

# Setup

## Git

1. Stworzenie nowego repozytorium (*np. Aplikacje Webowe*)
2. Sklonowanie nowego repozytorium

```
git clone https://github.com/[Username]/[repository].git
```

1. Stworzenie folderu dla laboratorium (*np. Lab\_01*)
2. Skopiowanie tego notebooka do stworzonego folderu (*będzie nam to potrzebne w kolejnych krokach*)
3. Dodanie zmian do repozytorium

```
git add .  
git commit -m "Initial setup"  
git push
```

## Visual Studio Code

**Można używać dowolnego edytora! Wtedy tą sekcję można opuścić.**

1. Pobieramy program z linku: <https://code.visualstudio.com/download>
2. Otwieramy nasze repozytorium przy użyciu VS Code
3. Pogranie pluginu [Live Server](#)
4. Pobranie pluginu [Prettier - Code formatter](#)
5. Włączenie formatowania przy zapisie
  - command + shift + p
  - open settings
  - wyszukać i włączyć - "Format On Save"
  - default formatter - "Prettier"

## Commitujemy zmiany

```
git add .  
git commit -m "Add templates"  
git push
```

## Lab 1 - Ćwiczenia [5.0 pkt]

- Na całe ćwiczenie należy stworzyć jeden osobny folder.
- Plik **index.html** zawiera nagłówek (np. `<h1>`) z tekstem "Wykonane ćwiczenia" oraz listę numerowaną składającą się z linków do wszystkich podstron, na których będą

zrealizowane ćwiczenia. Tekst każdego z linków to ćwiczenie z jego numerem np.  
**Ćwiczenie 1.** Każdy z linków powinien być innego koloru.

- **[1.0 pkt]** Ćwiczenie 1: Artykuł
  - Na górze artykułu ma znajdować się nagłówek, a pod nagłówkiem autor artykułu (imię i nazwisko autora ma być w wyróżniającym się kolorze)
  - Artykuł ma składać się z co najmniej 3 paragrafów. Treść można wygenerować z wykorzystaniem tzw. [Lorem Ipsum](#)
  - Pod artykułem należy umieścić obrazek
- **[2.0 pkt]** Ćwiczenie 2: Lista produktów w formie Tabeli (`<table>`)
  - Tabela ma zawierać następujące kolumny: **Nr.**, **Nazwa produktu**, **Cena produktu**, **Ilość sztuk**
  - Należy wymyślić 5 dowolnych produktów i umieścić je w tabeli.
  - Tabela powinna zawierać obramowania. Proszę zapoznać się z działaniem i wykorzystać atrybut CSS `border-collapse: collapse`
- **[2.0 pkt]** Ćwiczenie 3: Kolorowe kwadraty
  - Należy utworzyć 5 kwadratów (Element `<div>`) o wymiarach 100px szerokości i 100px wysokości. Każdy kwadrat ma mieć inny kolor tła.
  - Kwadraty należy ułożyć w dwóch wierszach - 2 w pierwszym wierszu i 3 w drugim wierszu.
  - Między kwadratami powinny być odstępy.
  - Jeden z kwadratów należy podmienić na dowolny obrazek - także o wymiarach 100x100 - ma on zachowywać się tak samo (być w tym samym miejscu co wcześniej div)

## Projekt 1 – Statyczna strona WWW [20.0 pkt]

W ramach projektu należy stworzyć prostą statyczną stronę WWW.

Tematyka projektu jest dowolna. Przykładowe tematy:

- Osobiste portfolio
- Strona informacyjna o hobby
- Strona promująca fikcyjną firmę lub produkt
- Strona wydarzenia
- Strona o filmie / grze / książce

### Wymagania funkcjonalne

#### Wymagania podstawowe [8.0 pkt]

- Istnieją przynajmniej 3 podstrony (Home, Kontakt, Galeria)
  - Podpowiedź: Każda podstrona może osobnym plikiem HTML
- Strona posiada layout [grid](#) / [flexbox](#)
  - U góry/z boku strony znajduje się **navbar** (`<nav>`) z linkami do wszystkich podstron.
  - Na dole strony znajduje się **footer** (`<footer>`) z Twoim imieniem i nazwiskiem.

- Należy wykorzystać [znaczniki semantyczne](#) HTML.
- Między podstronami różnic powinna się tylko zawartość główna. Nagłówek oraz stopka mają być jednolite na każdej z nich.
- Strona główna - **Home** może zostać zaprojektowana w dowolny sposób, zgodny z tematyką projektu. Należy pamiętać, że jest to najważniejszy element projektu i pierwsza rzecz, którą widzi użytkownik wchodzący na stronę - więc powinna być reprezentacyjna i robić dobre pierwsze wrażenie.
- Na podstronie **Kontakt** znajduje się formularz kontaktowy:
  - Wymagane pola w formularzu:
    - E-Mail (Pole tekstowe - input type **text**)
    - Temat wiadomości (Select z przynajmniej 3 dowolnymi opcjami)
    - Wiadomość (Textarea)
    - "Wyślij" (Przycisk, input type **submit**)
  - Pozostałe wymagania dotyczące formularza:
    - Pola **E-mail** oraz **Wiadomość** zawierać odpowiedni atrybut "placeholder"
- Na podstronie **Galeria** należy umieścić:
  - Kilka zdjęć
    - Z wykorzystaniem **grid** lub **flexbox** zdjęcia powinny układać się po kilka w jednym wierszu i zawijać, jeżeli jest ich dużo (wrap).
  - Film (<video>)

### Wygląd strony [6.0 pkt]

- Wykorzystać CSS do ostylowania strony (przykłady poniżej)
  - Dowolna czcionka
    - Mile widziane będzie użycie czcionki z [google fonts](#) - po wybraniu czcionki należy kliknąć "Get embed code" aby wyświetlić instrukcję dodania jej do strony.
  - wybór odpowiedniej [palety kolorów](#)
  - kolory tła i tekstu
- Poprawne ustawienie elementów na stronie i odpowiednie odstępy
- Konsystentny design, spójność graficzna
- Dobry/atrakcyjny wygląd strony

### Wymagania dodatkowe, na wyższą ocenę [6.0 pkt]

- **[1.0 pkt]** Walidacja danych w dowolnym formularzu znajdującym się na stronie - [Krótki poradnik](#)
- **[1.0 pkt]** Udostępnienie strony publicznie - np. [Github Pages](#)
- **[2.0 pkt]** [Responsywność](#) strony - poprawny wygląd bez względu na szerokość ekranu (na telefonie, na tablecie, na komputerze)
- **[2.0 pkt]** Wykorzystać JS (przykłady poniżej)
  - zapisywanie preferencji użytkownika (cookies / localStorage)
  - obsługą formularza
  - stworzenie dynamicznych elementów (np. show / hide)
  - zegar
  - przechodzenie między podstronami

## Przydatne linki

- <https://www.w3schools.com/>
- <https://css-tricks.com/>
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://developer.mozilla.org/en-US/>
- <https://fonts.google.com/>