

生成AI基礎

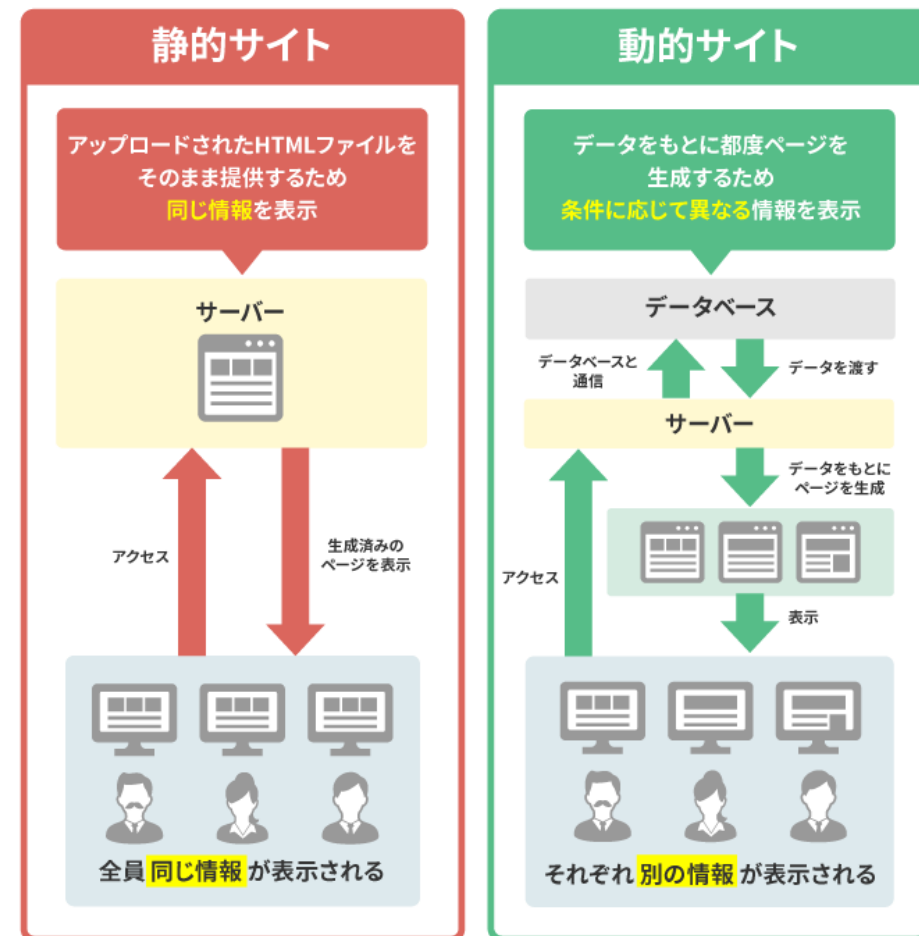
5 アプリケーションサーバーの基本

アプリケーションサーバーの役割

- ブラウザからのリクエストに対して動的にコンテンツを生成し、レスポンスを返す。
- データベース管理システムとの間でデータをやり取りし、ビジネスロジックを処理する。
- セキュリティ（認証・認可）の実装と管理を行う。

静的サイトと動的サイト

- 静的サイトは予め用意されたHTMLをそのまま配信するが、動的サイトはリクエストに応じてHTMLを生成する。
- 静的サイトは高速にアクセス可能だが、動的サイトはより柔軟なユーザー体験を提供する。
- 動的サイトはサーバーサイドでスクリプトが実行されるため、リソースの消費が大きくなることもある。



webアプリケーションフレームワーク

- アプリケーションの開発を効率化するための構造とツールセットを提供する。
- 例としてRuby on Rails, Django, Reactなどがあり、それぞれ特色と利点が異なる。

Streamlitの概要

Streamlitとは何か

- Pythonで書かれたインタラクティブなWebアプリケーションを簡単に作成できるフレームワーク。
- データサイエンスと機械学習プロジェクトに適したツール。

Streamlitのできること

- 単一のスクリプトでインタラクティブなダッシュボードを作成できる。
- リアルタイムデータ更新と視覚化。
- コード変更が即座にWebアプリに反映されるホットリローディングをサポート。

