me "Yoh"



Programming | プログラミング



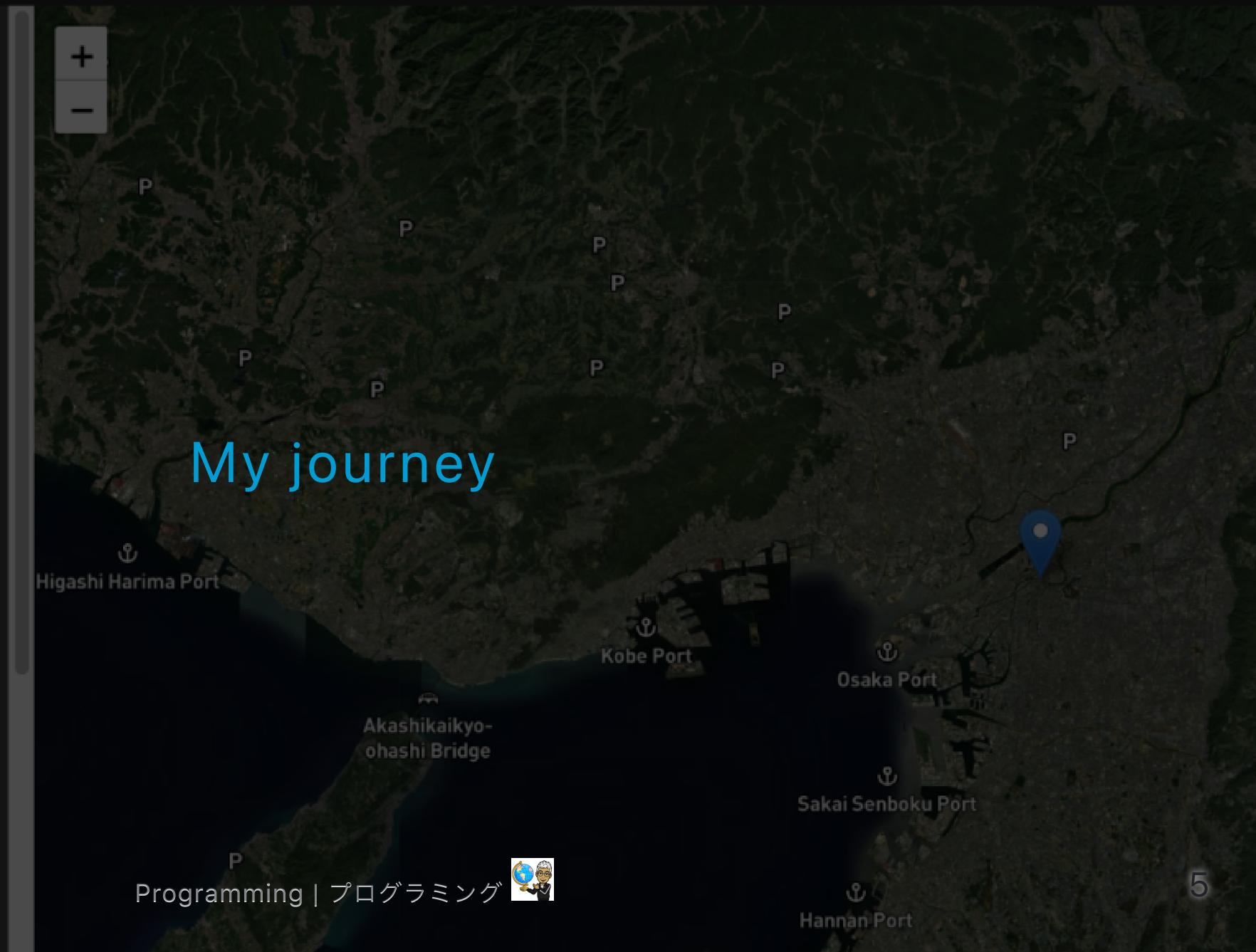
```
// my live in data
const data = [
  {
    'id': 0,
    'title':'Hello Osaka',
    'lat': 34.6937,
    'lon': 135.5023,
    'zoom': 10,
    'image': 'https://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/Osaka\_Castle\_02bs3200.jpg/320px-Osaka\_Castle\_02bs3200.jpg'
  },
  {
    'id': 1,
    'title':'Hello Cali',
    'lat': 3.429729892370133,
    'lon': -76.54106493872214,
    'zoom': 16,
    'image': 'https://upload.wikimedia.org/wikipedia/commons/thumb/e/e8/Pascual\_Guerrero\_U-20WC\_2011\_CMN-NZL.JPG/320px-Pascual\_Guerrero\_U-20WC\_2011\_CMN-NZL.JPG'
  },
  {
    'id': 2,
    'title':'Hello Bangkok',
    'lat': 13.7563,
    'lon': 100.5018,
    'zoom': 14,
    'image': 'https://upload.wikimedia.org/wikipedia/commons/thumb/f/f6/Bangkok\_skytrain\_sunset.jpg/320px-Bangkok\_skytrain\_sunset.jpg'
  },
  {
    'id': 3,
    'title':'Hello Tokyo',
    'lat': 35.687998301985985,
    'lon': 139.5297974778005,
    'zoom': 16,
    'image': 'https://www.stofficetokyo.ch/sites/default/files/styles/featured\_image\_840x572\_/public/2019-01/icu\_dronei\_34ab\_170430-42\_r.jpg?itok=mkG94UWD'
```



