

FRONT-END PREWORK



Spis treści

1. [Powitanie i podstawowe informacje](#)
2. [Sposób “oddania” ukończonego preworka](#)
3. [Przygotowanie środowiska](#)
4. [Instalacja oprogramowania](#)
5. [Przygotowanie merytoryczne](#)
 - [System kontroli wersji - GIT](#)
 - [HTML/CSS](#)
 - [JavaScript](#)
6. [Dokumentacja](#)
7. [Kontakt](#)

Cześć!

Miło nam Cię widzieć na naszym kursie! W infoShare Academy dbamy o to, aby kursy były na najwyższym poziomie i traktowały wyłącznie o najnowszych technologiach, na które jest aktualnie zapotrzebowanie na rynku.

Dlatego chcemy Cię przygotować do kursu jeszcze zanim się na nim pojawisz!

Większość z materiałów to zagadnienia, z których będą zajęcia podczas kursu, ale zapoznanie się z nimi już teraz jest konieczne do sprawnego rozpoczęcia nauki i opanowania dzięki temu całej niezbędnej wiedzy.

Cześć!

Dlatego niezwykle ważne jest, aby przerobić wszystkie zawarte tu materiały. Warto też skonfigurować środowisko pracy i się z nim „oswoić”. Dzięki temu podczas kursu gładko wejdiesz w programowanie i nie będziesz ryzykować zaległościami z podstaw, gdy zaczniemy bardziej zaawansowane tematy w drugiej części kursu.

Większość zagadnień jest obowiązkowa i nie wyobrażamy sobie, żeby ktoś rozpoczął kurs bez tej wiedzy.

Zagadnienia, które są fakultatywne, są wyraźnie oznaczone.

Nie traktuj preworku jak pracy domowej, którą trzeba zaliczyć, tylko jak **absolutny MUST HAVE**. Bez tego udział w kursie może nie mieć sensu.

UWAGA!

Konfiguracja środowiska nie jest tak ważna, jak przygotowanie merytoryczne!

Jeśli masz niewiele czasu, rozpocznij od rozdziału „**Przygotowanie merytoryczne**”, pracując na dowolnym systemie operacyjnym i edytorze kodu (IDE).

Dopiero w kolejnych krokach skonfiguruj środowisko.

Chcemy być pewni, że przygotowanie do kursu idzie jak najlepiej!

Z tego powodu, jako potwierdzenie przerobienia wszystkich materiałów użyj formularza dostępnego pod linkiem front.prewrite.is-academy.pl, by przesłać nam:

1. Screenshota ekranu z zainstalowanym Visual Studio Code
2. Screenshotów z ukończonych (ostatnich poziomów) z sekcji HTML/CSS:
 - a. CSS Diner
 - b. Flexbox Foggy
 - c. Flexbox Defense
 - d. Grid Garden
3. Screenshota „Completed” z portalu Edabit (po realizacji zadań kliknij „Your Profile” i zrób screen zakładki „Completed”)
4. Login z GitHuba

Jeśli napotkasz jakieś trudności podczas wykonania materiałów, napisz e-mail na adres prework@infohareacademy.com.

UWAGA! Pamiętaj, żeby w mailu podać nazwę kursu, w którym bierzesz udział!

Najlepiej zatytułuj maila w formacie: **„PYTANIE — Twoje imię i nazwisko | nazwa kursu | data startu”**

INSTALACJA OPROGRAMOWANIA

Instalacja oprogramowania

W pierwszej kolejności zainstaluj Google Chrome - przeglądarkę, z której będziemy korzystać podczas kursu.

Wystarczy, że ściągniesz ją z tego [LINKU](#) i otworzysz - instalator zajmie się resztą, a nowy program pojawi się w aplikacjach (dostępnych pod pierwszą ikoną z paska bocznego lub z przycisku Windows na klawiaturze). Przeciągnij ją na pasek boczny - dla szybkiego dostępu.

W identyczny sposób zainstaluj Slacka - komunikator dla zespołów, który będziemy wykorzystywać podczas pracy. [POBIERZ SLACKA](#).

Instalacja oprogramowania

Pobierz edytory grafiki [GIMP](#) (grafika rastrowa) i [Inkscape](#) (grafika wektorowa), które mogą Ci się przydać podczas tworzenia interfejsów i obrabiania zdjęć.

Polecamy także korzystanie z rozwiązań w chmurze, takich jak Google Drive.