Streamlitの基本的な使い方

Streamlitアプリケーションの作成手順

- Pythonスクリプトにビジュアルコンポーネントのコードを書く。

```
import streamlit as st
st.title("ChatGPTみたいなものを作ってみる")
```

- スクリプトを実行してローカルサーバーを起動。

```
streamlit run main.py --server.port 8501
```

- ブラウザで表示されるアプリケーションを確認。

```
http://{ipアドレス}:8501
```

テキストの表示

- `st.write`は基本的なテキスト表示用の関数。
- `st.title`, `st.header`, `st.subheader`で異なるレベルの見出しを表示。
- マークダウン記法もサポートしており、`st.markdown`で利用可能。

データの表示

- `st.dataframe`でインタラクティブなデータフレームを表示。
- `st.table`で静的なテーブルデータを表示。
- 大規模なデータセットでもスムーズなスクロールとインタラクションが可能。

グラフの表示

- `st.line_chart`で時系列データに適した折れ線グラフを簡単に表示。
- `st.bar_chart`でカテゴリカルデータの棒グラフを表示。
- データを直接引数として渡すだけでグラフが生成される。

地図の表示

- 緯度経度情報を持つデータセットを引数にして地図上にプロット。
- シンプルな使い方でインタラクティブな地図を提供。

ユーザー入力の取得

テキスト入力

- ユーザーからの任意の文字列を入力として受け取る。
- プレースホルダー、デフォルト値、ヘルプテキストを設定可能。
- 入力されたデータを変数に格納してアプリ内で利用。

数値入力

- 整数または浮動小数点数の入力フィールド。
- 最小値、最大値、ステップサイズ（増分）の設定が可能。

選択肢

- `st.selectbox`でドロップダウンリストから選択。
- `st.radio`でラジオボタンを使って選択。
- `st.multiselect`で複数選択可能な選択リストを提供。

ファイルのアップロード

- ユーザーがファイルをアップロードできるウィジェット。
- 画像、CSV、テキストファイルなど、様々なファイル形式をサポート。
- アップロードされたファイルを処理してアプリ内で利用する。

レイアウトとデザイン

サイドバーの追加

- 重要でないコントロールをサイドバーに配置してメインエリアをすっきりさせる。
- ユーザー入力や設定変更をサイドバーで行い、結果はメインエリアに表示。
- スクリプトの初めに`st.sidebar`のウィジェットを配置することで簡単に追加可能。

カラムレイアウト

- ページを複数のカラムに分割して、フレキシブルなレイアウトを作成。
- 各カラム内に異なるコンテンツやウィジェットを配置。
- レスポンシブなデザインをサポートし、異なるデバイスでの表示を最適化。

要素の非表示

- 動的にコンテンツを表示したり非表示にしたりするためのプレースホルダー。
- インタラクティブなアプリケーションでユーザーのアクションに基づいて内容を更新。
- `st.empty()`を使用して、後からコンテンツを追加または削除。

レイアウトとデザイン

チャット要素

- チャット要素（`st.chat_message`と`st.chat_input`）を使用してチャットベースのアプリケーション開発が可能。
- `st.chat_message`: ユーザーまたはアシスタントの役割でメッセージを表示するために使用。
- `st.chat_input`: ユーザーからのテキスト入力を受け付けるために使用。

Streamlit Cloud

Streamlit Cloudを使った共有

- Streamlitアプリをインターネット上で簡単に共有。
- GitHubリポジトリにコードをプッシュするだけでデプロイ可能。
- コラボレーションとフィードバック収集に便利。

まとめ

- StreamlitはPythonで簡単にWebアプリケーションを作成できるフレームワーク。
- データの表示、グラフの作成、ユーザー入力の取得など、様々な機能を提供。
- Streamlit Cloudを使ってアプリを共有・公開することができる。