Hello Osaka



Hello Cali









Your turn! | あなたの番

- your name | 名前
- your passion | 最近ハマってるもの
- your research interest | 研究するとすればこんなことやってみたい



My teaching philosophy

私の教育哲学

1: class = community

2: it's a two way street



Source: LA Times

3: sharing is caring



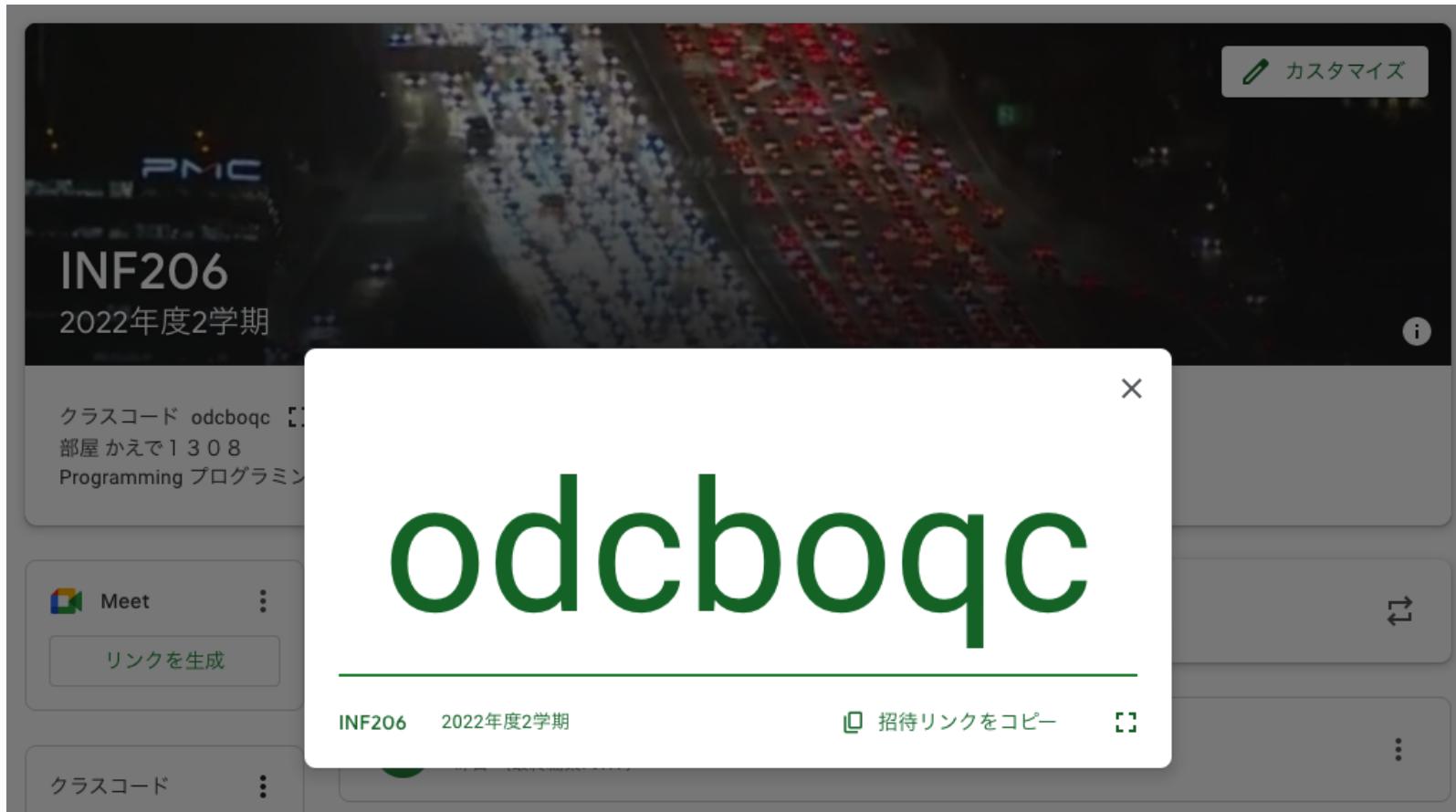
Source: CDTA Law

Join the class!

Step 1



Step 2



What is Programming?

プログラミングとは？

思い浮かぶものをJam Boardに書いてね



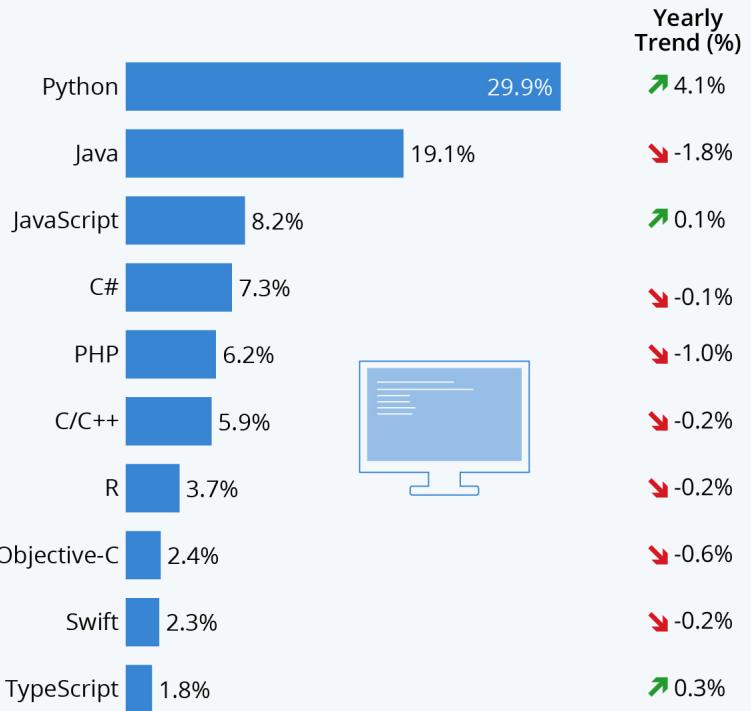
Why Python?

なぜPythonなの？



Python Remains Most Popular Programming Language

Popularity of each programming language based on share of tutorial searches in Google



Yearly trend compares percent change from Feb 2019 to Feb 2020
Sources: GitHub, Google Trends



