

# 環境構築と 競技プログラミングサイト紹介

# 目標

- ・ Python開発環境を各自のPC上に構築し  
Pythonコードを実行
- ・ 競技プログラミングサイトを知ってもらう  
(AtCoder)
- ・ 初学者向け

# 方針

- ・ このスライドを使って説明
- ・ 各自でこのスライドを参考に進めてOK
- ・ 詰まったらチャットを投げるとZoom内の主催者側がチャットで対応
- ・ Windowsを前提にしているので他OSで詰まったら積極的にチャットをお願いします

# 目次

## 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成して開く
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

## AtCoder紹介

1. AtCoderとは？
2. AtCoderへの登録
3. コンテストへの参加
4. 常設コンテスト
5. APG4b
6. JOI過去問集の紹介

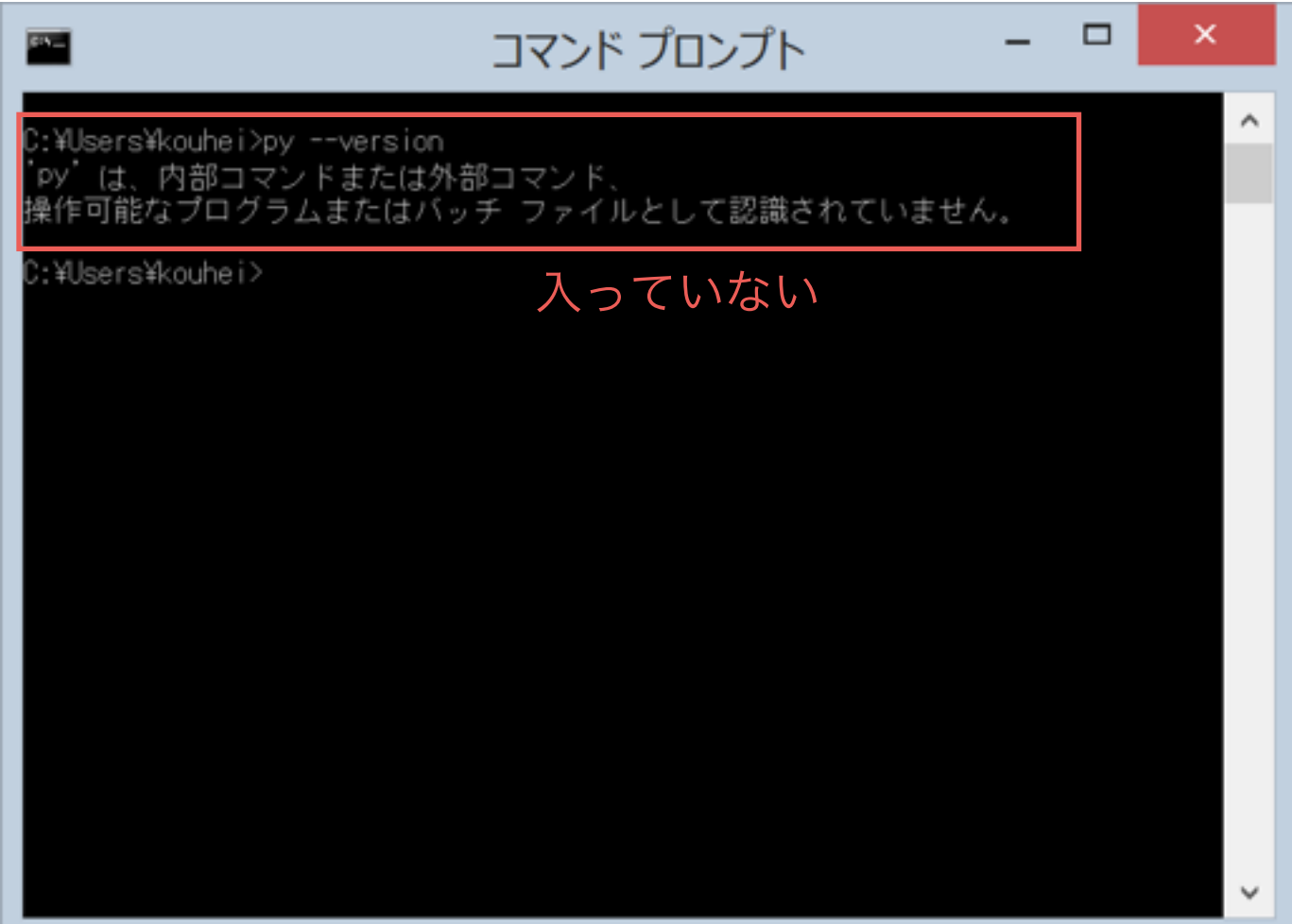
# 環境構築

# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

# Python3をインストール

- ・ Python3がPCにインストールされているか確認
  - Windowsの検索ツールに「コマンドプロンプト」
  - コマンドプロンプトアプリを選択し開く
  - コンソールに「py --version」
  - 結果が「Python 3.……」ならインストール済み
  - それ以外ならPython3のインストール



The screenshot shows a Windows Command Prompt window titled "コマンド プロンプト". The command prompt shows the command `C:\Users\kouhei>py --version` and the output: `'py' は、内部コマンドまたは外部コマンド、操作可能なプログラムまたはバッチ ファイルとして認識されていません。`. A red box highlights the error message. Below the command prompt, the text "入っていない" is written in red.

```
C:\Users\kouhei>py --version
'py' は、内部コマンドまたは外部コマンド、
操作可能なプログラムまたはバッチ ファイルとして認識されていません。
C:\Users\kouhei>
```

