



Dirbato

10

Summary: このドキュメントは、*Dirbato @ 42 Tokyo*の10モジュール用の課題である。

Contents

I	Instructions	2
II	Foreword	3
III	Mandatory Exercise 00 : FastAPI	4
IV	Mandatory Exercise 01 : Front H1	5
V	Mandatory Exercise 02 : Front Form	6
VI	Mandatory Exercise 03 : Front Form label and input	7
VII	Mandatory Exercise 04 : Front + FastAPI	8
VIII	Mandatory Exercise 05 : Backend for readfortune	9
IX	Mandatory Exercise 06 : Backend for readfortune + date	10
X	Mandatory Exercise 07 : Backend for readfortune + requests + PokeAPI	11
XI	Mandatory Exercise 08 : Front for readfortune	12
XII	Bonus Exercise 09 : Improve the website	13

Chapter I

Instructions

- 課題の確認と評価は、あなたの周りにいる受講者により行われる。
- 問題は、簡単なものから徐々に難しくなるように並べられている。
- 質問がある場合は、隣の人に聞くこと。それでも分からない場合は、隣の隣の人に聞くこと。
- 助けてくれるのは、Google / 人間 / インターネット / ...と呼ばれているものたちである。
- Mandatory Exerciseの問題までは可能な部分まで取り組むこと。
- Bonus Exerciseの問題は時間に余裕がある場合、取り組むこと。
- 出力例には、問題文に明記されていない細部まで表示されている場合があるため、入念に確認すること。
- 各問題でPythonのバージョンの指定がない場合は、次のバージョンを使用すること。: Python - python3.9.0
- 課題は、プロジェクトページのGit リポジトリに提出すること。リポジトリ内の提出物のみが、レビュー中の評価対象となる。提出ディレクトリやファイルの名前が正しいことを確認すること。

Chapter II


Foreword

今回の研修を通して獲得したスキルやコンセプトのまとめとして、
「今日のポケモン占い」アプリケーションを開発してもらいます！

以下、ご参考までに獲得したスキルやコンセプト：
エンジニアリングのマインドセット、
課題解決の手法（ググる、人に聞くなど）、
シェル、シェルスクリプト、コマンド、git、プログラミング、
テキストエディタ（vim, Visual Studio Code）、
Python、オブジェクト指向、標準パッケージ（os, arg）、
外部パッケージ（discordbot, requests, fastapi, PIL）、
pip、API、JSON、HTTPメソッド、HTTPステータスコード、
HTML/CSSなど

Chapter III

Mandatory Exercise 00 : FastAPI

	Exercise 00
Installing FastAPI	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

- `venv`を使用して、仮想環境を構築すること。

```
?>python3 -m venv myenv
?>source myenv/bin/activate
```

- `pip`を使用して、`fastapi`と`uvicorn`をインストールすること。

```
?>python3 -m pip install fastapi 'uvicorn[standard]'
```

- テキストエディタ（Visual Studio Code, vimなど）を使用して、以下の内容を含んだ`main.py`を作成すること。

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
def read_root():
    return {"pokemon": "pikachu"}
```

- 以下のコマンドでサーバを立ち上げること。


```
?>python3 -m uvicorn main:app --reload --port 8000
```

- 以下のURLにアクセスした際、`{"pokemon": "pikachu"}`が表示されるか確認すること。

```
http://localhost:8000/
```

Chapter IV

Mandatory Exercise 01 : Front H1

	Exercise 01
	Front
	提出するディレクトリ : <i>ex00/</i>
	提出するファイル : *
	使用可能な標準ライブラリ : *

前問の提出物に機能を追加していきます。プロジェクトページから`html.zip`をダウンロードし、解凍せよ。解凍したフォルダにアクセスし、`ex 01` フォルダから`index.html`のデータを活用し、以下の要件を満たしたファイルを提出せよ。


- `templates`ディレクトリを作成すること。
- `templates`ディレクトリの配下に、解凍した`index.html`を配置すること。
- `h1`タグのコンテンツに今日のポケモン占いを入力すること。



ブラウザのURLバーに`index.html`ファイルをドロップして、表示される内容を確認すること。もしくは、Visual Studio CodeのLive Serverプラグインをインストールすること。

Chapter V

Mandatory Exercise 02 : Front Form

	Exercise 02
Front	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

前問の提出物に機能を追加していきます。プロジェクトページからhtml.zipをダウンロードし、解凍せよ。解凍したフォルダにアクセスし、ex 02 フォルダからindex.htmlのデータを活用し、以下の要件を満たしたファイルを提出せよ。


- h1タグの下にformタグを作成すること。
- formタグのaction属性には、"readfortune"という値を入力すること。
- formタグのmethod属性には、"get"という値を入力すること。



<https://developer.mozilla.org/ja/docs/Web/HTML/Element/form>

Chapter VI

Mandatory Exercise 03 : Front Form label and input

	Exercise 03
Front Form label and input	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

前問の提出物に機能を追加していきます。プロジェクトページから`html.zip`をダウンロードし、解凍せよ。解凍したフォルダにアクセスし、`ex 03` フォルダから`index.html`のデータを活用し、以下の要件を満たしたファイルを提出せよ。


- `form`タグの中に`label`タグと`input`タグと`input`タグを追加すること。
- `label`タグの`for`属性には、`"name"`という値を入力すること。
- `label`タグのコンテンツには、`"あなたの名前"`を入力すること。
- 最初の`input`タグの`type`属性には、`"text"`という値を入力すること。
- 最初の`input`タグの`name`属性には、`"name"`という値を入力すること。
- 最初の`input`タグに`required`属性を追加すること。
- 2つ目の`input`タグの`type`属性には、`"submit"`という値を入力すること。
- 2つ目の`input`タグの`value`属性には、`"占う"`という値を入力すること。



<https://developer.mozilla.org/ja/docs/Web/HTML/Element/form>

Chapter VII

Mandatory Exercise 04 : Front + FastAPI

	Exercise 04
	Front + FastAPI
	提出するディレクトリ : <i>ex00/</i>
	提出するファイル : *
	使用可能な標準ライブラリ : *

前問の提出物に機能を追加していきます。プロジェクトページから`html.zip`をダウンロードし、解凍せよ。解凍したフォルダにアクセスし、`ex 04` フォルダから`index.html`のデータを活用し、以下の要件を満たしたファイルを提出せよ。

- 以下のURLにアクセスした際、`index.html`が表示されるようにコードを修正すること。


