

インソースセミナー



Python学院

～Webアプリケーション開発編／Djangoとデータベースの連携を学ぶ(3日間)

【事前準備手順書】

※ダウンロードいただく研修資料は、下記2点です

- ①事前準備手順書（本資料です）
→事前準備の手順書です。こちらの手順に沿って環境設定をお願いします。
- ②補足資料(zipファイル)
→本研修の主題ではない部分の資料です。
(HTMLとCSSの説明、打ち込むのが面倒な文章など)
zip形式で圧縮されておりますので、**解凍の上、デスクトップ上に張り付けをお願いします。**

0. 研修の事前準備について

本研修では皆様にPythonのプログラミングを行っていただくため、事前準備をお願いしています。本資料で事前準備の方法を説明しておりますので、手順に沿って**必ず**事前準備をお願いします。

■以下リンクから補足資料のダウンロードをお願いいたします。

<https://www.insource.co.jp/resource/office/noindex/python-django.zip>


■事前環境設定一覧

ページ	準備物	所要時間	完了☑
P2 ~P3	1. Pythonのインストール	約3分	
P4 ~P7	2. Visual Studio Codeのインストール	約3分	
P8	3. フォルダの作成	約3分	
P9 ~10	4. 仮想環境の構築	約3分	
P11 ~P13	5. 仮想環境内でDjangoのデフォルト画面を表示する	約3分	

また、事前準備中にエラーが発生した場合、下記のQ&Aサイトをご一読ください。

<https://www.insource.co.jp/python-gakuin/error-faq/index.html>

エラーカテゴリ検索から、エラーがどのアプリケーション上で発生しているのかをご確認の上、エラー解決をお願いいたします。

 エラーカテゴリ検索

頻出エラー
一覧

コマンドプロンプト
に表示されるエラー

Jupyter notebook上で
表示されるエラー

Visual Studio Code
に関するエラー

その他

1. Pythonのインストール(3.10.10)

(1) Python3.10.10 64bit版のダウンロード

■「Windows installer (64-bit)」を下記 URL よりダウンロード ※Python のバージョンに注意
 URL : <https://www.python.org/downloads/release/python-31010/>

<実際の手順>

- ①ブラウザを起動し上記 URL にアクセス
- ②ブラウザ上でキーボードの「Ctrl」+「F」を同時に押し、検索ウィンドウを表示する



出所 : python.org「Python 3.10.10」
<https://www.python.org/downloads/release/python-31010/>
 (最終アクセス日 : 2023年8月18日)

- ③検索ウィンドウに「Windows installer (64-bit)」と入力し、「Enter」キーを押す
- ④検索して表示されたリンクをクリック
- ⑤ファイル名「python-3.10.10-amd64 .exe」がダウンロードされます。

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		6dbe644dd1a520d9853cf6648084c346	26071329	SIG	CRT SIG
XZ compressed source tarball	Source release		7bf85df71bbe7f95e5370b983e6ae684	19627028	SIG	CRT SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	892634724ab799569b512082c8f48c83	41005648	SIG	CRT SIG
Windows embeddable package (32-bit)	Windows		a681a7f9b242fe35b4d96d79e15e57d6	7663448	SIG	CRT SIG
Windows embeddable package (64-bit)	Windows		f38a9e7e02a992daa62569b758d0a388	8625602	SIG	CRT SIG
Windows help file	Windows		448f8401ade49a7e2156d02512f2f9bf	9391521	SIG	CRT SIG
Windows installer (32-bit)	Windows		a81b81687bc2575c05a30f4b31d6ea00	27859200	SIG	CRT SIG
Windows installer (64-bit)	Windows	Recommended	9735797853cba809b13c8396c91354a0	29010904	SIG	CRT SIG

