

CSSフレームワークの利用

Webプログラミング

Webプログラミング ～本日の内容～

本日の内容

- FlaskでCSSのフレームワークを利用する方法を学ぶ
- Bootstrapについて学ぶ

進め方

- 資料中に掲載されているサンプルプログラムは全て作成してください
- 作成したプログラムは各自の環境において動作を確認してください

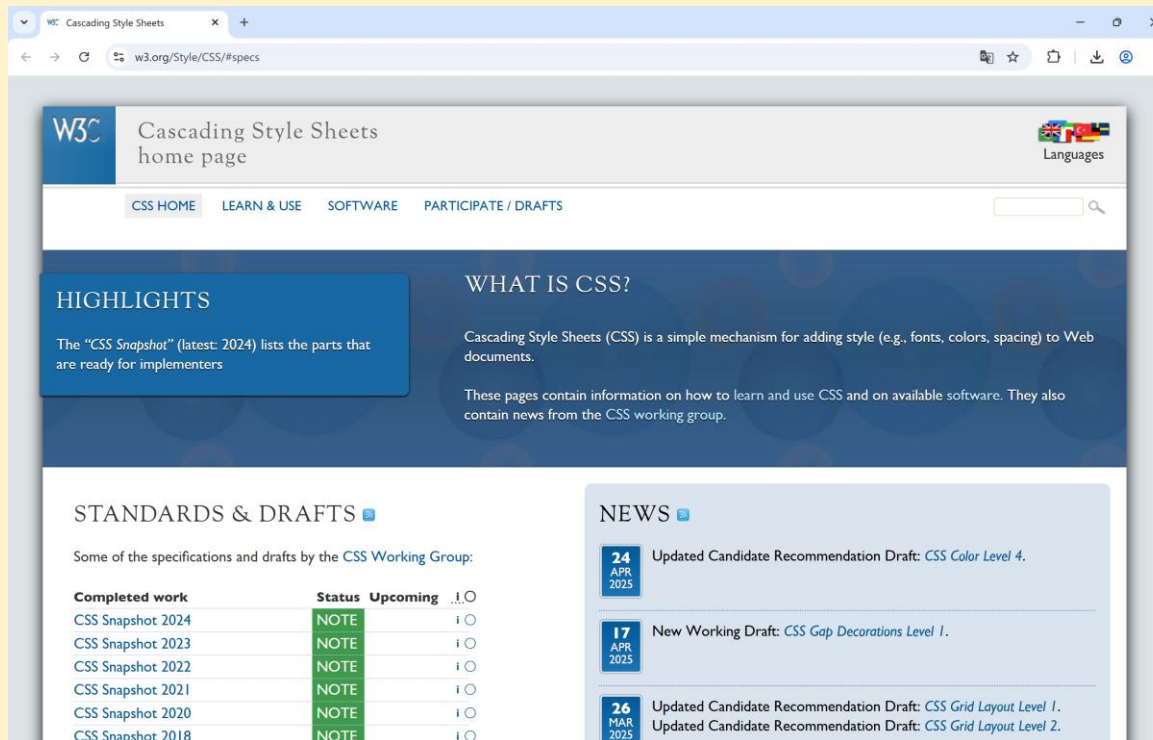
課題

- この資料の最後にある練習問題「CSSフレームワークの利用」を作成し、manabaのレポート機能から提出してください

CSSフレームワーク

CSSフレームワーク ～CSSとは～

- CSS（Cascading Style Sheets）とは，Web画面を構成するHTMLの色やレイアウトといった見た目を装飾するための言語です



W3C, Cascading Style Sheets home page, <https://www.w3.org/Style/CSS/>

CSSフレームワーク

～CSSフレームワークとは～

- CSSフレームワークとは、デザインを統一するためのフレームワークでCSSによる統一デザインを簡単に利用するための仕組みです。ここではBootstrapというフレームワークを利用します



Bootstrap, <https://getbootstrap.jp/>

CSSフレームワーク ～Bootstrapの使い方～

- BootstrapはメインとなるCSSとJSを読み込むことで様々なスタイルを簡単に使うことができるフレームワークです。
- 以下が基本のスターターテンプレートです

