

今日の授業：Webアプリの開発と公開

シラバス	
<p>Webアプリの作成をStreamlitで行うため、Streamlitの操作方法の復習を行う。</p> <ol style="list-style-type: none"> Webアプリの実装 <ul style="list-style-type: none"> Streamlitの基礎 Webアプリの公開 <ul style="list-style-type: none"> GitHubへの公開とStreamlit Cloudへのデプロイ API利用のための設定 <ul style="list-style-type: none"> sercets.tomlやcaching 	
今日の授業	キーポイント
Webアプリの開発と公開	
1. 開発環境の整備	1. 開発環境をcondaの仮想環境として作成し、streamlit開発のためのディレクトリ構成を行います。
2. Webアプリの実装：ユーザー入力の受け取り	2. テキストや日付など、適切な入力widgetを配置してユーザーからの入力を受け取ります。
3. Webアプリの実装：ユーザーへの情報表示	3. テキストや画像など、適切なwidgetを配置してユーザーに対して情報を表示します。
4. Webアプリの実装：サーバー側ファイルの読み込み	4. 開発ディレクトリに設置したファイルはコードの中からアクセスする事ができます。ファイルパスは作業ディレクトリからの相対パスとなることに注意しましょう。
5. Webアプリの実装：Webリソースの取得	5. 外部サイトからのリソースを取得する際は、requestsモジュールを使います。
6. Webアプリの実装：データのキャッシュ	6. Webリソースを取得する際は、同じデータを何回も取得する事を回避するため、@st.cache_dataデコレータによるキャッシュ機能を使いましょう。
7. ローカルでの動作検証	7. Webアプリが完成したら、公開前にローカル端末で動作検証をしましょう。ローカル端末でのWebアプリの実行方法はstreamlit run *****.pyです。「*****」は作業ディレクトリ直下に置かれたルートとなるファイル名です。
8. Webアプリの公開	8. Webアプリは、Githubへと公開してStreamlit Cloudへと登録する事で、一般公開することができます。
9. 非公開情報の管理：sercrets.tomlファイルへの記録	9. APIへのアクセストークンなどの非公開とすべき情報は、sercrets.tomlファイルへと記入して外部からアクセスできないように隔離することで、安全に扱うことができます。Streamlit Cloudからは設定の「Secrets」へと非公開情報を書き込みます。
10. 非公開情報の管理：.gitignoreでの追跡からの除外	10. 非公開情報が書かれているファイルは、誤ってGithubへとpushしてしまわないように、.gitignoreファイルへと書き込み、履歴の追跡から除外します。

[illegible]

データサイエンスの成果物を手軽に公開するために、この科目ではStreamlitを使います。まずはその使用方法を復習しましょう。

資格関連度