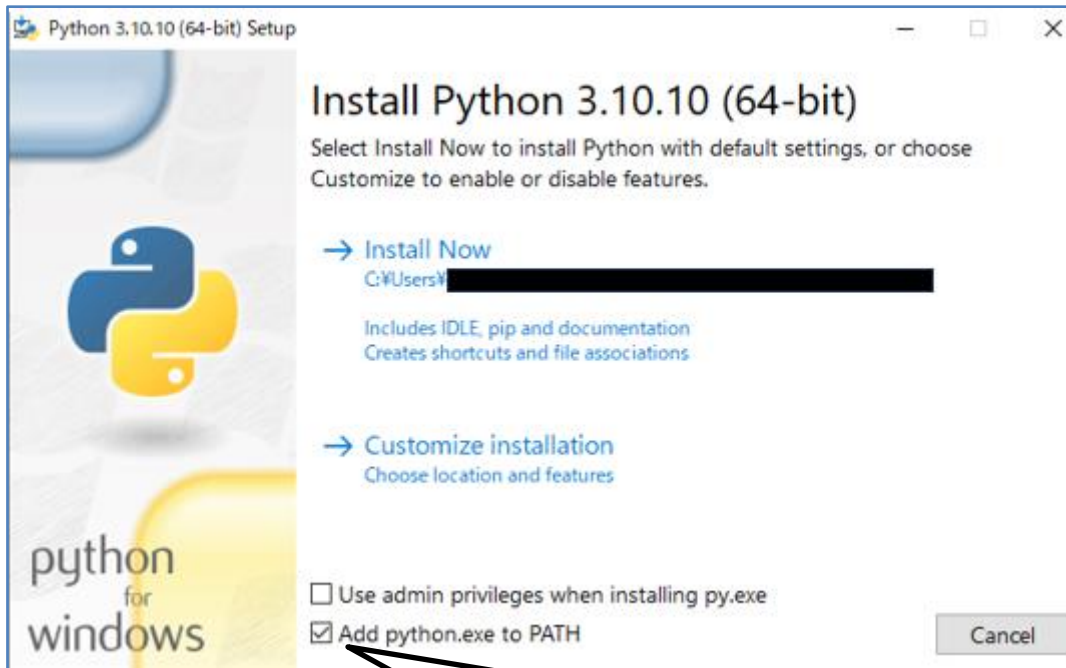
出所 : python.org「Python 3.10.10」
<https://www.python.org/downloads/release/python-31010/>
 (最終アクセス日 : 2023年8月18日)

(2) Python3.10.10 64bit版のインストール

■ダウンロードしたファイル「python-3.10.10-amd64 .exe」を実行し、下記の手順でインストールする。

<インストール手順>

- ① 「python-3.10.10-amd64 .exe」をダブルクリックする
- ② 下部のチェック項目「Install launcher for all users」のチェックを外す
- ③ 下部にある「Add Python.exe to PATH」にチェックを入れる(環境変数の設定)
- ④ 「Install Now」をクリック
- ⑤ 「Setup was successful」の文字が出たら「Close」ボタンを押して完了



「Use admin privileges when installing py.exe」→チェック外す
 「Add Python.exe to PATH」→チェック入れる

