

C# PREWORK

infoShare
ACADEMY

Spis treści

1. [Powitanie i podstawowe informacje](#)
2. [Sposób “oddania” ukończonego preworka](#)
3. [C# i .NET wprowadzenie](#)
4. [Przygotowanie środowiska](#)
5. [Visual Studio](#)
6. [Przygotowanie merytoryczne](#)
 - [System kontroli wersji - GIT](#)
 - [HTML/CSS + Bootstrap](#)
 - [C#](#)
 - [SQL](#)
7. [Dokumentacja](#)
8. [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 cs.prewrite.is-academy.pl, by przesłać nam:

1. Screenshot z zainstalowanym Visual Studio na systemie Windows.
2. Screenshot z wykonanymi ostatnimi etapami ćwiczeń z C# wykonanymi w przeglądarce.
3. Screenshot z rozwiązaniem piątym ćwiczeniem z kursu SQL ze strony sqlbolt.com.
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**”

C# i .NET - wprowadzenie

C# i .NET - wprowadzenie

Czytając to opracowanie często zauważysz odwołanie do **.NET**, poniżej małe wyjaśnienie

- .NET - platforma do uruchamiania kodu od Microsoft
- C# - język od Microsoft, najczęściej uruchamiany na platformie .NET

W dokumencie może pojawić się więcej nowych dla Ciebie zagadnień, jeśli są Ci obce - nie bój się wyszukać w internecie co się za nimi kryje jeśli nasz opis nie wyczerpał twojej ciekawości :)

Przygotowanie środowiska

Przygotowanie środowiska

W pierwszej kolejności należy przygotować system operacyjny, na jakim będziemy pracowali przez cały kurs.

Przy pracy ze środowiskiem .NET preferuje się pracę na systemie Windows.

Zalecana wersja systemu Windows to najnowszy **Windows 10**.

Może być to wersja **Home** lub **Professional**, nie ma to wpływu na większość funkcjonalności, które wykorzystujemy na co dzień.

Używasz innego systemu niż Windows?

Na jednym komputerze może być zainstalowana dowolna ilość systemów działających „obok siebie”.

Instalacja systemu operacyjnego to jednak proces, który wymaga przynajmniej podstawowej wiedzy o działaniu: komputera, systemów operacyjnych i tzw. bootloadera - programu służącego do załadowania systemu operacyjnego.

Na szczęście istnieją łatwiejsze drogi :)

Używasz innego systemu niż Windows?

OPCJA A Możemy na naszym komputerze stworzyć tak zwaną maszynę wirtualną, czyli program komputerowy, działającym na Twoim aktualnym systemie, symulujący komputer. To rozwiązanie jest polecane dla Linuxów, ale zadziała również po macOS. Możesz zainstalować Windows w ten sposób korzystając z poradnika. [Instalacja przez Virtualbox](#)

OPCJA B Twórcy macOS stworzyli swój indywidualny sposób instalacji Windowsa obok ich systemu. Nie zadziała on na Linuxie. Jest rekomendowany jeśli używasz sprzętu firmy Apple. [Równoległa instalacja przez Boot Camp](#)

W obu poradnikach jest link, ale umieszczamy go jeszcze raz - Windowsa możesz pobrać [TUTAJ](#).

Instalacja oprogramowania

1. W pierwszej kolejności pobierz i zainstaluj [Chrome](#)) - przeglądarkę, z której będziemy korzystać podczas kursu. Jeśli aktualnie używasz dowolnej innej przeglądarki internetowej - nie szkodzi, możesz mieć ich zainstalowaną dowolną ilość.
2. Pobierz i zainstaluj również [Slacka](#) - komunikator dla zespołów, który będziemy wykorzystywać podczas pracy.

Visual Studio

Visual Studio

Kolejnym krokiem jest instalacja [Visual Studio](#) czyli IDE, w którym będziesz pisać swój kod.

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

Visual Studio jest najpopularniejszym narzędziem jeśli chodzi o wytwarzanie oprogramowania w C# oraz pod platformę .NET.

Visual Studio

Być może zastanawiasz się, czym różni się IDE od edytora tekstu, takiego jak choćby podstawowy Notatnik, czy edytora kodu jak SublimeText, Atom, Brackets czy VS Code?

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).

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, na przykład z systemami kontroli wersji.

Visual Studio - Instalacja

Będziemy wykorzystywać **Visual Studio 2019 Community Edition** - w pełni funkcjonalną oraz wartościową wersję do użytku domowego oraz komercyjnego.

Instalator dla Windows można znaleźć pod [tym linkiem](#). Podczas instalacji wybierz pakiety zaznaczone na dwóch kolejnych slajdach,

Istnieje również wersja dla macOS jednak jest to oddzielne oprogramowanie, które NIE JEST zamiennikiem wersji pod system Windows!