入っていない

# Python3をインストール

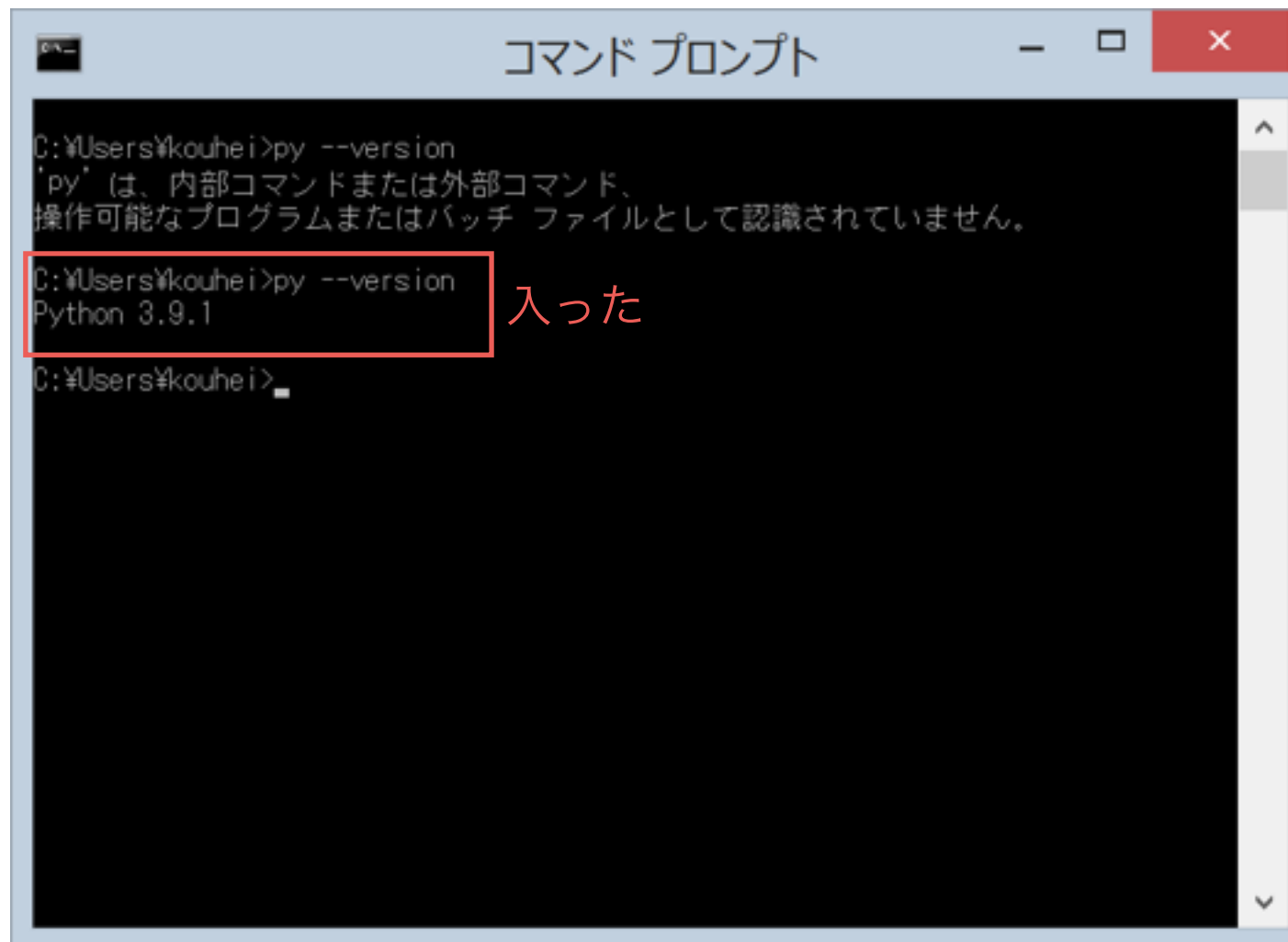
- ・ Pythonをダウンロードする
  - ・ Microsoft Storeを開く
  - ・ 「python3」で検索
  - クリックしてインストール
- もしくはPython公式からダウンロード
- (<https://www.python.org/downloads/>)





# Python3をインストール

- ・ Python3が入っているか確認する
  - 「コマンドプロンプト」 → 「py --version」



```
コマンド プロンプト

C:\Users\¥kouhei>py --version
'py' は、内部コマンドまたは外部コマンド、
操作可能なプログラムまたはバッチ ファイルとして認識されていません。

C:\Users\¥kouhei>py --version
Python 3.9.1

C:\Users\¥kouhei>
```

入った

# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

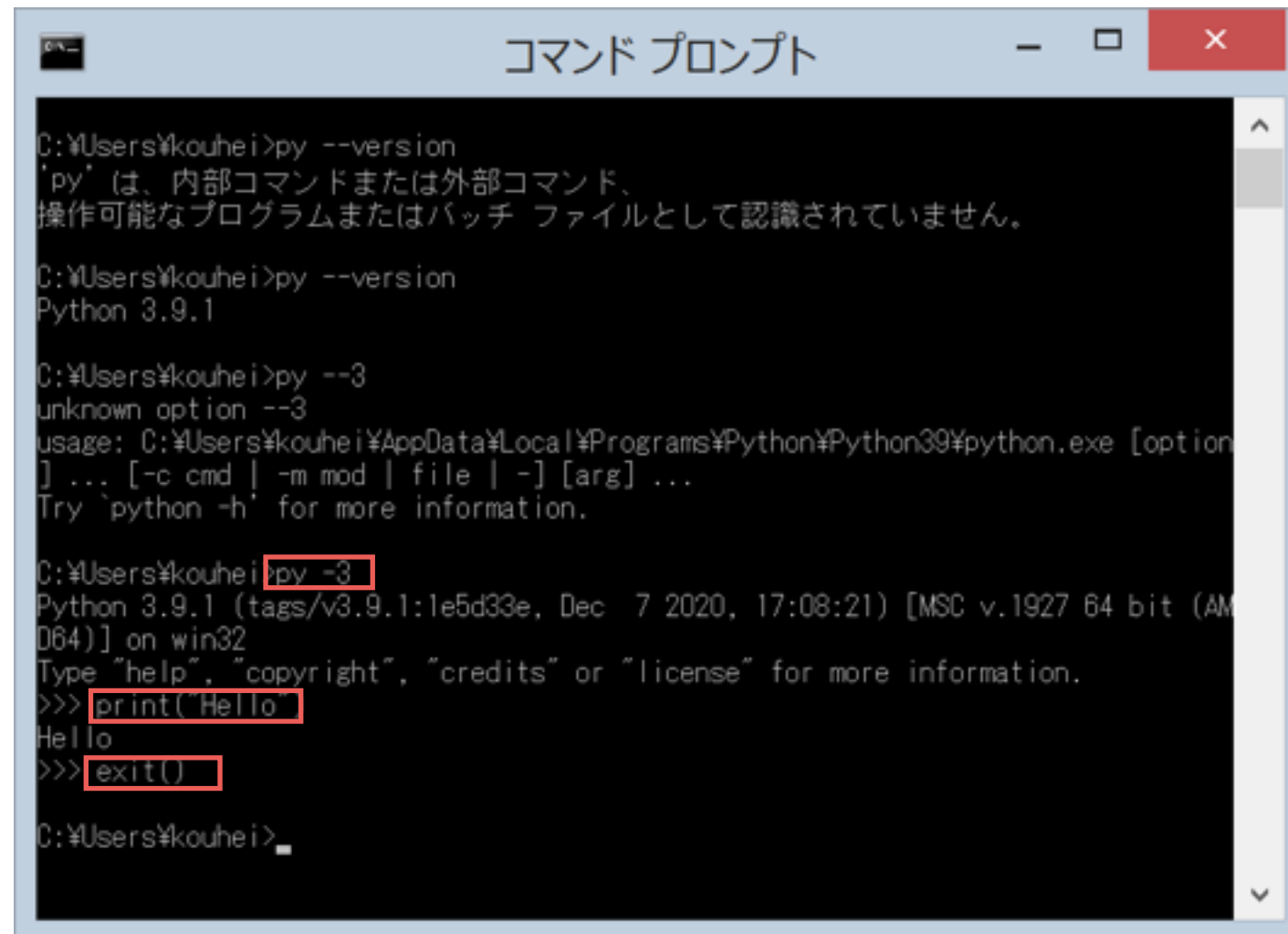
# 対話モードを利用

- ・ 「コマンドプロンプト」 → 「py -3」

- ・ 「print("Hello")」

- ・ 確認

- ・ exit()



```
コマンド プロンプト

C:\Users\kouhei>py --version
'py' は、内部コマンドまたは外部コマンド、
操作可能なプログラムまたはバッチ ファイルとして認識されていません。

C:\Users\kouhei>py --version
Python 3.9.1

C:\Users\kouhei>py --3
unknown option --3
usage: C:\Users\kouhei\AppData\Local\Programs\Python\Python39\python.exe [option
] ... [-c cmd | -m mod | file | -] [arg] ...
Try 'python -h' for more information.

C:\Users\kouhei>py -3
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello')
Hello
>>> exit()

C:\Users\kouhei>
```

# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

# Visual Studio Codeをインストール

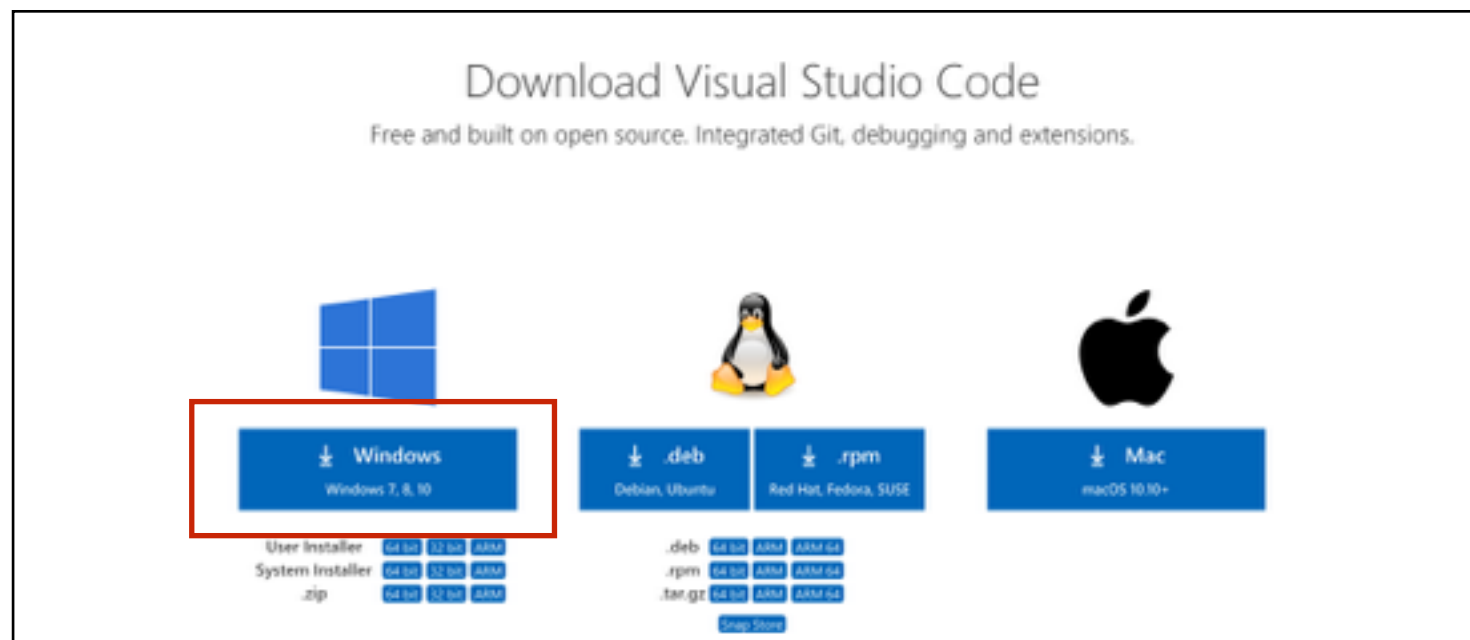
- ・ Visual Studio Code
  - ・ Microsoft社が開発したコードを記述するツール
  - ・ コードを書きやすくしてくれる
  - ・ 無料

# Visual Studio Codeをインストール

- ・ インストーラをダウンロード

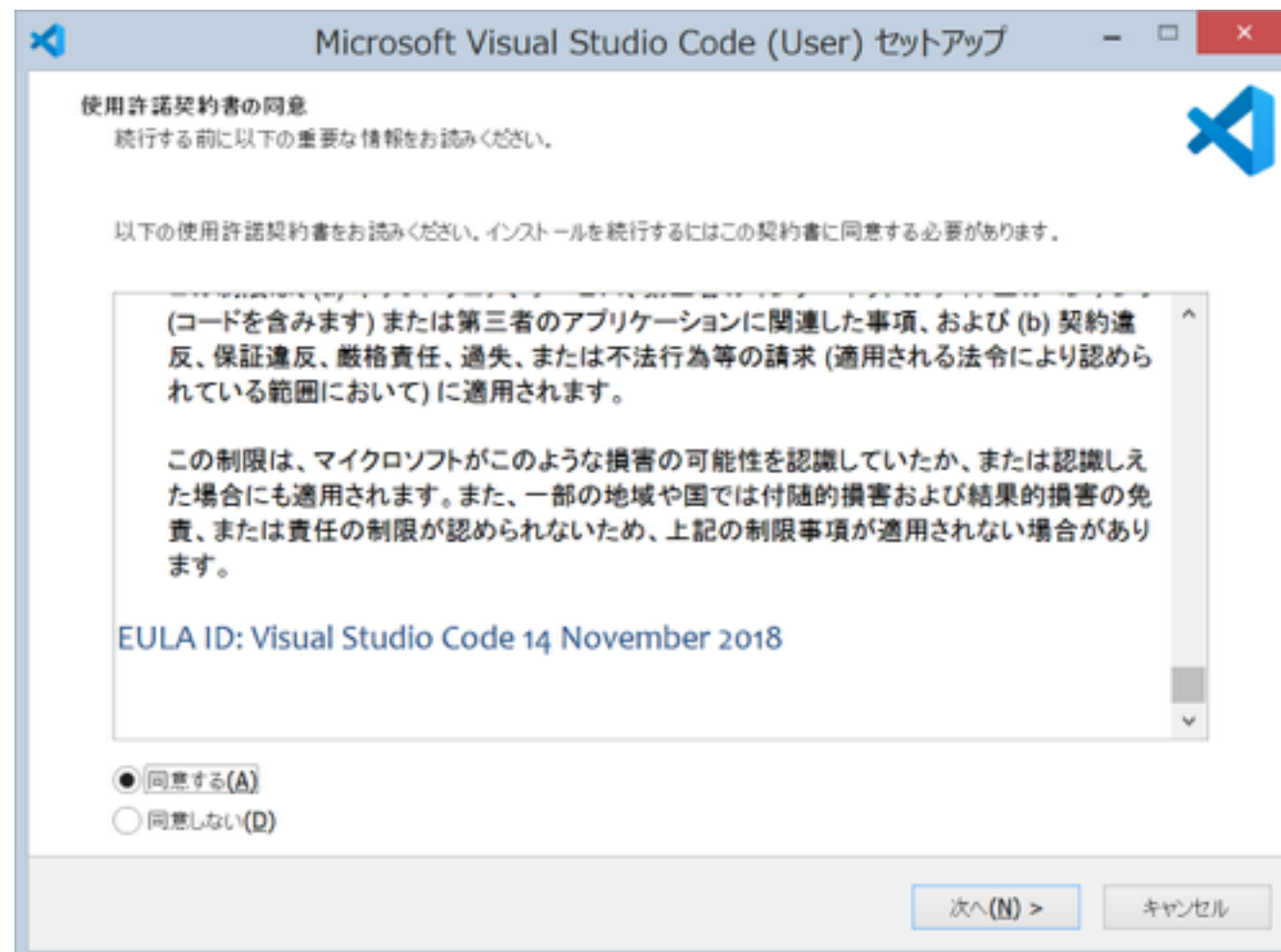
url: <https://code.visualstudio.com/Download>