2. Visual Studio Codeのインストール

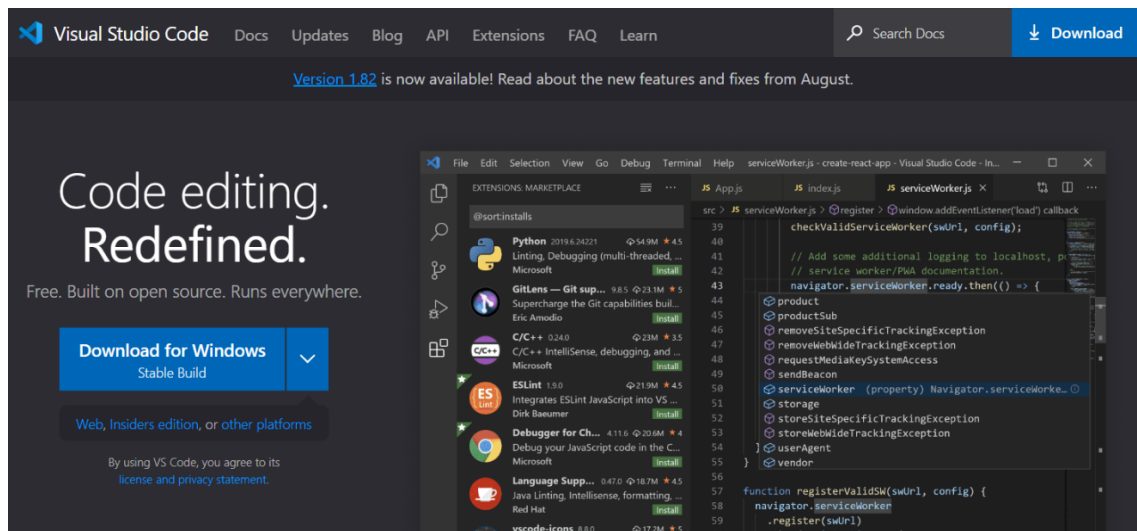
(1) Visual Studio Codeのダウンロード&インストール

■下記ページよりダウンロードしインストール

URL:<https://code.visualstudio.com/>

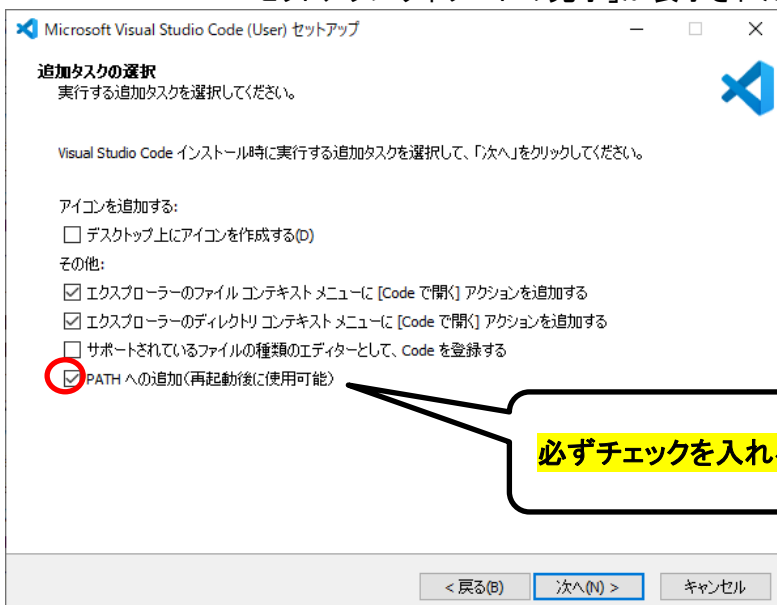
<実際の手順>

- ①ブラウザを起動し上記URLにアクセス
- ②「Download for Windows Stable Build」をクリックしインストーラーをダウンロード



- ③ダウンロードされた「VSCodeUserSetup-x64-XXXX.exe」をクリック
※”XXXX”に入る数字は現在の最新バージョンの数字が入ります。
- ④インストール画面が開くので「同意する(A)」にチェックを入れ「次へ(N)>」をクリック
- ⑤「追加タスクの選択」の画面にて、「PATHへの追加」にチェックを入れ、「次へ(N)>」をクリック
- ⑥「インストール(I)」をクリックし、

「Visual Studio Codeセットアップウィザードの完了」が表示されたら完了

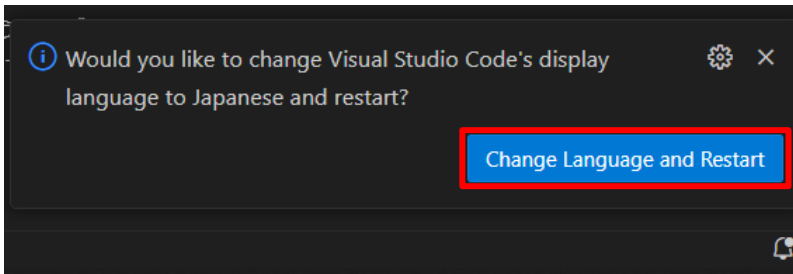


(2) Visual Studio Codeを使用する準備

- Japanese Language Pack for Visual Studio Code(拡張機能)のインストール
- 「Python(拡張機能)」のインストール

<実際の手順>

- ① インストールしたVisual Studio Codeを起動
- ② キーボードの「Ctrl」、「Shift」、「X」キーを同時に入力し、左側に「EXTENSIONS」というメニューを出現させる
- ③ 入力欄に「Japanese Language Pack for Visual Studio Code」と入力
- ④ 「Japanese Language Pack for Visual Studio Code」を選択し、「install」ボタンをクリック
- ⑤ 再度キーボードの「Ctrl」、「Shift」、「X」キーを同時に入力し、入力欄に「python」と入力する
- ⑥ 「Python」を選択し、「install(インストール)」ボタンをクリック
- ⑦ 以下の赤枠のボタンを押し日本語化する。またはVisual Studio Codeを再起動する



