Warsztaty front-endowe
Zacznij w IT

HTML, CSS i Git
Część 1

Coders Lab
SZKOŁA IT

HTML & CSS
Szybka powtórka

HTML

Na sam początek przypomnijmy sobie, czym jest HTML oraz jak wygląda struktura kodu. Przygotuj sobie edytor, bo zaraz przejdziemy do małego ćwiczenia :)

- HTML to hipertekstowy język znaczników
- HTML opisuje strukturę strony
- Elementami składowymi HTML są znaczniki, za pomocą których opisuje się strukturę
- HTML nie jest językiem programowania, bo nie można w nim wykonywać obliczeń, warunków ani iteracji
- Dokument HTML musi mieć odpowiednią strukturę, która zawiera m.in. takie znaczniki jak <html>, <head>, <body>

Przykładowa struktura dokumentu HTML

```
<!DOCTYPE html>
 <html>
  <head>
   <meta charset="UTF-8">
   <title>Tytuł strony</title>
  </head>
  <body>
   <div>
    <p>Przykładowy paragraf</p>
   </div>
  </body>
 </html>
```

Sprawdź się!

Stwórz nowy dokument HTML w swoim edytorze, a następnie wykonaj poniższe zadania:

1. Zbuduj odpowiednią strukturę dokumentu, czyli umieść znaczniki <html>, <head>, <body>.
2. Do <body> dodaj dwa przykładowe paragrafy. W każdym z nich umieść jakiś tekst.
3. Pod paragrafami dodaj listę.
4. Pod listą wstaw obrazek.

PODPOWIEDŹ

Jeśli nie pamiętasz, jak powinna wyglądać odpowiednia struktura, zajrzyj do kursu, który robiłaś/łeś albo poszukaj przykładów w internecie.

Gdy nie masz pomysłu na treść paragrafów, użyj generatora tekstu Lorem Ipsum, na przykład tego > http://pl.lipsum.com/

Dużo darmowych zdjęć do użycia na stronie znajdziesz na przykład w tym banku zdjęć > http://www.jestrudo.pl/darmowe-zdjecia/

CSS

Teraz czas na szybką powtórkę z zakresu CSS.

- CSS to kaskadowe arkusze stylów
- CSS służy od opisania wyglądu strony
- CSS składa się z reguł, które mają własności oraz ich wartości
- Selektory to zapisy, które przypisują reguły CSS do odpowiednich elementów dokumentu HTML
- Plik CSS dołączamy do pliku HTML za pomocą znacznika <link>, w którym podajemy ścieżkę do pliku. **Ścieżka zależy od tego, gdzie znajduje się plik HTML. W poniższym przykładzie plik *style.css* znajduje się w katalogu *css*, który jest na tym samym poziomie co *index.html*.**

```
<link rel="stylesheet" href="css/style.css">
```

Przykładowy kod CSS

```
p {
        color: red;
}

.examplary-class-name {
        font-weight: bold;
        border: 1px solid green;
}

#examplary_id {
        background-color: blue;
}
```

Sprawdź się!

W tym samym katalogu, w którym stworzyłaś/łeś plik HTML, załóż nowy folder o nazwie *Style*. W tym katalogu utwórz nowy plik CSS o nazwie *style.css*, a następnie wykonaj poniższe zadania.

1. Dołącz plik CSS do swojego pliku HTML za pomocą znacznika <link>.
2. Nadaj różne klasy dwóm paragrafom, a następnie w pliku CSS przypisz im różne kolory czcionek.
3. Zmień style dla swojej listy – spraw, by każdy z jej elementów miał inny kolor.
4. Dodaj do obrazka ramkę w wybranym przez siebie kolorze oraz zmień mu marginesy.