Instalacja oprogramowania - VS Code

Kolejnym krokiem jest instalacja Visual Studio Code czyli programu, w którym będziesz pisać swój kod. VS Code jest programem łączącym cechy IDE i edytora kodu.

IDE to skrót od „zintegrowane środowisko programistyczne” (ang. Integrated Development Environment) i jest to bardzo rozbudowana aplikacja stworzona specjalnie do pisania kodu w danym języku (czasem w kilku językach, podobnego zastosowania).

Edytory tekstu, jak sama nazwa wskazuje, służą do edycji tekstu, nie posiadają żadnych programistycznych funkcji, jak choćby kolorowanie składni kodu (czyli zaznaczanie i rozróżnianie jego fragmentów kolorem). Edytory kodu pozwalają na edycję kodu w wygodny sposób i posiadają wspomniane już kolorowanie, a także narzędzia usprawniające tworzenie kodu, takie jak np. obsługa wielu kursorów (możliwość edycji tekstu w wielu miejscach naraz).

Instalacja oprogramowania - VS Code

Edytory kodu, jak i edytory tekstu skupiają się raczej na pojedynczych plikach, które edytujesz, niż na całej aplikacji, którą tworzysz. IDE skupione są na projektach, a także oferują wiele „dużych” integracji.

Visual Studio Code jest programem, który łączy zalety edytora i IDE. Jest lekki i skupiony bardziej na plikach/folderach, ale oferuje bardzo dużo zaawansowanych narzędzi ułatwiających pisanie kodu jak i mnóstwo dodatków rozszerzających jego możliwości.

By zainstalować Visual Studio Code należy pobrać go ze [strony](#).

Edytor tekstu (Windows)

Przyda nam się podstawowy edytor tekstu - Notepad++ będziemy go również używać przy pracy z git, którego opis znajdziesz w kolejnym punkcie.

Pobierz i zainstaluj Notepad++ z [oficjalnej strony projektu](#).

GIT - system kontroli wersji (Windows)

Ostatnim krokiem konfiguracji jest instalacja systemu kontroli wersji git. Jest to narzędzie do wersjonowania oraz wspólnej pracy nad kodem.

Aby zainstalować gita na naszym systemie:

1. Pobierz oraz uruchom instalator z strony git-scm.com
2. Zainstaluj z domyślnymi opcjami oprócz jednej - jako domyślny edytor wiadomości wybierz Notepad++
3. Uruchom konsolę systemową - wpisz w *menu start* `cmd` aby ją uruchomić
4. Dla potwierdzenia wpisz `git --version`, tym poleceniem możesz sprawdzić wersję gita

Jeśli masz problemy z instalacją gita, więcej informacji znajdziesz [tutaj](#).

GIT - system kontroli wersji (Ubuntu)

Ostatnim krokiem konfiguracji jest instalacja systemu kontroli wersji git, czyli jak już wspominaliśmy wyżej, narzędzia do wersjonowania i współpracy nad kodem. Instalacja na Ubuntu sprowadza się do jednego polecenia w Terminalu.

1. **Wpisz polecenie** `sudo apt-get install git` i poczekaj do końca instalacji.
2. **Dla potwierdzenia** wpisz `git --version`, tym poleceniem możesz sprawdzić wersję gita.

Jeśli instalacja powiodła się, w terminalu pojawi się numer wersji.
Jeśli masz problemy z instalacją gita, więcej informacji znajdziesz [TUTAJ](#).

PRZYGOTOWANIE MERYTORYCZNE

UWAGA!

Przerabiając materiały zalecamy korzystanie z IDE/edytora instalowanego w poprzednich punktach, niezależnie od edytorów wykorzystywanych w proponowanych przez nas kursach!

WSKAZÓWKA!

Pamiętaj, że oglądając materiały online zazwyczaj możesz zmieniać ich prędkość odtwarzania w prawym dolnym rogu!

System kontroli wersji - GIT

Jeśli pracowałeś kiedyś z gitem, potrafisz założyć repozytorium na GitHubie, odpowiednio je opisać oraz utworzyć i zmerge'ować pull requesta, możesz pominąć ten punkt.

Zacznij od obejrzenia przygotowanego przez nas obszernego kursu, omawiającego wszystkie podstawy gita po polsku, dostępnego na [NA YOUTUBE](#).

Jak już wiesz z powyższego kursu, repozytoria z naszym kodem najczęściej przechowujemy na serwerach zdalnych (remotes), które zapewniają ich dostępność 24 godziny na dobę.