Windowsを選択



# Visual Studio Codeをインストール

- ・ インストーラーを開始
- ・ ダウンロードされたexeファイルを開く



# Visual Studio Codeをインストール

- ・ Visual Studio Codeを開く
  - ・ 「Visual Studio Code」で検索
- ・ 開く
- ・ もしくはデスクトップアイコンにしたならダブルクリック

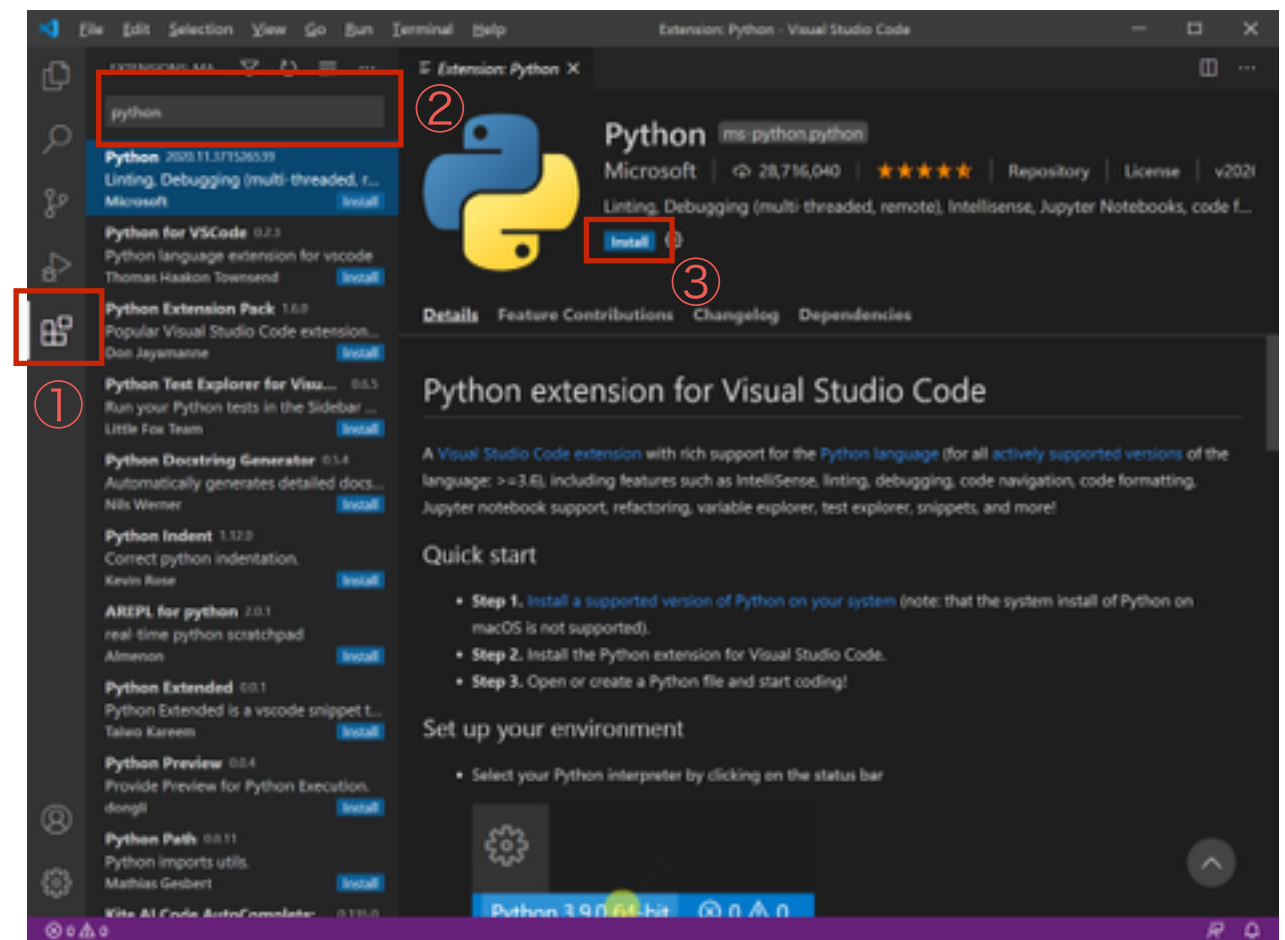


# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

# Visual Studio Code用の拡張機能をインストール

- ・ 拡張機能ビューを開く
  - 「ビュー」 → 「拡張機能」
- ・ 拡張機能を検索
  - 検索ボックスに「python」
  - Microsoftが提供している拡張機能を選択
  - installボタンをクリック