statista



Pythonで何が作れる？



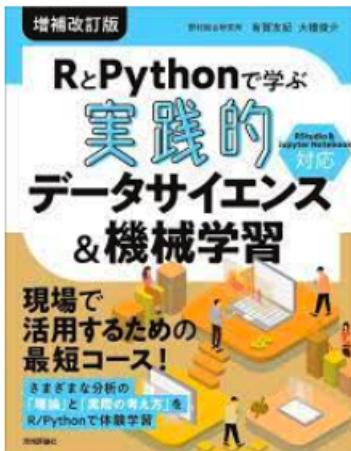
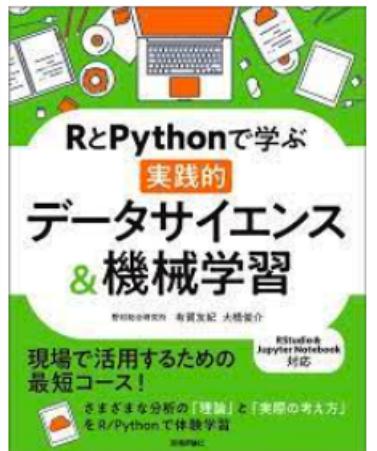
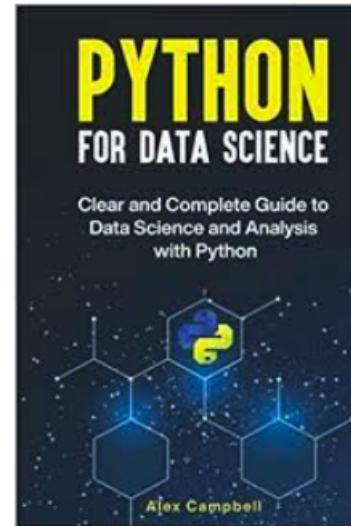
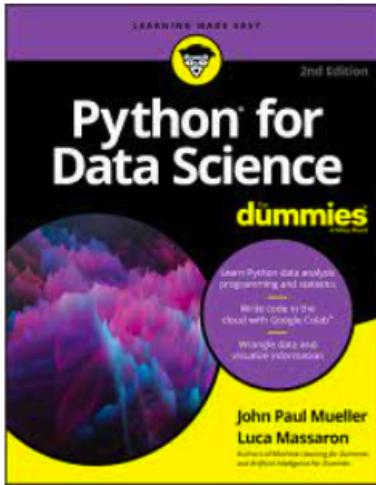
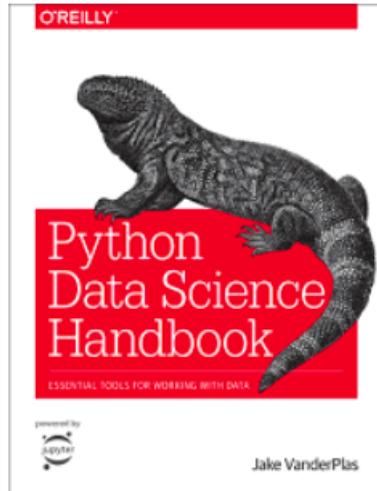
Web development



Who uses python?