PODPOWIEDŹ

Nie musisz znać na pamięć wszystkich własności CSS. Możesz łatwo je sprawdzić na przykład na tej stronie > https://developer.mozilla.org/pl/docs/Web/CSS/CSS_Reference

Plan działania

Plan działania

Po tej krótkiej powtórce zabieramy się za konkrety! Przyszedł czas, żeby zająć się naszą warsztatową stroną. Poniżej znajdziesz plan działania na dziś. Zobacz, czym zajmiemy się po kolei :)

1. Git – co to takiego? Wprowadzenie oraz przygotowanie środowiska do pracy (to tutaj w końcu zobaczysz warsztatową stronę!)
2. Struktura naszej strony + omówienie Boostrapa
3. Czas dla Ciebie – zmodyfikuj kod HTML i CSS i zobacz, jak zmienia się strona!
4. JavaScript – omówienie podstaw i kilka prostych ćwiczeń na rozgrzewkę
5. Warsztat, czyli wprowadzamy funkcjonalności JS do strony warsztatowej na podstawie instrukcji na GitHubie

Gotowi do działania? Ruszamy!

P.S. Nie martw się, w międzyczasie zrobimy przerwę na lunch ;)

Git – co to takiego?

Git – co to takiego?

. Git to system kontroli wersji
. Śledzi wszystkie zmiany dokonywanie w plikach
. Pozwala zobaczyć, jakich zmian dokonywały inne osoby pracujące nad projektem
. Umożliwia przywrócenie dowolnej, poprzedniej wersji kodu, jeśli zajdzie taka potrzeba
. Jest podstawą, gdy nad kodem pracuje dużo osób

PRZYDATNE POJĘCIA

Repozytorium – miejsce przechowywania Twojego kodu (zazwyczaj oznacza jeden projekt)
Commit – zatwierdzenie zmian, które wprowadzasz na plikach
Push – wysłanie paczki zmian (commita) stworzonych lokalnie do zdalnego repozytorium
Pull – pobranie zmian ze zdalnego repozytorium do repozytorium lokalnego
Branch – gałąź w repozytorium; polega na zrobieniu kopii naszego repozytorium na którym możemy pracować; pomiędzy takimi kopiami (gałęziami) możemy w łatwy sposób się przełączać; służy to do rozwijania jednego projektu przez wielu developerów w taki sposób, w którym każdy z nich pracuje na swojej gałęzi; zazwyczaj produkcyjna wersja programu znajduje się na gałęzi *master*, a wersje deweloperskie na tzw. *feature branch* (nazwy *feature branchy* są ściśle związane z tym co na nich jest tworzone)

Pobieramy projekt

Pobieramy projekt

Projekt oraz wszystkie warsztatowe zadania umieszczone są w serwisie github.com. Aby móc pobrać projekt, musisz mieć dostęp do repozytorium *Programuj, dziewczyno!* Aby zacząć pracę nad stroną, wykonaj poniższe polecenia. W razie problemów – pytaj!

1. Wejdź na stronę https://github.com/ i zaloguj się do swojego konta.
2. Wejdź do repozytorium *Warsztaty_JS* *https://github.com/CodersLab-core/Warsztaty_JS*
3. W prawym, górnym rogu kliknij w przycisk **Fork**
4. Wybierz swój profil. Teraz masz już repozytorium podpięte do swojego konta i możesz zacząć na nim pracować.

Pobieramy projekt cd.

5. Czas na włącznie konsoli. Jeśli korzystasz z Linuxa albo OS X, włącz terminal. Jeśli korzystasz z Windowsa, wyszukaj Git Bash.

6. Przez terminal wejdziemy teraz do miejsca, do którego chcesz sklonować repozytorium, żeby zacząć na nim pracę. Może to być na przykład Pulpit. Polecenie **cd** pozwoli Ci przejść do konkretnej ścieżki. Z kolei **pwd** pokaże Ci, w jakim miejscu jesteś teraz. Załóżmy, że chcesz skopiować repozytorium na pulpit. Wtedy powinnaś wpisać następującą komendę:

cd Desktop

W zależności od ustawień języka Twojego systemu, może to być też:

cd Pulpit

7. Teraz znajdujesz się na pulpicie. Czas na sklonowanie repozytorium. Skopiuj z przeglądarki adres repozytorium podpiętego do Twojego konta. Następnie wywołaj komendę **git clone** razem ze skopiowanym adresem:

git clone https://github.com/CodersLab-core/Warsztaty_JS

Pobieramy projekt cd.

