

# W e b プログラミング I

## 第1回：イントロダクションと環境構築

2025年度 大妻女子大学 社会情報学部

担当：余野

出席認証コード: 0635

## 本日の内容

1. 授業の概要と目標
2. 担当教員の紹介
3. Webアプリケーションとは
4. Streamlitの特徴
5. 開発環境の構築
6. はじめてのStreamlitアプリ

# 授業の概要と目標

## この授業で学ぶこと

- Pythonプログラミングの基礎知識
- **Streamlit**というフレームワークを使ったWebアプリ開発
- データの可視化と対話的なユーザーインターフェース作成

## 到達目標

- Webアプリケーション開発の基礎概念を理解する
- PythonとStreamlitを使って簡単なWebアプリが作成できる
- データ入力フォームやインタラクティブなグラフが作成できる

## 担当教員の紹介

氏名: 余野

専門: ビッグデータデータ分析、経済統計等の指数作成

連絡先:

- メール: [kyoto.yono@otsuma.ac.jp](mailto:kyoto.yono@otsuma.ac.jp)

## 授業の進め方

### 全15回の構成

- 第1-3回: Python基礎
- 第4-7回: Streamlit基本・応用機能
- 第8-10回: データ処理と可視化
- 第11-15回: アプリケーション構築と最終課題

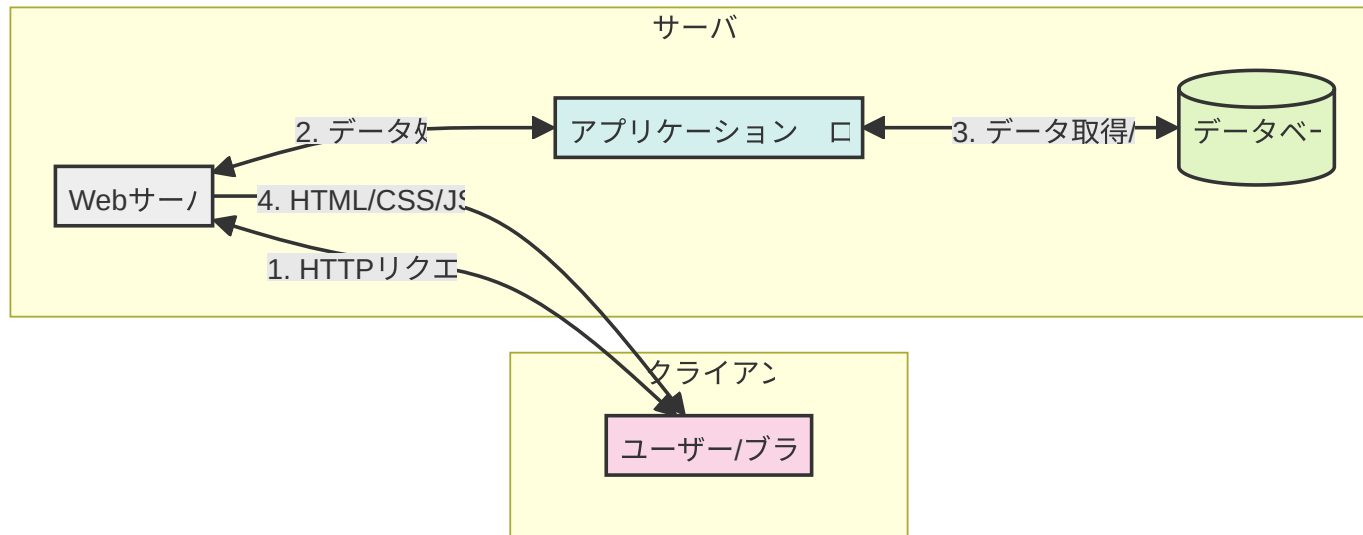
### 評価方法

- 出席: 60%
- 課題: 40%

# Webアプリケーションとは？

## 静的なWebサイトとの違い

- ユーザーからの入力に対して**動的に反応**する
- サーバー側で処理を行い結果を返す



# 一般的なWeb開発の流れ

## 従来のWeb開発に必要な技術

- フロントエンド: HTML, CSS, JavaScript
- バックエンド: Python, Ruby, PHP, Java など
- データベース: SQL, NoSQL
- インフラ: サーバー設定、デプロイ