Data science

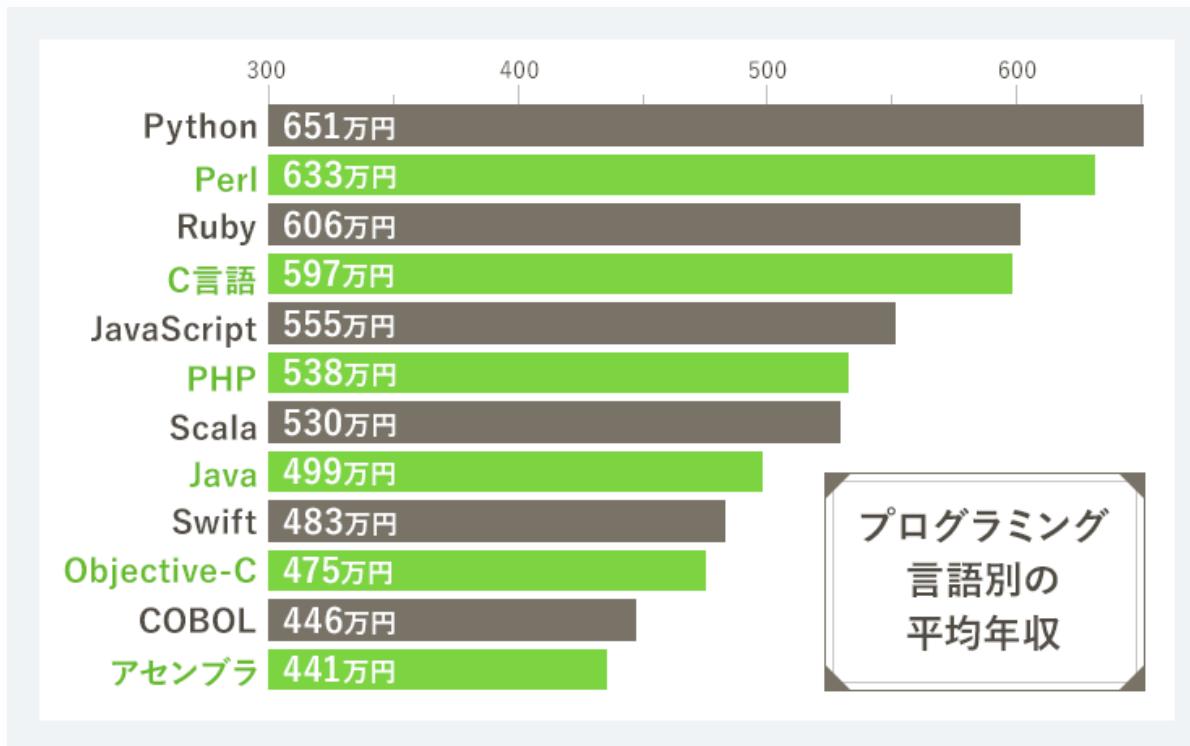


Gaming

```
conn.execute(''CREATE TABLE users
sqlite3.OperationalError: table users already exists
(base) MacBook-Pro-3:Shopping-Cart-master users$ python3 main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with fsevents reloader
* Debugger is active!
* Debugger PIN: 328-110-320
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/css/topStyle.css HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/images/logo.png HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/css/home.css HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/uploads/Kinkaku_3_i_by_Elizabeth_K_Joseph.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/Images/shoppingCart.png HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/uploads/Untitled_by_Troy_Jarrell.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/uploads/The_Sky_Is_The_Limit_by_Kaushik_Panchal.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/uploads/Mountainous_View_by_Sven_Scheuermeier.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/uploads/Untitled_by_Aaron_Burden.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/uploads/Untitled_0026_by_Mike_Sinko.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/uploads/Untitled_7019_by_Mike_Sinko.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:44] "GET /static/uploads/Yellow_Jacket_by_Manuel_Freil.png HTTP/1.1" 200 -
127.0.0.1 - - [15/Jun/2021 14:26:45] "GET /favicon.ico HTTP/1.1" 404 -
`~(base) MacBook-Pro-3:Shopping-Cart-master users$ cd /Users/users/Downloads/M-Level-1-master