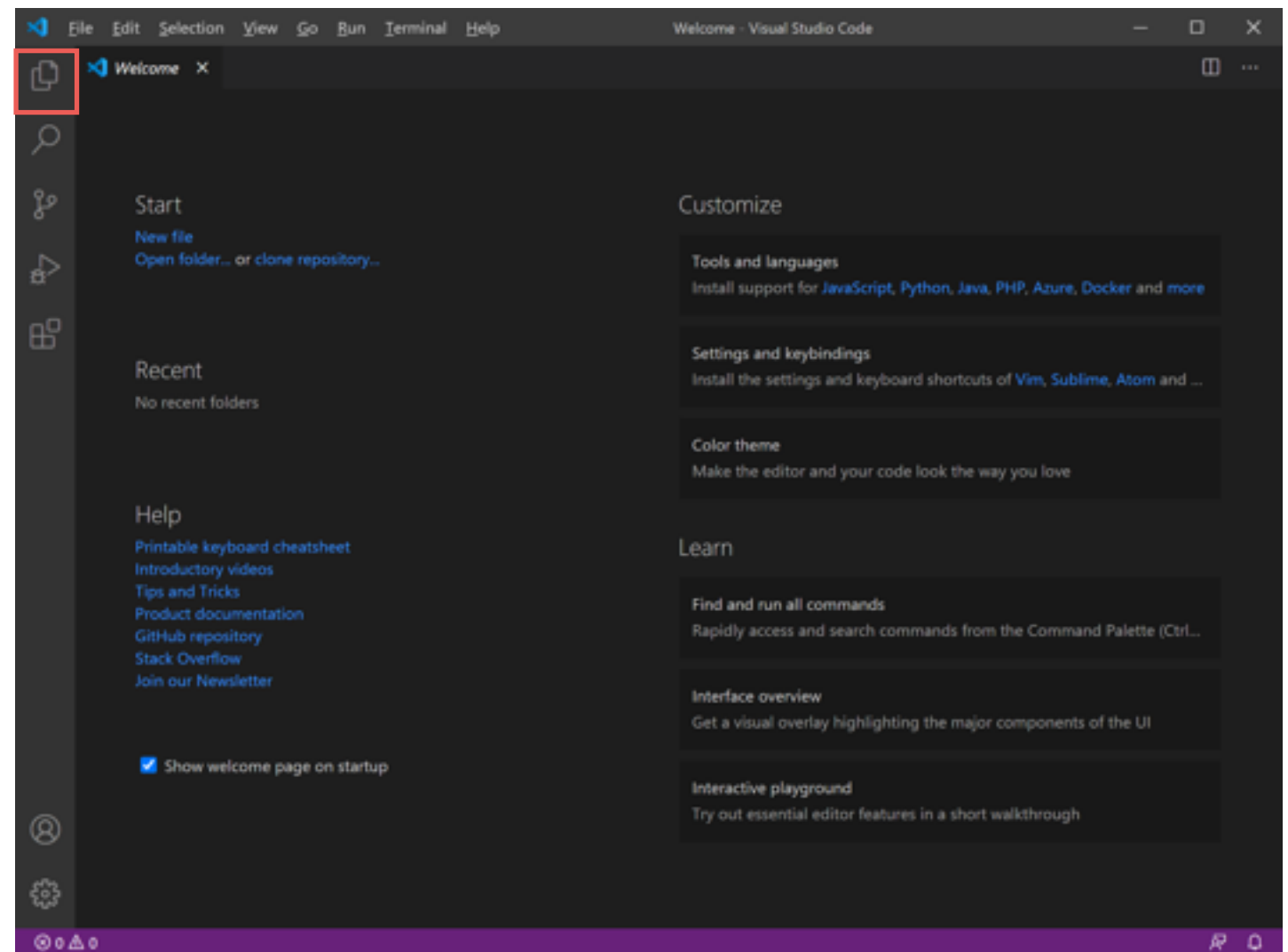


# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

# フォルダを作成

- ・ Explorerアイコンをクリック
- ・ 「Open Folder」 をクリック
- ・ 「PC」 > 「ドキュメント」  
の配下で「新しいフォルダー」  
をクリック
- ・ 「RegioSample」を作成
- ・ 「フォルダーの選択」