Dla nabrania wprawy [przerób jeszcze jeden krótki tutorial](#), o zakładaniu repozytorium na GitHubie :)

UWAGA!

Od teraz każde następne ćwiczenie czy kurs przygotowujący wykonuj w publicznym repozytorium na GitHubie!

Dzięki temu jeszcze przed zajęciami wyrobisz sobie nawyk pracy z gitem i udostępnisz swój kod światu! Nie martw się, jeśli to nie będzie kod, którym chcesz się pochwalić, na Twoim etapie najbardziej liczy się regularność kodowania, a nie idealnie napisany kod!

System kontroli wersji - GitHub

Na sam koniec zbiór dobrych porad, które warto wdrożyć:

1. nazwij swój profil imieniem i nazwiskiem
2. wstaw zdjęcie
3. uzupełnij profil
 - podepnij link do LinkedIn
 - dodaj miasto
 - w "Jobs profile" zaznacz, że szukasz pracy
4. czytelnie nazwij repozytorium, dodaj krótki opis
5. dbaj o regularne commitowanie swojej pracy (szczególnie po kursie)
6. pisz krótki opis swoich repozytoriów na GitHubie (description)
7. opisuj swoje projekty w pliku README.md
8. dbaj o jakość kodu - to Twoja wizytówka

[OPCJONALNIE] Markdown

Rozszerzenie .md (np. w pliku README.md) to rozszerzenie Markdown, czyli narzędzia do pisanie tekstu, które zostało stworzone w celu jak największego uproszczenia tworzenia i formatowania tekstu.

Markdown jest bardzo często używany w dokumentacjach i najróżniejszych opisach kodu. Warto zapoznać się z jego podstawami. Możesz to zrobić w 10 minut [TUTAJ](#). Jeśli chcesz pominąć wstęp (co to jest Markdown) możesz zacząć z [TEGO MIEJSCA](#).

W [TYM MIEJSCU](#) możesz sprawdzić swoje umiejętności.

HTML/CSS

Jeśli tworzenie stron WWW nie jest Ci obce, a czas goni, wiesz co to HTML i czym różni się jego wersja 4 od 5 oraz potrafisz używać selektorów CSS - nie tylko tych prostych, ale np. selektorów atrybutów czy `:nth-child()` lub `:not()`, możesz pominąć poniższą treść i od razu sprawdzić swoje umiejętności w teście na dole tego punktu.

Pierwszym krokiem do zostania front-end developerem jest bezbłędne przyswojenie wiedzy z zakresu HTML5 i CSS3. Te technologie są wykorzystywane bezwzględnie wszędzie i musisz je znać jeszcze przed rozpoczęciem kursu. Podczas kursu są przewidziane zajęcia z czystego HTML-a i CSS-a, jednakże w niewielkiej ilości, tak by utrwalić i wyrównać wiedzę w grupie.

UWAGA!

Program kursu zakłada, że podstawy (jak podstawowe tagi HTML, czy selektory CSS) opanujesz podczas tego przygotowania. Z tego zakresu na kursie zostaną omówione dobre praktyki i trudniejsze przypadki takie jak responsywność, media queries i grid.

HTML/CSS

Kolejnym zadaniem jest przerobienie czterech kursów na YouTube.

Pierwszy z nich omawia HTML. Trwa około 2 godziny i jest podzielony na krótkie średnio 7-minutowe filmiki. Zalecamy założenie nowego repozytorium na GitHubie i kodowanie razem z instruktorem w Visual Studio Code na bieżąco oraz commitowanie po każdym z filmików.

[LINK DO KURSU](#)

Kolejny kurs bardzo dokładnie omawia CSS rozpoczynając od prostych selektorów, a kończąc na tych bardziej specyficznych i złożonych. Ten kurs trwa niecałe 6 godzin i również jest podzielony na krótkie średnio 6-minutowe filmiki. Pamiętaj o nowym repozytorium i commitach!

[LINK DO KURSU](#)

HTML/CSS

Dwa kolejne kursy omawiają stosunkowo nowe zagadnienia w CSS - flexbox i CSS grid.