(base) MacBook-Pro-3: M-level-1-master users$ python3 mario_level_1.py
pygame 2.0.1 (SDL 2.0.14, Python 3.8.3)
Hello from the pygame community. https://www.pygame.org/contribute.html
libpng warning: iccCP: known incorrect sRGB profile
libpng warning: iccCP: known incorrect sRGB profile
![[A^[[A^[[A(base) MacBook-Pro-3:Mario-Level-1-master users$ python3 mario_level_1.py [
pygame 2.0.1 (SDL 2.0.14, Python 3.8.3)
Hello from the pygame community. https://www.pygame.org/contribute.html
libpng warning: iccCP: known incorrect sRGB profile
libpng warning: iccCP: known incorrect sRGB profile
![[A^[[A^[[A(base) MacBook-Pro-3:Mario-Level-1-master users$ python3 mario_level_1.py [
pygame 2.0.1 (SDL 2.0.14, Python 3.8.3)
Hello from the pygame community. https://www.pygame.org/contribute.html
libpng warning: iccCP: known incorrect sRGB profile
libpng warning: iccCP: known incorrect sRGB profile
```



Python 将来性



source



Schedule

Schedule スケジュール

Week	Topic
Week 1	オリエンテーション：プログラミングとは？
Week 2	第1章 変数とデータ型
Week 3	第2章 リスト、ディクショナリ、タプル、セット

Week	Topic
Week 4	第3章 条件分岐、フローチャート-1-
Week 5	第3章 条件分岐、フローチャート-2-
Week 6	第4章 繰り返し-1-

Week	Topic
Week 7	第4章 繰り返し-2-
Week 8	復習①
Week 9	第5章 関数

Week	Topic
Week 10	組み込み関数、モジュール
Week 11	ファイル操作
Week 12	Pandas, Numpy

Week	Topic
Week 13	TBD
Week 14	演習問題

Textbook | 教科書



【タイトル】
スッキリわかるPython入門

【著者】
国本大悟, 須藤秋良

【発行社】
インプレス

【定価】
2,640円 (+税)



はじめての人・つまずいた人・納得したい人のための入門書
**サクサク進めてしくみもバッチリ!
プログラミングの「本質」を
身に付け未来への
道を切り拓こう!**



業務自動化、データサイエンス、人工知能、
アプリ開発プログラマへの第一歩!