## 学習の難しさ

- 多くの言語・技術を学ぶ必要がある
- フロントエンドとバックエンドの連携が複雑

## Streamlitとは？

- Pythonだけで**Webアプリを開発**できるフレームワーク
- 2019年に登場した比較的新しい技術
- データ科学者やPython開発者に人気
- コード数行で動くアプリが作れる
- 特にデータ可視化に強み





# Streamlitの特徴

## メリット

- Pythonだけで開発可能（HTML/CSSの知識不要）
- コードの記述量が少ない
- リアルタイムで変更が反映される
- データの可視化が簡単
- 機械学習モデルのデモに最適

## デメリット

- 大規模アプリには向かない
- カスタマイズ性に制限がある

## Streamlitで作れるもの

- データ分析ダッシュボード
- 対話型データ可視化
- 機械学習モデルのデモ
- 簡単なツール・ユーティリティ

より多くの例は、[Streamlit Gallery](#) を参照してください。

# はじめてのStreamlitアプリを実行する

## 1. Streamlitをインストール

```
pip install streamlit
```

## 2. Pythonファイルを作成 ( `app.py` という名前で保存)

```
import streamlit as st

st.title("はじめてのStreamlitアプリ")
st.write("こんにちは！Streamlitの世界へようこそ！")

name = st.text_input("あなたの名前は？")
if name:
    st.write(f"こんにちは、{name}さん！")

st.subheader("Streamlitでできること")
st.write("- データの可視化")
st.write("- インタラクティブなWebアプリ")
```

## 3. アプリを実行

```
streamlit run app.py
```

## Streamlitの基本的な機能

- `st.title()` : タイトル表示
- `st.write()` : テキスト・データの表示
- `st.text_input()` : テキスト入力フィールド
- `st.button()` : ボタン
- `st.selectbox()` : ドロップダウンリスト
- `st.checkbox()` : チェックボックス
- `st.slider()` : スライダー

## Google Colaboratoryの使い方

- Pythonの学習用として活用できます。
- ブラウザ上で**すぐに始められ**、環境構築は不要です。

### 手順

1. 大学のIDとPWでGoogleアカウントでアクセス: <https://colab.research.google.com/>
2. 「ファイル」メニューから「ノートブックを新規作成」を選択。
3. セルにPythonコードを記述し、「Shift + Enter」で実行。

### 例: Hello Worldを表示

```
print("Hello, World!")
```

## GitHubアカウントの作成

今後の演習で作成したコードやアプリを保存・共有するために、GitHubアカウントの作成します。

1. <https://github.com/> にアクセス。
2. 右上の「Sign up」ボタンをクリック。
3. 画面の指示に従って、ユーザー名、メールアドレス、パスワードを設定。
  - ※ ユーザー名は OWU 学籍番号 例) OWU131300000
  - ※ メールアドレスは大学のもの

## GitHub Codespacesの使い方（クラウドIDE）

授業用のテンプレートから、自分専用の開発環境をブラウザ上で起動します。

1. 以下のテンプレートリポジトリURLにアクセス:

`https://github.com/kyouto-yono-ac/web\_programming\_2025`

2. 緑色の **Use this template** ボタンをクリックし、「Open in codespace」を選択。
3. 少し待つと、ブラウザにVS Codeのようなエディタが表示されます。
4. エディタでコードを編集し、ターミナルで実行します。

## 本日のまとめ

- Streamlitはシンプルにアプリを作成できるPythonフレームワーク
- 従来のWeb開発と比べて学習コストが低い
- データ分析・可視化に特化している

## 次回予告

「Python基礎 (1): 変数、データ型、演算」

- Pythonの基本的な文法を学びます



## 参考資料

- Streamlit公式ドキュメント: <https://docs.streamlit.io/>
- Streamlit Gallery (サンプルアプリ集): <https://streamlit.io/gallery>
- Streamlit公式チュートリアル: <https://docs.streamlit.io/library/get-started>