

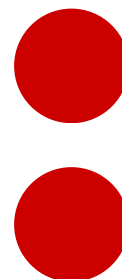


CODE:ME
UCZYMY PROGRAMOWANIA

Jak zacząć uczyć się programowania

Od 2016 na rynku

Jak zacząć uczyć się programowania?



Wokół początków nauki programowania i szerzej – dróg wejścia do świata IT – narosło wiele mitów i przekonań, które albo nadmiernie podsycają entuzjazm, odrzucając zdrowy rozsądek i racjonalne myślenie, albo wręcz przeciwnie, napawają strachem i zniechęcają nawet bardzo zdeterminowanych, by w ten świat wejść.

Ten artykuł, będąc pierwszym z serii, jest próbą racjonalizacji i zdroworozsądkowego podejścia do nauki programowania samego w sobie oraz garścią dobrych rad dla tych, którzy wiedzą już, że chcą spróbować swoich sił w nauce programowania, ale nie mają pojęcia, od czego zacząć.

Złe dobrego początki

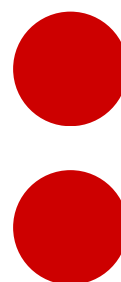
Początki nie są łatwe. Nigdy. Może tego nie pamiętasz, ale jazda na rowerze na dwóch kółkach na początku nie była taka prosta. Początki bolą, niekiedy dosłownie. Często potrzebujemy też wsparcia, na przykład w postaci osoby dorosłej trzymającej przyczepiony do bagażnika kij. W bólach i trudach, nie od razu, ale w końcu udaje się utrzymać równowagę i przejechać pierwszych kilka metrów bez wywrotki. Co za radość!

Zapytasz - *Tylko co ma jazda na rowerze do nauki programowania?*
Odpowiedź jest bardzo prosta – **złe dobrego początki.**

Na początku nauki czegokolwiek nawet instrukcje wydają się trudne, a operowanie narzędziami, czymkolwiek by one nie były, jest wyczerpujące i toporne. Frustracja rośnie, pierwsze efekty są mierne, a niezrozumiane przy pierwszych próbach zasady odbijają się czkawką. Tak właśnie jest też z programowaniem, tym bardziej, jeśli nie masz za sobą technicznego (najlepiej informatycznego lub podobnego) wykształcenia, gdzie w ten czy inny sposób miałeś/aś okazję zapoznać się z choćby podstawowymi pojęciami. Na początku trudno przebić się przez całkiem nowy język, którym operują sprawnie i bez zająknięcia doświadczeni programiści. Czytasz ten artykuł prawdopodobnie dlatego, że chcesz uczyć się programowania, ale zagubiwszy się w gąszczu dziwnych słów właściwie już nie wiesz, od czego zacząć i czy to w ogóle jest możliwe bez podpisywania paktu z siłami nadprzyrodzonymi.

Jesteś w dobrym miejscu! Mam nadzieję, że powyższe akapity przekonały cię, że nauka programowania to nie czarna magia, tylko dziedzina wiedzy, jak każda inna. W dalszej części spróbujemy odpowiedzieć na dwa najczęściej zadawane przez aspirujących programistów pytania: jak zacząć naukę i jaki język programowania wybrać.

I to nie będą jedynie słuszne odpowiedzi, tylko wnioski wyniesione z własnego doświadczenia i niezliczonej ilości rozmów tak z przyszłymi, jak i doświadczonymi programistami.



Krok 1. Dowiedz się, co naprawdę lubisz

Programowanie to pojęcie tak obszerne, jak biologia, a i nauk pobocznych ma równie dużo. Naprawdę. Jest biologia zwierząt, biologia roślin, mikrobiologia i różne inne biologie, których nazw prawdopodobnie nie potrafilibyśmy nawet wymówić. Wbrew pozorom początki poznawania jednej i drugiej dziedziny są bardzo podobne.

W każdym z dwóch przypadków zaczynasz od wiedzy ogólnej, od zorientowania się, co w trawie piszczy i z czym w ogóle masz do czynienia. Biologii uczymy się już w szkole podstawowej, więc jest trochę łatwiej, z technologią dawniej bywało różnie, więc całą wiedzę prawdopodobnie zdobyłeś/aś na własną rękę po prostu czytając o nowych smartfonach czy kolejnych wersjach Ubuntu. Przeglądając w Internecie czy prasie różne treści siłą rzeczy wybieramy te tytuły, które nas najbardziej interesują, oczywiście jeśli robimy to z własnej, nieprzymuszonej woli. Dzięki temu mamy na starcie większe szanse na sukces niż osoby, które dany temat zgłębiają z przymusu. O wiele łatwiej pokonać trudności, gdy sam proces sprawia nam przyjemność, a cel jawi się jako coś wspaniałego, niż jeśli na samą myśl o wykonaniu jakiegokolwiek prac związanej z tym zagadnieniem nie chce nam się wstać z łóżka.