Grading | 成績評価

- 出席（15回） 30%
- 宿題（10回） 30%
 - 回数は変わらるかも
- テストもしくはプレゼン（2回） 40%

出席 $(1500 \times 1/15 \times 0.3)$

+

宿題 $(1000 \times 1/10 \times 0.3)$

+

プレゼン $(200 \times 1/2 \times 0.4)$

=

100

- 出席率60%以上で評価対象となります
(大学全体と同様)
- 宿題は期限までに出せば採点

Let's try some code

いきなりだけど、Pythonやりましょう！



go to

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

例 最近 Google ドライブ GitHub アップロード

ノートブックを絞り込む

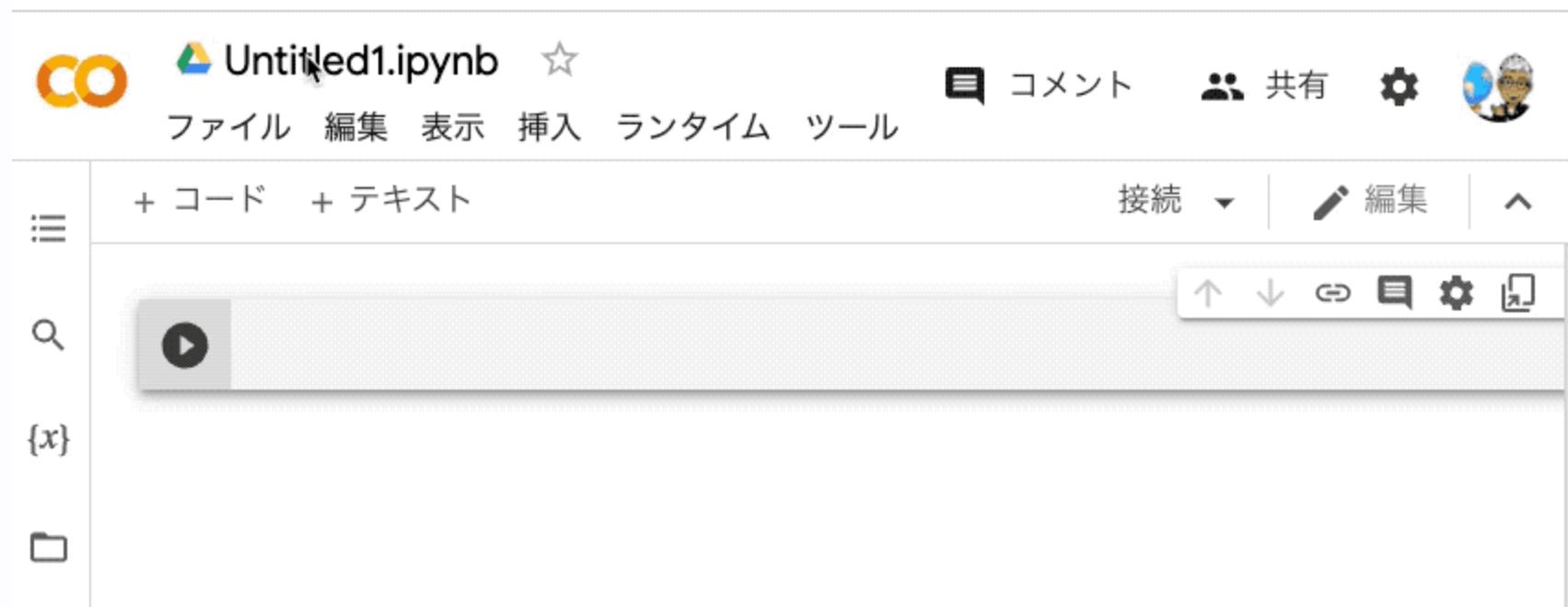
タイトル	最終閲覧	最初に開いた日時	
05_choropleth.ipynb	8月18日	8月18日	
Untitled0.ipynb	5月20日	4月29日	
Welcome To Colaboratory	5月20日	2020年11月12日	
ee-api-folium-setup.ipynb	2021年11月30日	2020年11月13日	
Mathematical Operations	2021年11月20日	2021年11月20日	