テンプレート

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhlLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>
  </body>
</html>
```

Bootstrap, <https://getbootstrap.jp/docs/5.3/getting-started/introduction/>

CSSフレームワーク ～Bootstrapの使い方～

- 日本語に変更, htmlとbodyにクラスを追加, CSS追加

ex_boot.html

```
<!doctype html>
<html lang="ja" class="h-100">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>基本レイアウト用のテンプレート</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-SgOJa3DmI69IUzQ2PVdRZhwQ+dy64/BUtbMJw1MZ8t5HZApCHrRKUc4W0kG879m7" crossorigin="anonymous">
  <link href="../static/boot.css" rel="stylesheet">
</head>
<body class="d-flex flex-column h-100">

<!-- ヘッダー部分 -->

<!-- メインのコンテンツ部分 -->

<!-- フッター部分 -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.min.js" integrity="sha384-k6d4wzSIapyDyv1kpU366/PK5hCdSbCRGRCMv+epIQJWyD1fbcAu9OCUj5zNLiq" crossorigin="anonymous"></script>

</body>
</html>
```

boot.css

```
main > .container {
  padding: 60px 15px 0;
}
```

CSSフレームワーク ～Bootstrapの使い方～

• ヘッダー部分を追加します

ex_boot.html (ヘッダー部分)

```
<!-- ヘッダー部分 -->
```

```
<header>
```

```
<nav class="navbar navbar-expand-md fixed-top bg-primary" data-bs-theme="dark">
```

```
<div class="container-fluid">
```

```
<a class="navbar-brand" href="#">Webプログラミング</a>
```

```
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse" aria-  
controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
```

```
<span class="navbar-toggler-icon"></span>
```

```
</button>
```

```
<div class="collapse navbar-collapse" id="navbarCollapse">
```

```
<ul class="navbar-nav me-auto mb-2 mb-md-0">
```

```
<li class="nav-item"><a class="nav-link" href="#">ページA</a></li>
```

```
<li class="nav-item"><a class="nav-link" href="#">ページB</a></li>
```

```
<li class="nav-item"><a class="nav-link" href="#">ページC</a></li>
```

```
<li class="nav-item"><a class="nav-link" href="#">ページD</a></li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
</nav>
```

```
</header>
```


CSSフレームワーク ～Bootstrapの使い方～

- メインのコンテンツ部分とフッター部分を追加します
- メイン部分には見出しと文章を追加
- フッターは下部に位置固定

ex_boot.html (メインのコンテンツ部分)

```
<main class="flex-shrink-0">
  <div class="container">
    <h1 class="mt-5 display-4">Bootstrapを用いた基本レイアウト</h1>
    <p class="lead">Bootstrapを用いてレイアウトします。 上部にヘッダーと各ページへのナビゲーションを作成して、下部にはフッターを配置します。 </p>
  </div>
</main>
```

ex_boot.html (フッター部分)

```
<!-- ヘッダー部分 -->

<footer class="footer mt-auto py-3 bg-primary" data-bs-theme="dark">
  <div class="container text-end">
    <span class="text-light">Webプログラミング（火曜日5時間目7201教室） </span>
  </div>
</footer>
```

CSSフレームワーク ～Bootstrapの使い方～

- ex_boot.htmlをtemplateフォルダに配置して，実行用のex_boot.pyを作成します

ex_boot.py

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("ex_boot.html")

if __name__ == "__main__":
    app.run()
```



HTMLのテンプレート化

CSSフレームワーク ～Bootstrapの使い方～

- headタグ内のテンプレート化として以下の箇所を変更
 - ◆ headタグ内の全体を変更可能にheadのブロックを作る
 - ◆ タイトルにデフォルトを設定して中身をtitleブロックとして変更可能にする
 - ◆ CSS用にstyleブロックを作り、デフォルトではurl_for関数でboot.cssを読み込む構成とする

ex_boot_base.html (headタグ内)

```
<head>
  {% block head %}
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>{% block title %}Webプログラミング{% endblock %}</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-SgOJa3DmI69IUzQ2PVdRZhwQ+dy64/BUtbMJw1MZ8t5HZApCHrRKUc4W0kG879m7" crossorigin="anonymous">
    {% block style %}<link href="{{ url_for('static', filename='boot.css') }}" rel="stylesheet">{% endblock %}
  {% endblock %}
</head>
```