Kurs flexbox trwa niewiele ponad godzinę i jest podzielony na krótkie średnio 6-minutowe filmiki. Nie zapomnij o repozytorium :)

[LINK DO KURSU](#)

Kurs CSS grid również jest krótki, trwa niecałe półtora godziny i jest podzielony na krótkie średnio 8-minutowe filmiki. Po raz kolejny przypominamy o nowym repozytorium ;)

[LINK DO KURSU](#)

HTML/CSS

Na koniec (i trochę powtórkowo) sprawdź swoją wiedzę z HTML i CSS w interaktywnym kursie z CodeCademy:

[LINK DO KURSU HTML](#)

[LINK DO KURSU CSS](#)

Najczęściej podczas pisania kodu, czy to w CSS, czy w JavaScriptcie będziesz korzystał z selektorów, odnoszących się do tagów, klas, identyfikatorów czy nawet atrybutów elementów! Jest to kluczowa wiedza z tej części przygotowania! Zweryfikuj ją interaktywnie poniżej:

[LINK DO "WERYFIKATORA"](#)

HTML/CSS

Zweryfikuj również swoją wiedzę z Flexboxa, CSS Grida rozwiązując 3 interaktywne mini-gry:

[Flexbox Foggy](#)

[Flexbox Defense](#)

[Grid Garden](#)

UWAGA! Nawet jeśli pominąłeś treść punktu HTML/CSS, koniecznie wykonaj powyższy test i minigry!

UWAGA!

Kolejny rozdział czyli JavaScript jest najważniejszą umiejętnością front-end developera! Nawet jeśli zdarzyło Ci się tworzyć wcześniej strony WWW i korzystać z JS czy jQuery KONIECZNIE przerób dokładnie poniższe materiały! Gdy wejdiesz w drugą część kursu, gdzie będziesz uczyć się pisać webowe aplikacje w React (czyli bibliotece/rozszerzeniu JavaScriptu), nie będzie miejsca i czasu na żadne braki z tego zakresu!

JavaScript

Naukę JavaScriptu zaczniemy od pierwszej części bardzo ciekawej serii książek. Ciekawej nie tylko ze względu na merytoryczną zawartość, ale i na sposób jej wydania. Książka ta została przez autora bezpłatnie opublikowana na GitHubie, gdzie inni użytkownicy mogli zgłaszać swoje uwagi i dawać wkład w jej rozwój!

Książka dostępna [TUTAJ](#) (do przeczytania jest około 50 stron), a jej polską wersję można nabyć za kilkanaście złotych [TUTAJ](#).

UWAGA! Zakończ lekturę tej książki na Rozdziale 2 -> Funkcje jako wartości -> Domknięcie (Chapter 2 -> Functions As Values -> Closure). Dalsze tematy wykraczają ponad podstawową wiedzę, jaką musisz pozyskać przed kursem!

JavaScript

Kolejnym krokiem jest zrobienie kursu na YouTube, dobrze znanego Ci już autora :) Kurs trwa niecałe 5 godzin i jak zwykle podzielony jest na kilkuminutowe filmiki.

UWAGA! KONIECZNIE stwórz repozytorium i commituj swoją naukę - kod JavaScriptowy na GitHubie to dla rekrutera zawsze dobra wiadomość!

[LINK DO KURSU](#)

JavaScript

Zweryfikuj swoją wiedzę i powtórz zagadnienia rozwiązując interaktywne zadania na edabit.com:

Zacznij od zadania wprowadzającego:

- [Zadanie 1](#)

Następnie wykonaj zadania polegające na napisaniu prostych funkcji:

- [Zadanie 2](#)
- [Zadanie 3](#)
- [Zadanie 4](#)

Wykonaj również zadania w oparciu o tablice danych:

- [Zadanie 5](#)
- [Zadanie 6](#)
- [Zadanie 7](#)

Dokumentacja

Podczas wykonywania ćwiczeń, jak i na kursie na pewno będziesz szukał odpowiedzi na tematy, których jeszcze nie umiesz rozwiązać lub definicji, których nie zdołałeś zapamiętać w całości. Oczywiście jest to naturalne i żaden developer nie zna na pamięć wszystkich funkcji i metod, a często korzysta z dokumentacji :)

Bardzo dobre dokumentacje z zakresu HTML/CSS od Mozilli znajdują się [POD TYM ADRESEM](#).

Świetnym miejscem na odnajdywanie odpowiedzi na bardziej złożone kwestie jest [STACK OVERFLOW](#).

Masz pytania?



prework@infoShareAcademy.com

www.infoshareacademy.com

UWAGA! Pamiętaj, żeby w mailu podać nazwę kursu, w którym bierzesz udział!
Najlepiej zatytułuj maila w formacie:

„PYTANIE — Twoje imię i nazwisko | nazwa kursu | data startu”