再起動は、「Ctrl」、「Shift」、「P」キーを同時に入力し、入力欄に「reload window」を入力する

- ⑧ 日本語化されているのを確認したのち、下記の(3)Pythonプログラムの動作確認を行う



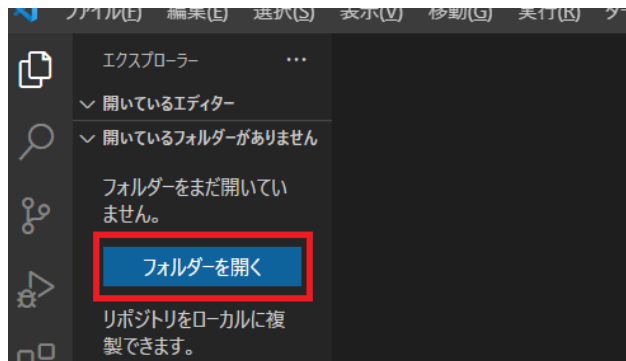
Visual Studio Code起動したら、キーボードの「Ctrl」+「Shift」+「X」を同時に押し、「EXTENSIONS」というメニューを出現させる

入力欄に「Japanese Language Pack for Visual Studio Code」と入力&選択
Japanese Language Pack for Visual Studio Codeのページが開いたら、「Install」ボタンをクリック

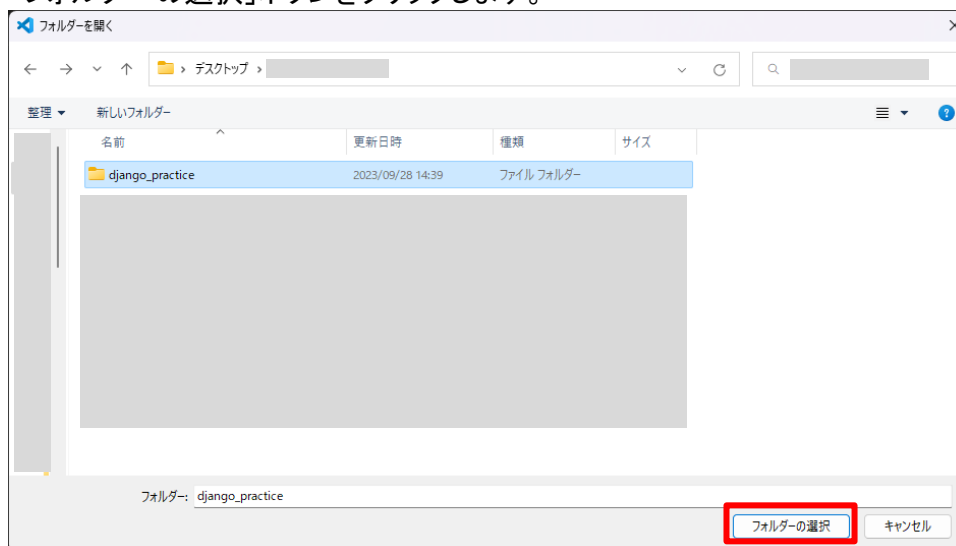
(3) Pythonプログラムの動作確認

Visual Studio Codeのエクスプローラー上にフォルダを設定する必要があります。
このフォルダの中に、Pythonのプログラムファイルを作成していきます。
フォルダは、お送りした圧縮ファイルを解凍した際に、当ファイルと同じフォルダ内に
格納されていた「**django_practice**」フォルダを設定します。