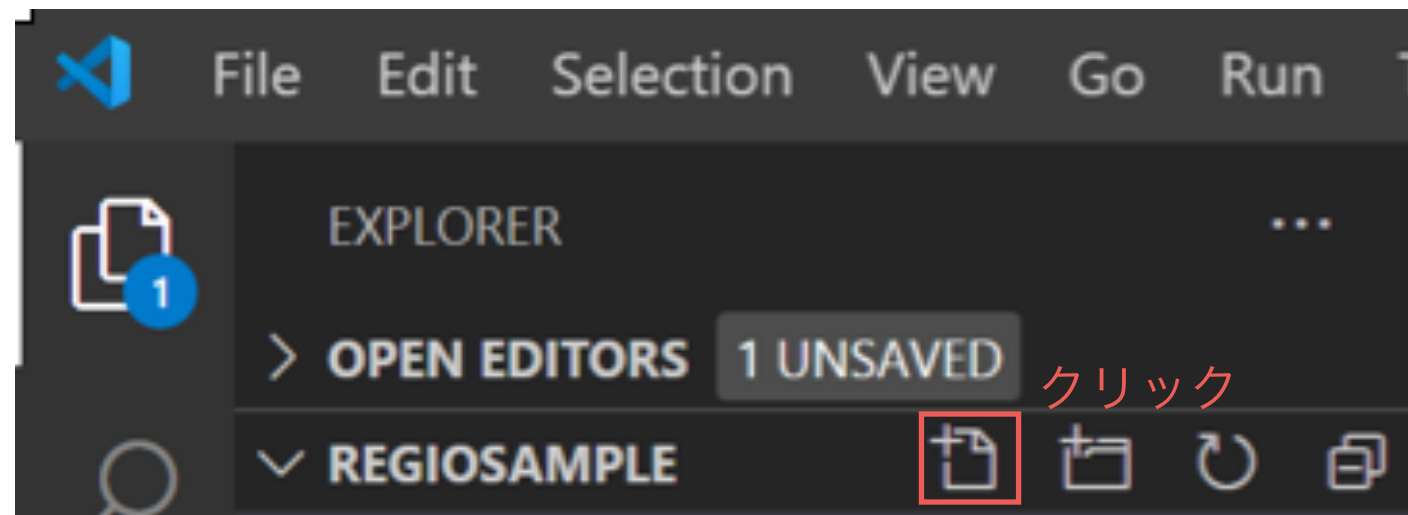


# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

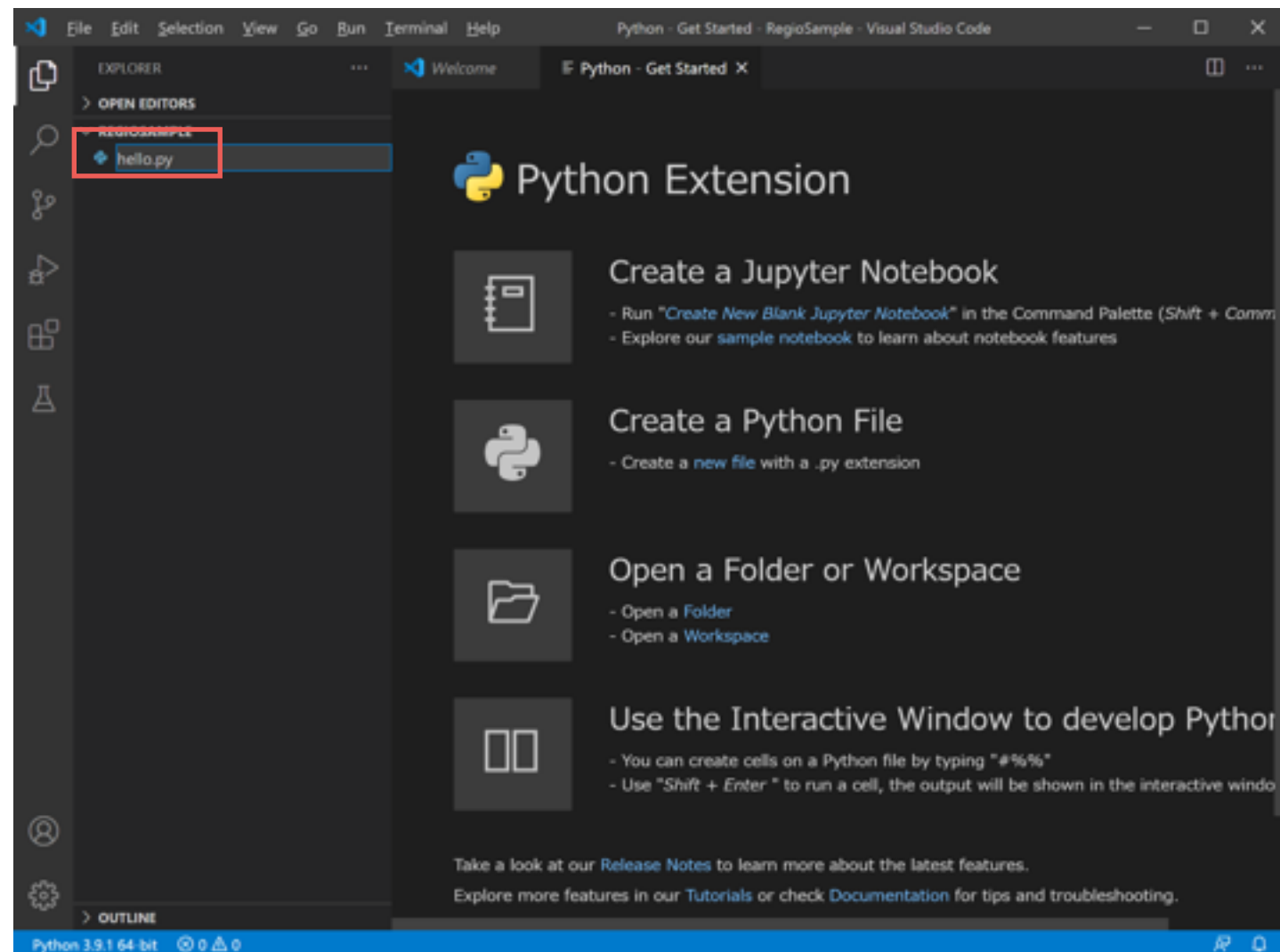
# ファイルを作成

- ・ 「REGIOSAMPLE」 にカーソルを当てる
- ・ 一番左のアイコン 「New File」 をクリック



# ファイルを作成

- ・ 入力窓に「hello.py」を入力



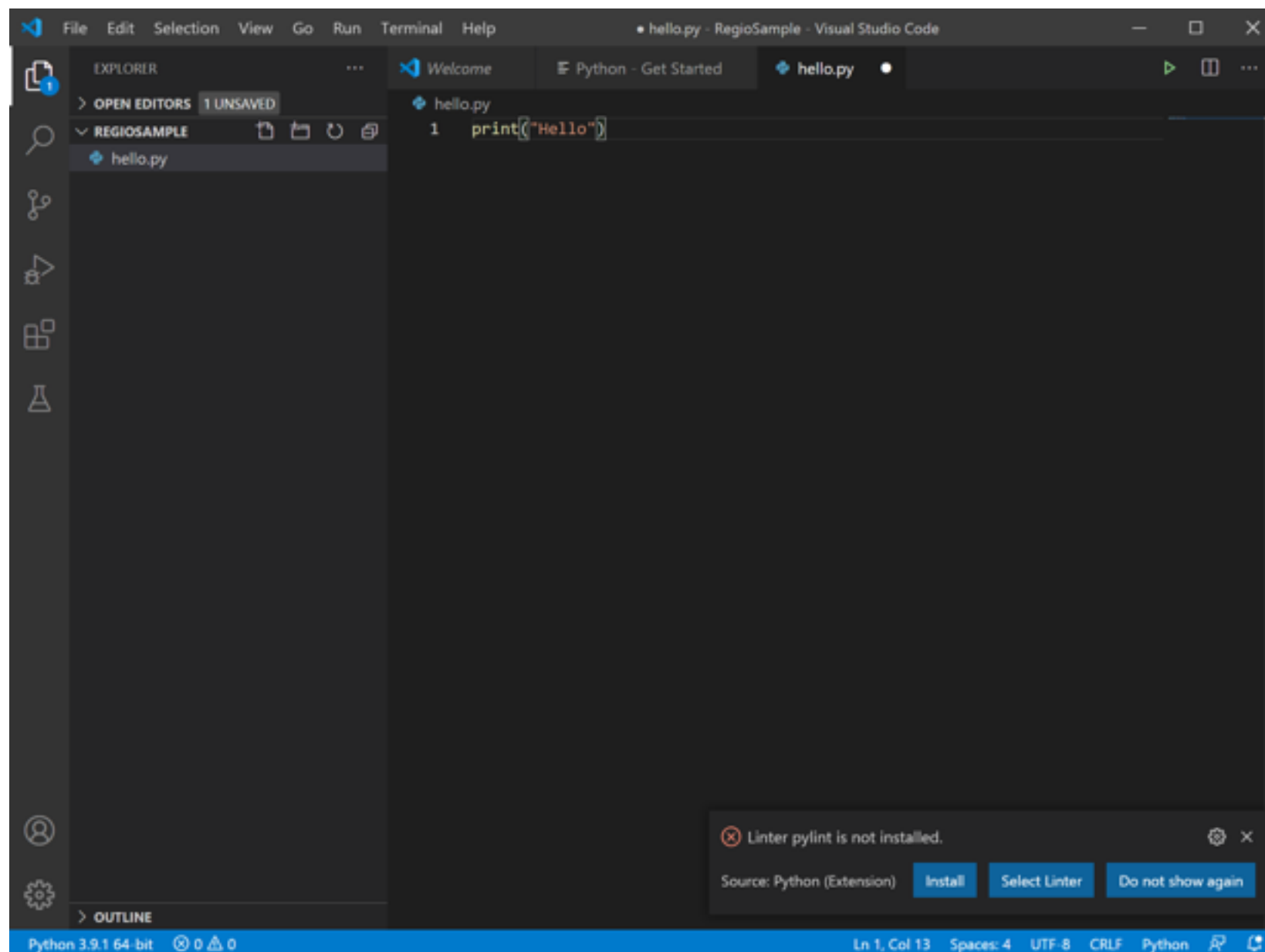
# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行