ノートブックを新規作成 キャンセル



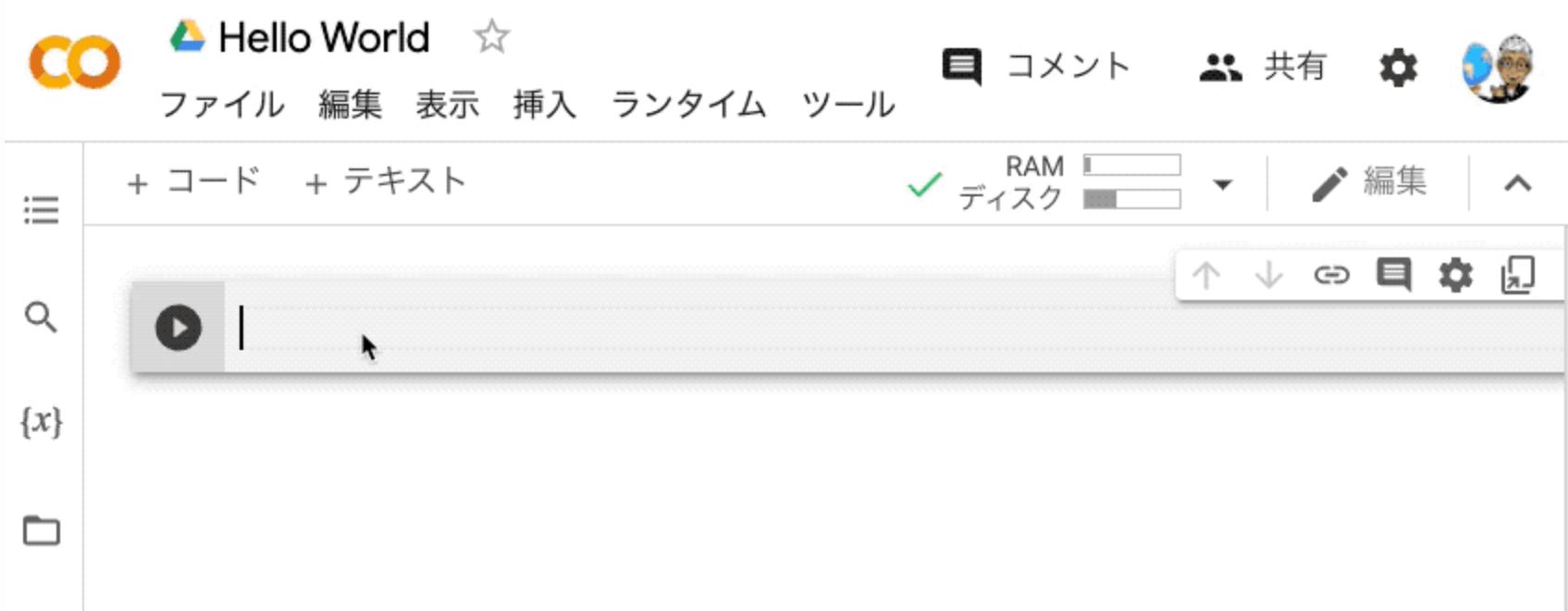
Add a title

ノートブックにタイトルを



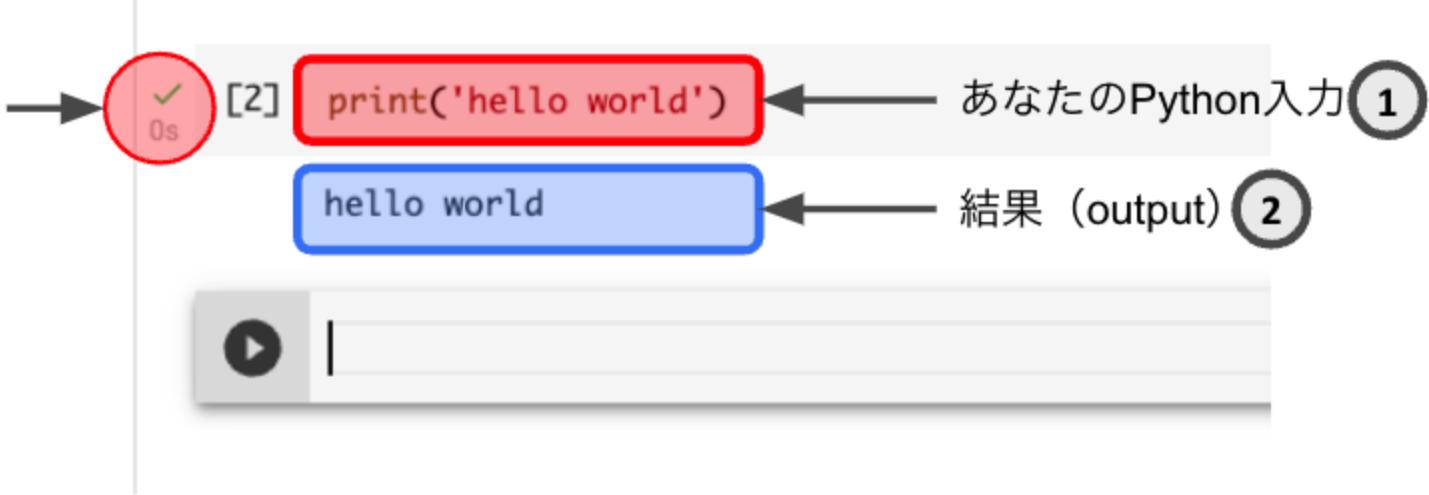
Write some code
コードを書いてみよう

```
print('hello world')
```



打ち終わったら shift + enter

この作業がかかった時間



Let's do something advanced!

ではちょっとかっこいいことやろう

Import a library

ライブラリーをインポート

- **plotly express**

```
import plotly.express as px
```



Write some code

コードを書こう

```
fig = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])  
fig.show()
```

楽しいでしょう 😊

他のコードサンプルをコピペしてみよう

- [plotly express](#)

ではColabでレポートを書こう



そうなんです。ColabではPython Codeとテキストを両方同じ
ページに出せる！

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

- Title Bar:** CO Hello World.ipynb ☆
- Toolbar:** ファイル 編集 表示 挿入 ランタイム ツール
- Cell Header:** + コード + テキスト ✓ RAM ディスク | 編集
- Cell Content:**
 - Code cell: print('hello world')
 - Output cell: hello world
 - In-cell code cell: [1] import plotly.express as px

Hello World.ipynb ☆

ファイル 儲蓄 表示挿入 ランタイムツール ヘルプ すべての変更を保存しました

+ コード + テキスト

RAM ディスク 編集

▼ Hello World