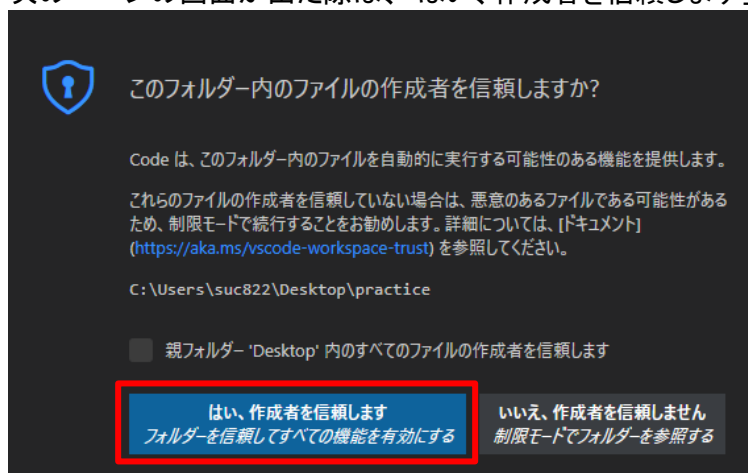
- ①「**django_practice**」フォルダをデスクトップ上に配置します。
- ②下記「エクスプローラー」内の「フォルダーを開く」ボタンをクリックします。



フォルダ選択ダイアログが表示されますので、「**django_practice**」フォルダを選択し、「フォルダーの選択」ボタンをクリックします。



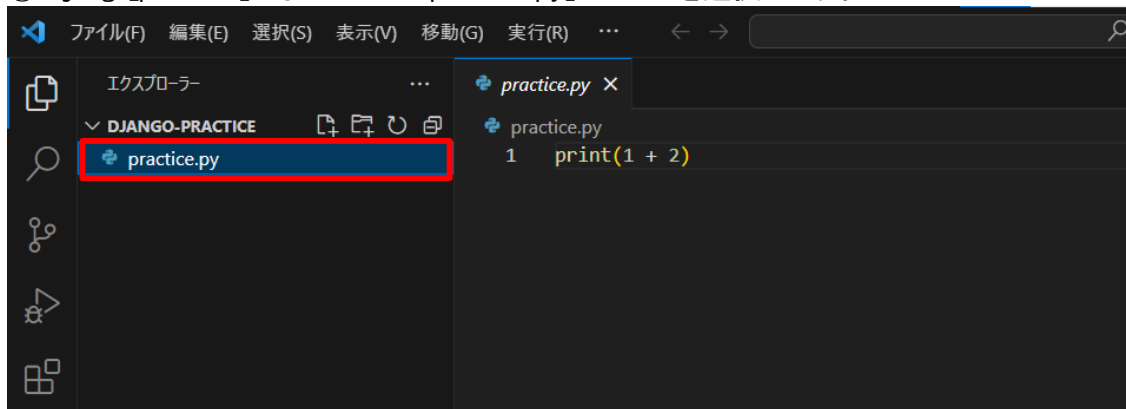
次のページの画面が出た際は、「はい、作成者を信頼します」を選択します。



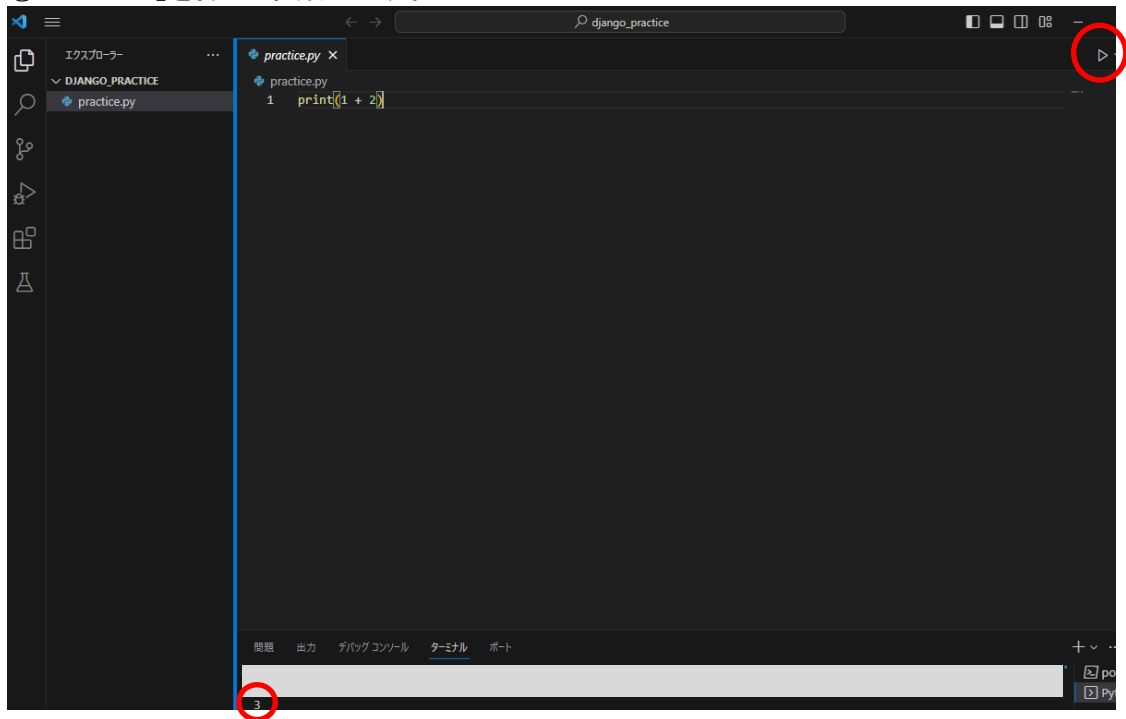
これで、Visual Studio Code上で「django_practice」フォルダの設定が完了です。