8. W swoim katalogu powinieneś widzieć teraz nowy katalog razem z zawartością repozytorium. Wejdź do niego i dwukrotnie kliknij w plik index.html. W przeglądarce powinna otworzyć Ci się strona warsztatowa.

9. Otwórz katalog z plikami z repozytorium w swoim edytorze. Możesz teraz zacząć pracę nad plikami :) Wszystkie zmiany będą zapisywane lokalnie.

Coders Lab
SZKOŁA IT

Struktura strony & Bootstrap

Struktura strony

Prześledź kod HTML oraz CSS warsztatowej strony. Postaraj się zidentyfikować konkretne sekcje strony, poszukaj elementów odpowiedzialnych za wyświetlanie jej poszczególnych części. Zobacz, jakie klasy mają dane elementy oraz jakie style są do nich przypisane.

Widzisz coś nowego?
Czegoś nie rozumiesz?
Pytaj!

Bootstrap

W kodzie HTML Twoją uwagę powinny zwrócić takie nazwy klas jak np. c*ol-md-12, col-sm-2* albo *jumbotron*, których nie widzisz w pliku CSS. Skąd więc wzięły się one w kodzie HTML i jakie style są do nich przypisane?

Bootstrap to framework CSS, który dzięki wbudowanym klasom pozwala budować responsywne strony. Ma mnóstwo gotowych stylów, których można używać na stronach. Posiada takie gotowe elementy jak m.in. menu czy nawigację. Najczęściej używany jest do grida, czyli tzw. siatki na stronie, dzięki czemu łatwo określić szerokość i położenie elementów na stronie (za to właśnie odpowiedzialne są klasy zaczynające się od *col* oraz *row*).

Boostrap pomaga budować stronę z predefiniowanymi stylami, dzięki czemu zaoszczędzamy dużo czasu, bo nie musimy wszystkiego stylowac od zera.

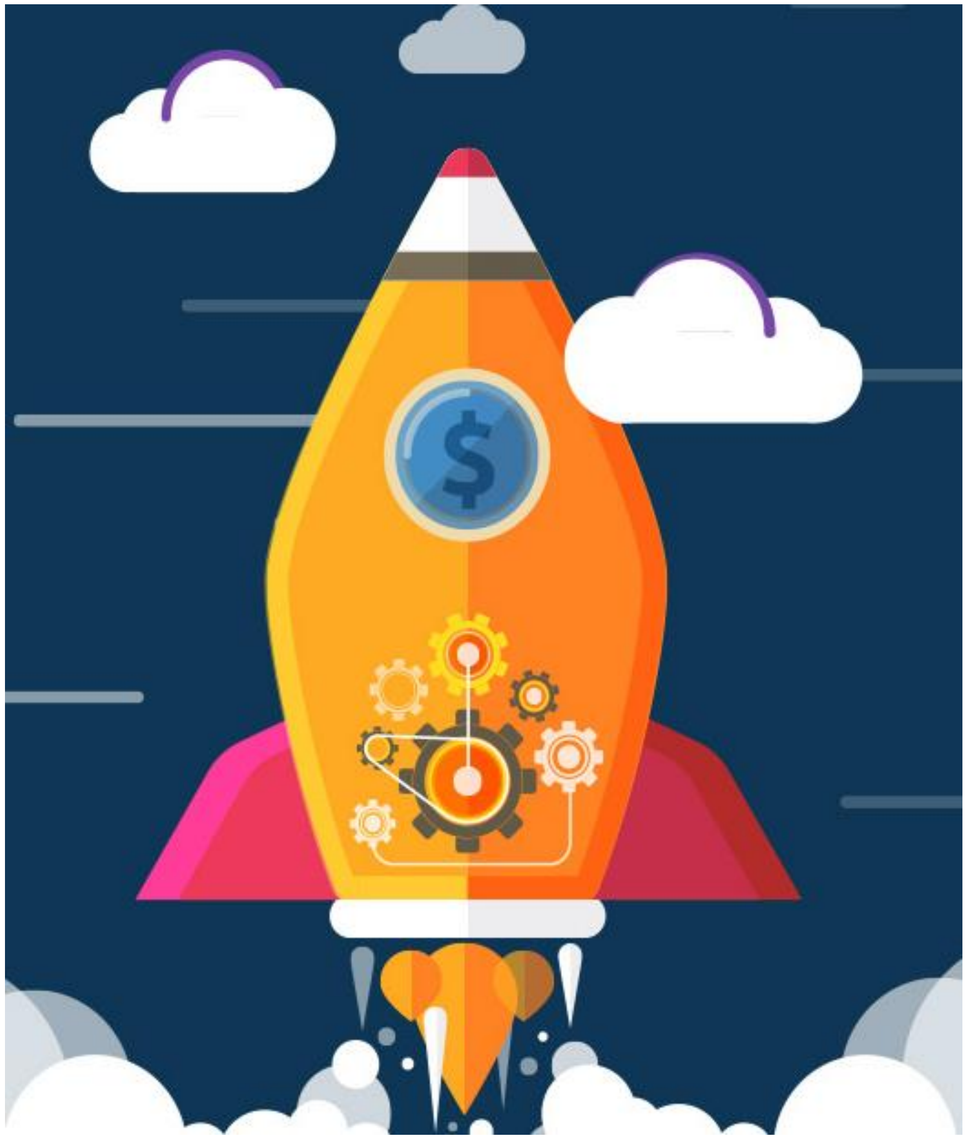
Chcesz wiedzieć więcej?
Poczytaj oficjalną dokumentację Boostrapa > http://getbootstrap.com/
Zrób kurs na Codecademy > https://www.codecademy.com/courses/web-beginner-en-yjvdd/0/1