```
http://localhost:8000/
```



```
https://fastapi.tiangolo.com/advanced/templates/
```

Chapter VIII

Mandatory Exercise 05 : Backend for readfortune

	Exercise 05
Backend for readfortune	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

前問の提出物をアップデートしていきます。前問の提出物を利用し、以下の要件を満たすコードを提出せよ。


- <http://localhost:8000/>にアクセスしフォームを送信した後に、`{"name":"XXX"}` (XXXにはフォーム送信時に入力した値が入ります) が表示されるようにすること。



<https://fastapi.tiangolo.com/tutorial/query-params/>

Chapter IX

Mandatory Exercise 06 : Backend for readfortune + date

	Exercise 06
Backend for readfortune + date	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

前問の提出物をアップデートしていきます。前問の提出物を利用し、以下の要件を満たすコードを提出せよ。


- <http://localhost:8000/>にアクセスしフォームを送信した後に、{"name": "XXX", "date": "yyyy-MM-dd"} (yyyy-MM-ddにはその日の日付が入ります) が表示されるようにすること。



<https://www.geeksforgeeks.org/get-current-date-using-python/>

Chapter X

Mandatory Exercise 07 : Backend for readfortune + requests + PokeAPI

	Exercise 07
Backend for readfortune + requests + PokeAPI	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

前問の提出物をアップデートしていきます。前問の提出物を利用し、以下の要件を満たすコードを提出せよ。

- <http://localhost:8000/>にアクセスしフォームを送信した後に、{"pokemon":"piakchu", "date":"yyyy-MM-dd"} (yyyy-MM-ddにはその日の日付が入ります) が表示されるようにすること。
- どのpokemonの名前を表示するかに関しては、以下の関数を活用して表示するポケモンのIDをまず取得すること。その後に、そのポケモン番号に付随すポケモンの名前をPokeAPIを活用して取得すること。

```
def get_pokeid(name):
    import datetime

    POKEMON_LENGTH = 1017


    today = datetime.datetime.today()
    name_orcs = [ord(c) for c in name]

    fortune_value = (sum(name_orcs) + today.year + today.month + today.day) % POKEMON_LENGTH
    pokeid = fortune_value if fortune_value != 0 else POKEMON_LENGTH

    return pokeid
```

Chapter XI

Mandatory Exercise 08 : Front for readfortune

	Exercise 08
Front for readfortune	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

前問の提出物に機能を追加していきます。プロジェクトページから`html.zip`をダウンロードし、解凍せよ。解凍したフォルダにアクセスし、`ex 08` フォルダから`readfortune.html`のデータを活用し、以下の要件を満たしたファイルを提出せよ。

- <http://localhost:8000/>にアクセスしフォームを送信した後に、以下のような画面が表示されるようにすること。

← → ↺ ⓘ localhost:8000/readfortune/?name=nop

今日のポケモン占い


2023年11月09日のnopさんのポケモン



crawdaunt

Chapter XII

Bonus Exercise 09 : Improve the website

	Exercise 09
Improve the website	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

前問の提出物をアップデートしていきましょう。以下の内容から好きな機能を実装すること。1つでも実装されている場合、ボーナス点を獲得できます。

- ポケモンの名前を日本語で表示すること。
- 先週の占いを表示すること。
- CSSを活用してウェブサイトを装飾すること。
- ホームに戻るボタンを追加すること。
- 他のアプリケーションと連携しやすくするためにAPIを作成すること（HTMLを返さず、JSONを返す）。
- SEO対策として、metaタグを追加すること。
- ウェブサイトのUI/UXの改善に関すること。