続いて、Visual Studio Codeを使ってPythonのファイルを起動させます。

③「django_practice」フォルダ内の「practice.py」ファイルを選択します。



④右上の「▶」を押して実行します。



⑤画面下部に「3」と表示されればVisual Studio Code上での確認は終了です。

実行結果が上手く表示されない場合は、下記のQ&Aサイト「[Visual Studio Codeに関するエラー](https://www.insource.co.jp/python-gakuin/error-faq/index.html)」からご解決をお願いします。

URL:<https://www.insource.co.jp/python-gakuin/error-faq/index.html>

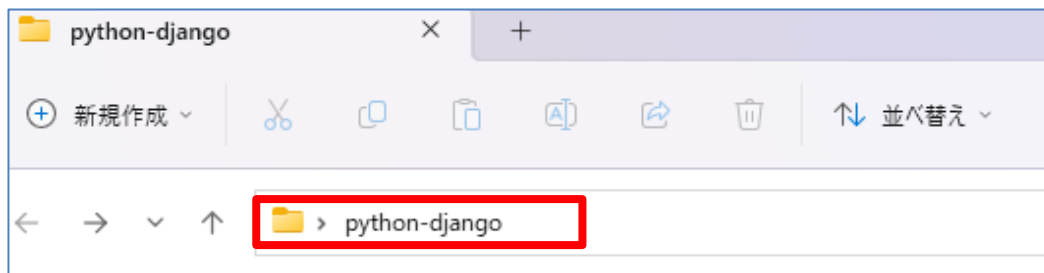
3. フォルダの作成

この章で作成するフォルダの内容は、以下のリンクよりダウンロード可能です。

- ① project2: https://www.insource.co.jp/resource/office/noindex/python_application_project2.zip
- ② project3: https://www.insource.co.jp/resource/office/noindex/python_application_project3.zip