# コードを記述

- `print("Hello")`

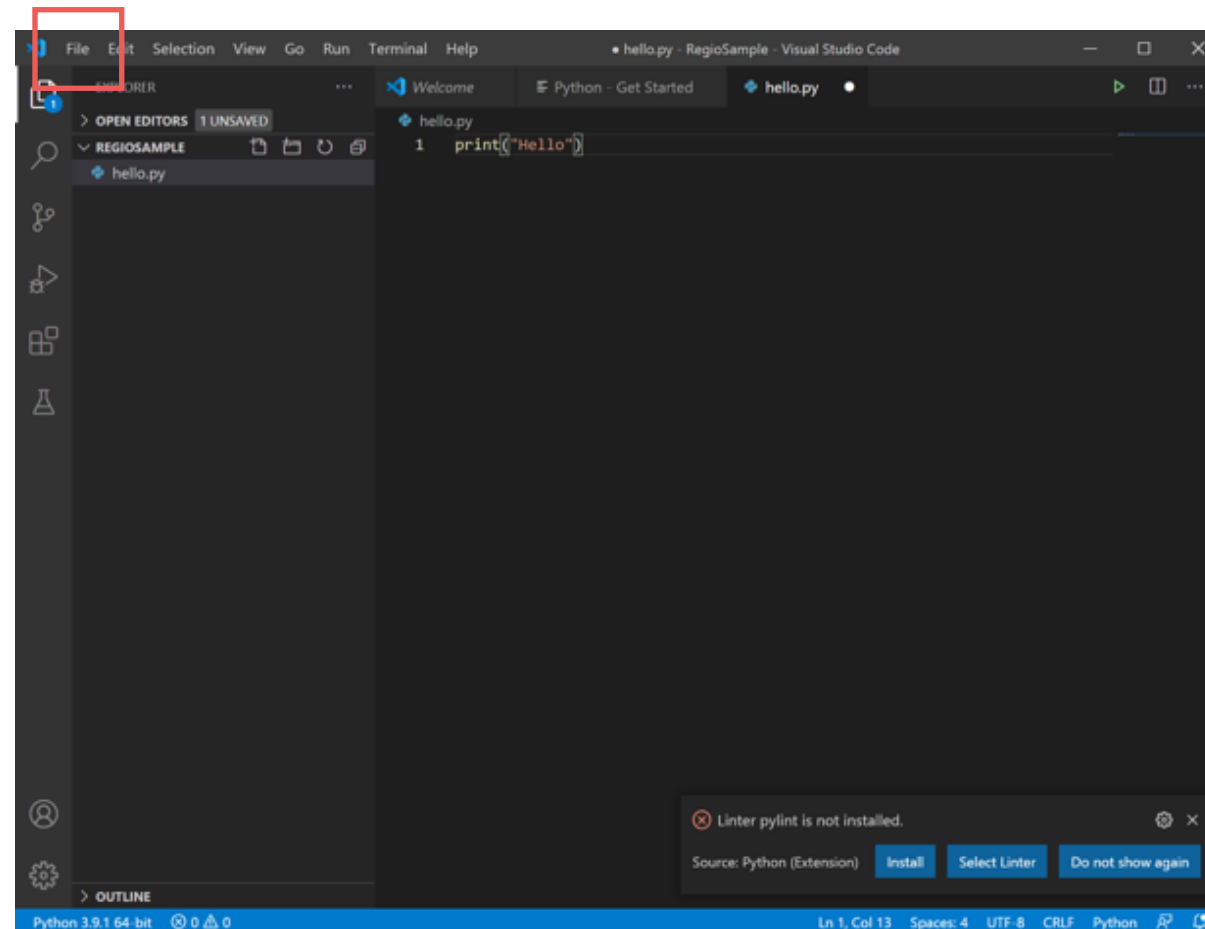


# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

# コードを保存

- ・ 「File」メニューの「Save」
- ・ またはショートカットキーのCtrl + s

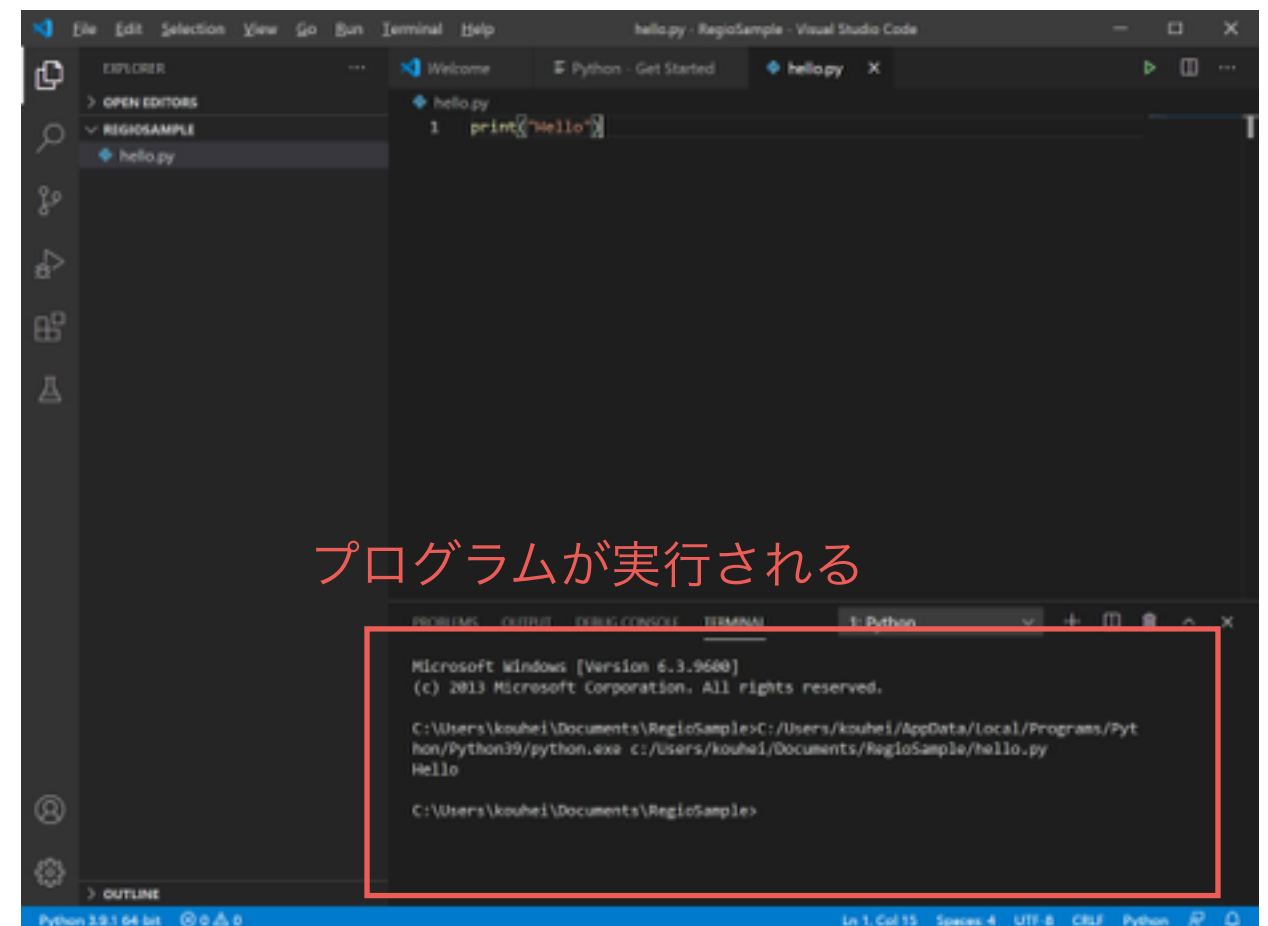
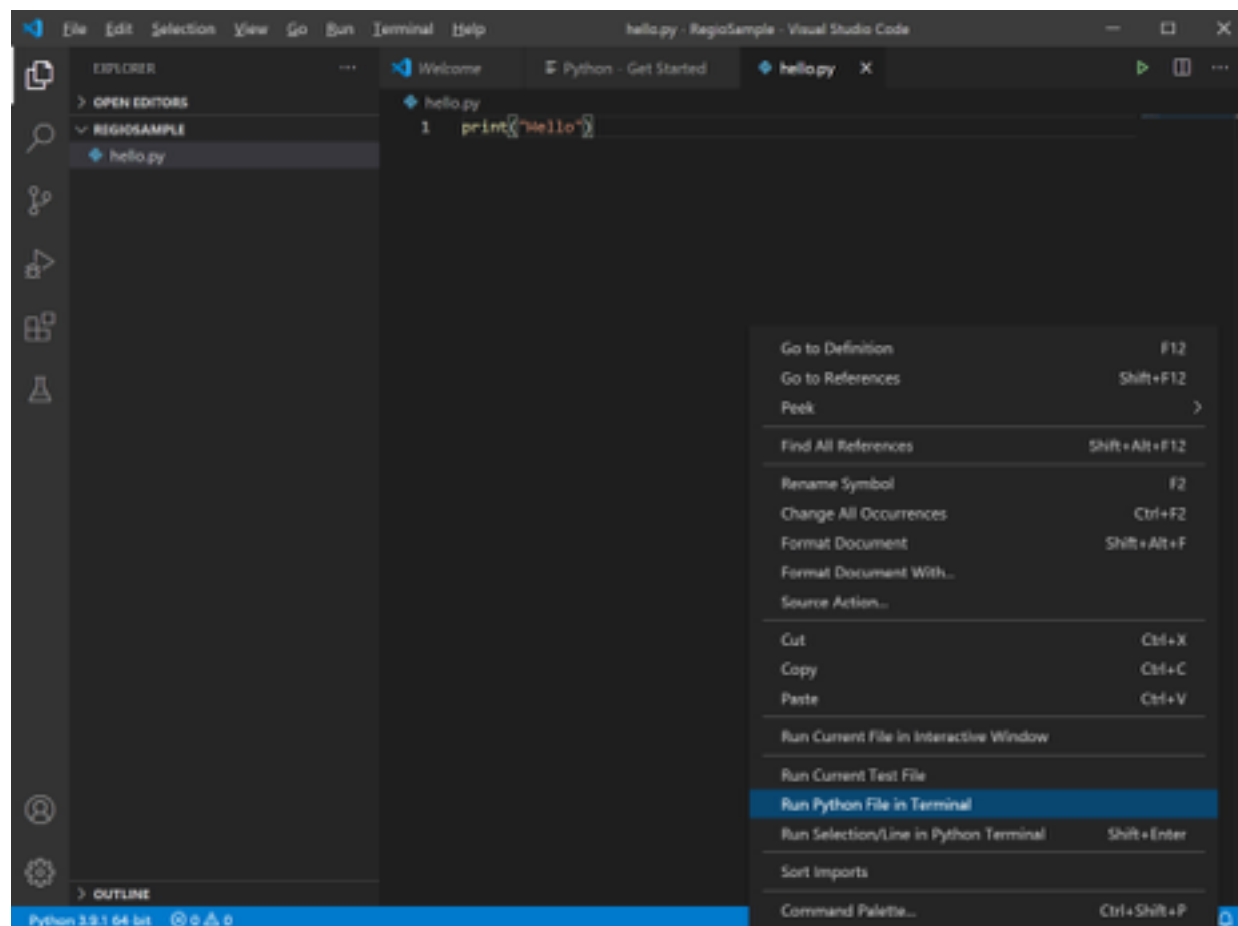