Na kursie będziemy korzystać **wyłącznie** z wersji pod Windows.

Workloads Individual components Language packs Installation locations

Windows (3)



.NET desktop development

Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.



Desktop development with C++

Build Windows desktop applications using the Microsoft C++ toolset, ATL, or MFC.



Universal Windows Platform

Create Windows apps that run on Windows 10, Windows 10 IoT, and Windows 10 Mobile. Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#. with C#, VB, JavaScript, or optionally C++.



Web & Cloud (7)



ASP.NET and web development

Build web applications using ASP.NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker support.



Azure development

Azure SDKs, tools, and projects for developing cloud apps, creating resources, and building Containers including...



Python development

Editing, debugging, interactive development and source control for Python.



Node.js development

Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.



Data storage and processing

Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.



Data science and analytical applications

Languages and tooling for creating data science applications, including Python, R and F#.



Office/SharePoint development

Create Office and SharePoint add-ins, SharePoint solutions, and VSTO add-ins using C#, VB, and JavaScript.



Mobile & Gaming (5)



Mobile development with .NET

Build cross-platform applications for iOS, Android or Windows using Xamarin.



Game development with Unity

Create 2D and 3D games with Unity, a powerful cross-platform development environment.



Mobile development with JavaScript



Mobile development with C++



Installation details

- > Visual Studio core editor
- > ASP.NET and web development
- > Azure development
- > Data storage and processing
- ✓ .NET Core cross-platform development
 - Included
 - ✓ .NET Core 2.1 development tools
 - ✓ .NET Framework 4.6.1 development tools
 - ✓ ASP.NET and web development tools prerequisites

Optional

- ✓ Cloud tools for web development
- ✓ .NET profiling tools
- ☐ Developer Analytics tools
- ✓ Web Deploy
- ✓ .NET Core 2.0 development tools
- ☐ .NET Core 1.0 - 1.1 development tools for Web

Other Toolsets (3)



Visual Studio extension development

Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.



Linux development with C++

Create and debug applications running in a Linux environment.



.NET Core cross-platform development

Build cross-platform applications using .NET Core, ASP.NET Core, HTML/JavaScript, and Containers including Docker...



Edytor tekstu

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

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

Aby zainstalować git 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](#).

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 pisania 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 + Bootstrap

Z racji, że będziesz testować aplikacje webowe (takie, które działają w przeglądarce) nie ominie Cię podstawowa wiedza z technologii odpowiadających za wyświetlanie ich treści.

Zapoznaj się z podstawami HTML i CSS podczas [kursu przygotowanego przez Udacity.](#)

Bardzo popularnym narzędziem do szybkiego tworzenia stron WWW jest biblioteka Bootstrap. Zobacz jak się z niego korzysta [podczas krótkiego kursu na YouTube.](#)

C#

Język C# to podstawa Twojej pracy jako backend developera.

Na poznaniu tej technologii powinieneś/powinnaś skupić się najbardziej!

Na kolejnej stronie znajdziesz linki z interaktywnymi ćwiczeniami lub przykładami, które uruchomisz bezpośrednio w przeglądarce, oraz ćwiczeniami do uruchomienia na twojej maszynie.

C#

1. Microsoft, twórca C#, przygotował zestaw interaktywnych przykładów które możesz uruchomić w przeglądarce a które pozwolą Ci zapoznać się z podstawami języka. Znajdziesz je pod [tym linkiem](#).
2. Rozwinięciem przykładów z punktu 1. są interaktywne ćwiczenia, które również uruchomisz w przeglądarce pod [linkiem](#). Przerób wszystkie z nich!
3. Na zakończenie wykonaj poprzednie ćwiczenia oraz jedno dodatkowe na swoim komputerze w Visual Studio, wszystko znajdziesz [opisane w tym miejscu](#). Dzięki temu zapoznasz się z IDE oraz uruchomisz pierwszy projekt na komputerze.

Nie zapomnij o pracy na repozytorium GIT podczas wykonywania ćwiczeń! :)

SQL

Back-end developerzy w codziennej pracy mają do czynienia z bazami danych. W tej kwestii od wielu lat niezmiennie najpopularniejszym językiem zapytań jest SQL. W projektach .NET najczęściej wykorzystuje się popularną bazę danych od Microsoftu - MS SQL.

By sprawnie pisać aplikacje podczas kursu, należy poznać jego podstawy. Bardzo dobry interaktywny kurs można znaleźć na sqlbolt.com. Przerób 5 pierwszych ćwiczeń z tego kursu.

Warto również obejrzeć te dwa paro-minutowe nagrania:

- [Learn SQL Programming - 1 - Intro to SQL - YouTube](#)
- [Learn SQL Programming - 2 - Filter Queries With WHERE - YouTube](#)

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](#).

Tematy związane z C# oraz platformą .NET to ogromna baza [Microsoft Docs](#).

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”