(1) 「python-django」フォルダからコマンドプロンプトを開く

- ① 3で解凍した「python-django」フォルダ内で「cmd」と入力する



上の画像の赤枠を押し、「cmd」と入力すると、コマンドプロンプトが開きます。

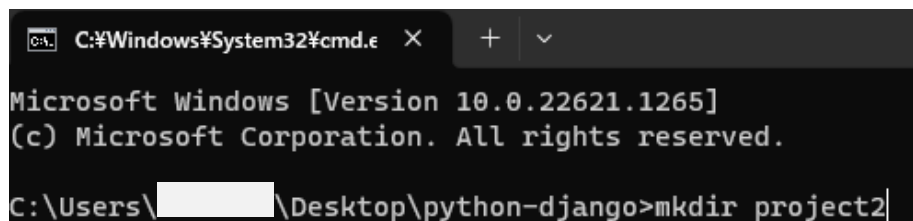
(2) 「project2」フォルダを作成する

- (1)で開いたコマンドプロンプトに以下のコードを打ち込む

```
mkdir project2
```

「mkdir」で「project2」フォルダを作成できます。

※「project2」フォルダは、テキストの第3章で使用します。



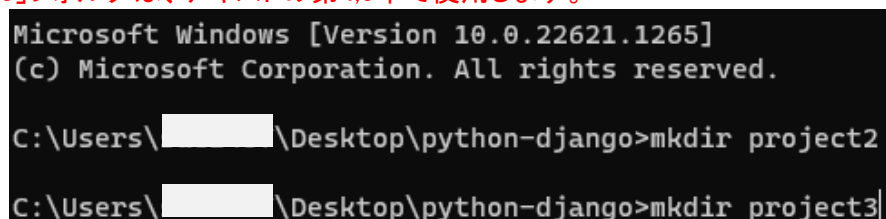
(3) 「project3」フォルダを作成する

- (1)で開いたコマンドプロンプトに以下のコードを打ち込む

```
mkdir project3
```

「mkdir」で「project3」フォルダを作成できます。

※「project3」フォルダは、テキストの第4,5章で使用します。



4. 仮想環境の構築

(1)「project3」フォルダの仮想環境を構築する

①「python-django」フォルダから「project3」フォルダに移動する

```
cd project3
```

「cd」でフォルダ内を移動することができます。
今回の場合は「python-django」から「project3」に移動できます。

②「project3」フォルダで仮想環境を作成する

```
python -m venv venv
```

このコードを入力し、仮想環境を作成します。
少し待つと「C:¥Users¥●●●●¥Desktop¥python-django¥project3」のようなパスが出てきます。
パスが出たら、(2)に進みます。

```
C:\Users\██████\Desktop\python-django>cd project3
C:\Users\██████\Desktop\python-django\project3>python -m venv venv
C:\Users\██████\Desktop\python-django\project3>
```

(2)「project2」フォルダの仮想環境を構築する

①「project3」フォルダから「project2」フォルダに移動する

```
cd ../
```

「cd ../」で1つ前のフォルダに戻ることができます。
まずは「project3」フォルダから「python-django」フォルダに移動します。

```
cd project2
```

(1)の①同様、「python-django」から「project2」に移動します。

②仮想環境を作成する

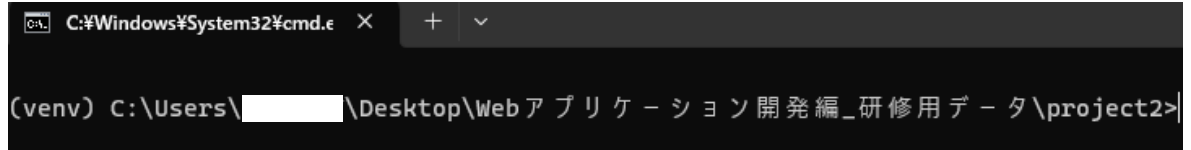
```
python -m venv venv
```

(1)の②同様、仮想環境を作成します。
少し待つとパスが出てきます。パスが出たら(2)に進みます。

③仮想環境に切り替える

```
venv¥Scripts¥activate.bat
```

上記のコードを入力すると、仮想環境内でプログラミングを行うことができるようになります。
仮想環境を使用中は次のページのように「(venv)」と出てきます。



```
C:\Windows\System32\cmd.e × + v  
(venv) C:\Users\██████████\Desktop\Webアプリケーション開発編_研修用データ\project2>
```

5. 仮想環境内でDjangoのデフォルト画面を表示する

この章ではproject2フォルダ内で、以下の2つを行います。

- ・仮想環境内にライブラリ(便利機能のようなもの)を入れる
- ・Webサーバーを立ち上げ、Djangoのデフォルト画面を表示する

※project3フォルダへのライブラリ追加は研修時間中に行います。

(1) 仮想環境内でDjangoをインストールする

以下のコードを入力し、ライブラリ(django)をインストールする

```
pip install django==3.2
```

以下のように「Successfully」と出ている場合は、問題なくインストールできています。

※黄色の文字で「WARNING」と出ることもございますが、エラーではないためご安心ください。

```
Downloading Django-3.2-py3-none-any.whl (7.9 MB)
|#####| 7.9 MB ...
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
|#####| 41 kB ...
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting pytz
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
|#####| 502 kB ...
Collecting typing-extensions>=4; python_version < "3.11"
  Downloading typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Installing collected packages: sqlparse, typing-extensions, asgiref, pytz, django
Successfully installed asgiref-3.7.2 django-3.2 pytz-2023.3.post1 sqlparse-0.4.4 typing-extensions-4.8.0
WARNING: you are using pip version 20.1.1; however, version 23.2.1 is available.
You should consider upgrading via the 'c:\users\suc1562\desktop\python-django\project2\venv\scripts\python.exe -m pip install --upgrade pip' command.
```