Twoje zmiany & pierwszy commit

Czas na zmiany!

Otwórz w edytorze katalog z projektem i wprowadź następujące zmiany:

1. Zmień tytuł strony w <head> na “Warsztaty”. Taki napis powinien pojawić się na zakładce podglądu pliku index.html.
2. W liście znajdującej się w divie o klasie “dropdown” dodaj jeden dodatkowy element z kolejnym linkiem do materiałów do nauki. Jeśli nie masz pomysłu, wklej ten link: https://www.codewars.com/
3. Zwiększ odrobinę odstęp między elementem <h2> o klasie “main-slogan” i przyciskiem o klasie “tasks-btn”. Użyj do tego własności *margin* dla odpowiedniego elementu w pliku CSS.
4. Spraw, aby tekst paragrafu znajdującego sie w divie o klasie “plan-description” był pogrubiony. Użyj do tego właśności *font-weight* w pliku CSS.



Pierwszy commit

Czas zapisać wprowadzone przez Ciebie zmiany! :)

1. Otwórz terminal i wejdź do katalogu z projektem używając komendy **cd** i nazwy katalogu z repozytorium. Jeśli swój projekt sklonowałaś/łeś na pulpit, powinieno się móc wejść do niego wpisując:

cd Desktop/Warsztaty_JS

2. Wpisz w terminalu komendę:

git status

Pozwoli Ci ona zobaczyć, w jakich plikach zostały dokonane zmiany.

3. Jeśli chcesz te zmiany zachować wpisz komendę:

git add .

Ta kropka na końcu nie jest przypadkowa! Koniecznie o niej pamiętaj. Dzięki niej wszystkie pliki ze zmianami zostaną dodane do commita.

Pierwszy commit cd.

4. Możesz też dodawać poszczególne pliki wpisując odpowiednią nazwę zamiast kropki w poleceniu **git add**. Na przykład, jeśli chcąc zachować tylko zmiany wprowadzone w pliku index.html, wpiszemy:

git add index.html

Jednak na potrzeby warsztatów, cały czas używaj **git add .** (z kropką na końcu), aby za każdym razem miec pewność, że wszystkie pliki zostały dodane.

5. Czas na commit – wpisz w konsoli:

git commit -m "tutaj podaj treść commita"

Treść commita powinna opisywać wprowadzone zmiany w taki sposób, by łatwo było Ci zidentyfikować, co zrobiłaś/łeś w tej części kodu.

6. Aby wysłać commita do repozytorium wpisz w konsoli **git push**

7. Teraz powinieneś móc zobaczyć aktualną wersję kodu w swoim repo na Githubie.

Ciąg dalszy w drugiej części prezentacji :)