# 環境構築

1. Pythonをインストール
2. 対話モードを利用
3. Visual Studio Codeをインストール
4. Visual Studio Code用の拡張機能をインストール
5. フォルダを作成
6. ファイルを作成
7. コードを記述
8. コードを保存
9. コードを実行

# コードを実行

- ・ エディタを右クリック→Run Python File in Terminal
- ・ 環境構築おしまい



# AtCoder紹介

# AtCoderとは？

- ・ 競技プログラミングと呼ばれるコンピュータプログラムのコンテストを行うサービス、運営会社
- ・ 定められた時間内に問題を解き、正答した問題に応じた配点の合計が得点

# AtCoderへの登録

- AtCoderのサイトへ移動→新規登録→必要情報を入力





# コンテストの参加

- ・ 上部のコンテストから参加したいコンテスト→  
コンテスト一覧→参加したいコンテストをクリック  
→参加登録（途中参加OK）
- ・ ABC(AtCoder Beginner Contest)は土曜の21時  
から100分間



# 常設コンテスト

- ・ コンテスト→コンテスト一覧→常設中のコンテスト
- ・ AtCoderやプログラミングの初学
- ・ AtCoder Programming Guide for beginners(APG4b)

The screenshot shows the AtCoder website interface. The top navigation bar includes links for Home, Contest List, Rankings, and Useful Links. The main content area is divided into sections for '現在のコンテスト' (Current Contests) and '過去のコンテスト' (Past Contests). The '過去のコンテスト' section includes a search bar and a list of past contests. The '現在のコンテスト' section is further divided into '常設中のコンテスト' (Ongoing Contests) and '予定されたコンテスト' (Scheduled Contests). The '常設中のコンテスト' section contains a table with the following data:

コンテスト名	Rated対象
practice contest	-
C++入門 AtCoder Programming Guide for beginners (APG4b)	-
AtCoder Beginners Selection	-
AtCoder Library Practice Contest	-

The '予定されたコンテスト' section contains a table with the following data:

開始時刻	コンテスト名	時間	Rated対象
2020-12-17(木) 00:00	Hitachi Hokudai Lab. & Hokkaido University Contest 2020	743.59	-
2020-12-18(金) 21:00	パナソニックプログラミングコンテスト (AtCoder Beginner Contest 186)	01:40	-1999
2020-12-24(木) 19:00	Xmas Contest 2020	04:00	-
2020-12-26(土) 21:00	AtCoder Grand Contest 050 (Good Bye rrg, 5th Day 1)	04:00	2000+
2020-12-27(日) 21:00	AtCoder Grand Contest 051 (Good Bye rrg, 5th Day 2)	04:00	2000+

# 常設コンテスト

- ・ APG4b
  - ・ AtCoder Programing Guide for beginner
  - ・ 「プログラミングの基礎を学ぶ」ことを目的
- ・ C++の解説
- ・ 有志によるPythonの解説  
(Python APG4bでWeb検索)

# APG4b

- ・ APG4b b-1.01のEx1-コードテストと出力の練習
- ・ APG4b→参加登録ボタンをクリック
- ・ 問題→出力とコメント→ページ下部の「EX1.コードテストと出力の練習」



# APG4b

- ・ Visual Studio Codeを開く→新規ファイル作成→コード記述
- ・ Visual Studio Codeで実行
- ・ コードテスト
- ・ 提出



# APG4b

- ・ コードテスト
- ・ 言語をPython
- ・ ファイルを開く → PC → ドキュメント → RegioSample → ソースコード
- ・ ページ下の実行ボタンをクリック
- ・ 標準出力を確認



# APG4b

- ・ 提出
- ・ 「問題」 → 「1.01.出力とコメント」 → 「Ex1.コードテストと出力の練習」



# APG4b

- ・ 提出結果が表示
- ・ 結果がAccepted(AC)なら正解
- ・ わからなかったら解答例や「Python APG4b」検索して有志による解答例を読んで再挑戦



# 日本情報オリンピック（JOI）の過去問集

- <https://atcoder.jp/contests/archive?category=200>
- コンテスト→コンテスト一覧→カテゴリ内の「JOI過去問」



過去のコンテストを検索

過去のコンテスト

Rated対象

ABCクラス (Rated対象: ~1999)

ARCクラス (Rated対象: ~2799)

AGCクラス (Rated対象: All)

カテゴリ

全て

AtCoder Typical Contest

FAST過去問

非公式コンテスト (unrated)

JOI過去問

企業コンテスト本番

企業オープンコンテスト (rated)

企業オープンコンテスト (unrated)

マラソン

開始時刻	コンテスト名	時間	Rated対象
2020-12-13(日) 13:00	JOI 2020/2021 二次予選 過去問	03:00	-
2020-11-21(土) 14:00	JOI 2020/2021 一次予選 (第3回) 過去問	01:20	-
2020-10-18(日) 14:00	JOI 2020/2021 一次予選 (第2回) 過去問	01:20	-
2020-09-19(土) 14:00	JOI 2020/2021 一次予選 (第1回) 過去問	01:20	-
2020-03-20(金) 09:00	JOI 2019/2020 春合宿 過去問	00:00	-
2020-02-09(日) 09:00	JOI 2019/2020 本選 過去問	04:00	-
2019-12-08(日) 13:00	JOI 2019/2020 二次予選 過去問	03:00	-
2019-11-16(土) 14:00	JOI 2019/2020 一次予選 (第3回) 過去問	01:40	-
2019-10-27(日) 14:00	JOI 2019/2020 一次予選 (第2回) 過去問	01:40	-
2019-09-21(土) 14:00	JOI 2019/2020 一次予選 (第1回) 過去問	01:40	-
2019-03-20(水) 09:00	JOI 2018/2019 春合宿 過去問	00:00	-
2019-02-10(日) 09:00	JOI 2018/2019 本選 過去問	04:00	-

# 環境構築と競技プログラミングサイト紹介

## おわりに

- ・ Pythonを各自のPC上（ローカル）で実行
- ・ Visual Studio Codeを利用してプログラムを作成
- ・ AtCoderを知りAPG4bでコードを提出

プログラミングを行うための準備が完了

楽しくプログラミングに関わってもらえたら嬉しいです