(2) プロジェクトを開始する

「startproject」というコマンドを使用する

※「startproject」というコマンドにつきましては、研修内で詳細をご説明いたします

```
django-admin startproject helloworldproject
```

(3) Webサーバーを立ち上げる

①以下のコード2つを入力する

```
cd helloworldproject
```

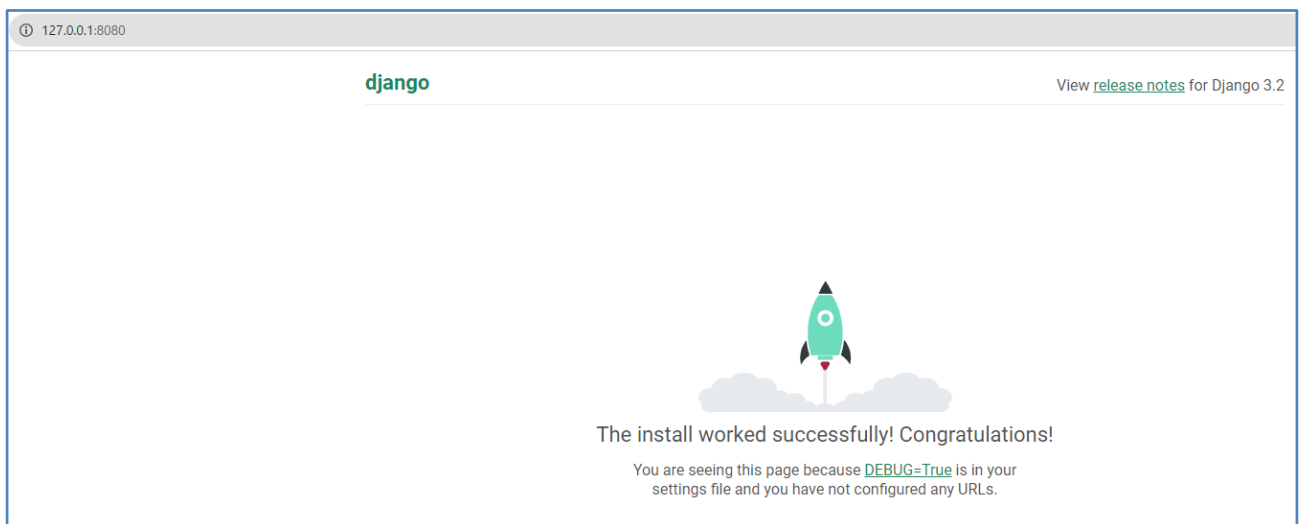
```
python manage.py runserver
```

- ②「http://127.0.0.1:●●●●/」というURLをWebブラウザに入力する
①のコードを入力すると、以下の画像と同様の文字列が出力されます。

```
You have 18 unapplied migration(s). Your project may not work pro
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
August 25, 2023 - 18:11:57
Django version 3.2, using settings 'helloworldproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

この文字列の中の「http://127.0.0.1: ●●●●/」というURLをブラウザに入力してください。
上記は「http://127.0.0.1:8000/」というURLになっておりますが、他の番号の可能性もございます。

入力しエンターキーを押すと、次のページの画面が出てきます。
これがDjangoのデフォルト画面となります。



※もし、「http://127.0.0.1:8000/」というURLでDjangoのデフォルト画面が出なかった場合は、
①の「python manage.py runserver」を「python manage.py runserver 8080」に変更して
お試しくださいと幸いです。

(4) Webサーバーを終了する

コマンドプロンプトに「ctrl」+「pause」か「ctrl」+「C」を押すとサーバーが終了します。


(5) 仮想環境を終了する

以下のコードを実行する

```
deactivate
```

エラーで事前準備が完了しなかった場合、下記のQ&Aサイトをご一読ください。
<https://www.insource.co.jp/python-gakuin/error-faq/index.html>

エラーカテゴリ検索から、エラーがどのアプリケーション上で発生しているのかをご確認の上、エラー解決をお願いいたします。

 エラーカテゴリ検索

頻出エラー 一覧	コマンドプロンプト に表示されるエラー	Jupyter notebook上で 表示されるエラー
Visual Studio Code に関するエラー	その他	