CSSフレームワーク ～Bootstrapの使い方～

- headタグ内の全体を変更可能にheadのブロックを作る
- メニュー用を変更可能にするためにheadermenuブロックを作成

ex_boot_base.html (headerタグ内)

```
{% block header %}  
<header>  
  <nav class="navbar navbar-expand-md fixed-top bg-primary" data-bs-theme="dark">  
    <div class="container-fluid">  
      <a class="navbar-brand" href="#">Webプログラミング</a>  
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse" aria-  
controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">  
        <span class="navbar-toggler-icon"></span>  
      </button>  
      <div class="collapse navbar-collapse" id="navbarCollapse">  
        <ul class="navbar-nav me-auto mb-2 mb-md-0">  
          {% block headermenu %}  
            <li class="nav-item"><a class="nav-link" href="#">ページA</a></li>  
            <li class="nav-item"><a class="nav-link" href="#">ページB</a></li>  
            <li class="nav-item"><a class="nav-link" href="#">ページC</a></li>  
            <li class="nav-item"><a class="nav-link" href="#">ページD</a></li>  
          {% endblock %}  
        </ul>  
      </div>  
    </div>  
  </nav>  
</header>  
{% endblock %}
```

CSSフレームワーク ～Bootstrapの使い方～

- メインコンテンツの追加領域としてcontentブロックを作成し、フッターとスクリプト用のブロックをそれぞれ作成

ex_boot_base.html（残りの箇所）

```
{% block content %}{% endblock %}
```

```
{% block footer %}
```

```
<footer class="footer mt-auto py-3 bg-primary" data-bs-theme="dark">
```

```
<div class="container text-end">
```

```
<span class="text-light">Webプログラミング（火曜日5時間目7201教室）</span>
```

```
</div>
```

```
</footer>
```

```
{% endblock %}
```

```
{% block script %}
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.min.js" integrity="sha384-k6d4wzSIapyDyv1kpU366/PK5hCdSbCRGRCMv+epIQJWyd1fbcAu9OCUj5zNLiq" crossorigin="anonymous"></script>
```

```
{% endblock %}
```

CSSフレームワーク ～Bootstrapの使い方～

- ex_boot_base.htmlを継承して, ex_boot2.htmlを作成
- メインのコンテンツ部分をここに移す

ex_boot2.html

```
{% extends "ex_boot_base.html" %}
{% block title %}ページのタイトルを入れます{% endblock %}
{% block content %}
<main class="flex-shrink-0">
  <div class="container">
    <h1 class="mt-5 display-4">Bootstrapの基本レイアウト</h1>
    <p class="lead">Bootstrapを用いてレイアウトします。 上部にヘッダーと各ページへのナビゲーションを作成して, 下部にはフッターを配置します。 </p>
  </div>
</main>
{% endblock %}
```

CSSフレームワーク ～Bootstrapの使い方～

- ex_boot_base.htmlとex_boot2.htmlをtemplateフォルダに配置して、実行用のex_boot2.pyを作成します

ex_boot2.py

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("ex_boot2.html")

if __name__ == "__main__":
    app.run()
```



練習問題

CSSフレームワーク

～練習：CSSフレームワークの利用～

演習（次回までの課題）

ex_boot_base.htmlを利用し，テンプレートを継承することでBootstrapの部分のリンクメニューを共通化してみましよう

- HTMLは「ex_boot_base.html」を継承した「ex_boot3.html」を作成
- URLは「/」と「/web」, 「/python」, 「/boot」として，ヘッダーの中でリンクで繋ぐ
- 「ex_boot3.py」でリンクを制御