[x] Hi, my name is Yoh. I am here to teach you Python. Let's have fun! よろしく!

▼ Getting started

まずはPython Kernelとコミュニケーションをとりましょう。

```
[ ] print('hello world')  
hello world
```

▼ ライブラリーの取り入れ

次に外部からのライブラリーを取り入れて、パワーアップしたノートブックにしましょう。

ここでは次のライブラリーを使います。

- [plotly.express](#)

```
[1] # import library  
import plotly.express as px
```

```
[2] fig = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])  
fig.show()
```

A scatter plot with the x-axis labeled 'X' and the y-axis labeled 'Y'. The x-axis ranges from 0 to 4 with increments of 0.5. The y-axis ranges from 0 to 16 with increments of 2. Five blue square markers are plotted at the coordinates (0,0), (1,1), (2,4), (3,9), and (4,16).



宿題

Collab Notebookを提出



必要最低項目：



Introduction

- タイトルと自分の自己紹介（テキストブロック）
 - なぜコーディングを始めたいか
 - コーディングでどんなものを作りたいか



Coding

- 今日習ったコーディングの課題をコードブロックを作成し、アウトプットを表示（2つ以上）
- クラスで習ってないコーディングの課題（教科書、もしくはWebなどで調べて）をコードブロックを作成し、アウトプットを表示（2つ以上）
- 各コードブロックの前に必ず説明をするテキストブロックを足すこと
- 各コードブロックのアウトプットの後に必ず結果を解釈するテキストブロックを足すこと