Dlatego właśnie, że chętniej i lepiej wykonujemy rzeczy, jeśli je lubimy, zanim zaczniesz szukać w Google najlepszego tutorialu „jedyne go słusznego języka programowania”, zastanów się, która z dziedzin nowych technologii sprawia ci największą frajdę.

PROTIP

Podczas poszukiwań tego, co cię interesuje, pamiętaj o jednym – świat IT to nie tylko programowanie sensu stricto.

Ktoś te wszystkie aplikacje i systemy musi zaprojektować, i to nie tylko, żeby były ładne (**grafik**), ale i użyteczne (**UX/UI**).

Gdy sam produkt jest już stworzony lub jakaś jego zamknięta część, przydałby się ktoś, kto najzwyczajniej w świecie sprawdzi, czy to działa (**testerzy manualni i automatyczni**) i nie jest to tak proste zadanie, jak mogłoby się wydawać.

Krok 2. Podejmij decyzję, w którą stronę chcesz iść

Jeśli już wiesz, co Ci się podoba, czas wybrać odpowiednią ścieżkę nauki. Nie jest to proste zadanie, tak jak nie są łatwe początki nauki gry na gitarze. To drugie jest o tyle łatwiejsze, że podstawowych rodzajów gitar mamy zaledwie kilka, podczas gdy w przypadku programowania tych ścieżek jest o wiele więcej. By ułatwić Ci decyzję i podpowiedzieć, czym na początku możesz się zainteresować, podrzucamy kilka podstawowych pojęć, na które możesz się natknąć przeszukując Internet w kierunku nauki programowania (i nie tylko):

Front-end wszystko to, co w aplikacji widzi użytkownik. Nie tylko rozłożenie ikon i przycisków, ale również “warstwa logiczna”, czyli np. sposoby pobierania danych od użytkownika i przekazywanie ich do back-endu

Back-end “zaplecze aplikacji”, czyli wszystko to, czego użytkownik nie widzi, a działa. Sposób przechowywania danych, operacje na serwerach i procesy odpowiedzialne za poprawne działanie aplikacji.

Full Stack front-end i back-end w jednym. Full Stack developer to osoba, która jest biegła zarówno w tworzeniu front-endu, jak i back-endu.

Mobile potoczna nazwa technologii mobilnych, czyli działających na urządzeniach mobilnych/przenośnych, takich jak smartfony i tablety.

Jeśli już wiesz, co Ci się podoba, czas wybrać odpowiednią ścieżkę nauki. Nie jest to proste zadanie, tak jak nie są łatwe początki nauki gry na gitarze. To drugie jest o tyle łatwiejsze, że podstawowych rodzajów gitar mamy zaledwie kilka, podczas gdy w przypadku programowania tych ścieżek jest o wiele więcej.

By ułatwić Ci decyzję i podpowiedzieć, czym na początku możesz się zainteresować, podrzucamy kilka podstawowych pojęć, na które możesz się natknąć przeszukując Internet w kierunku nauki programowania (i nie tylko):

Front-end wszystko to, co w aplikacji widzi użytkownik. Nie tylko rozłożenie ikon, przycisków, ale również “warstwa logiczna”, czyli np. sposoby pobierania danych od użytkownika i przekazywanie ich do back-endu

Back-end “zaplecze aplikacji”, czyli wszystko to, czego użytkownik nie widzi, a działa. Sposób przechowywania danych, operacje na serwerach i procesy odpowiedzialne za poprawne działanie aplikacji.

Full Stack front-end i back-end w jednym. Full Stack developer to osoba, która jest biegła zarówno w tworzeniu front-endu, jak i back-endu.

Mobile potoczna nazwa technologii mobilnych, czyli działających na urządzeniach mobilnych/przenośnych, takich jak smartfony i tablety.

UX designer – osoba odpowiedzialna za użyteczność aplikacji czy systemu, czyli, prościej rzecz ujmując, żeby tym, którzy będą używać danej aplikacji dobrze wygodnie się z nie z korzystało.

Tester to osoba, która sprawdza, czy aplikacja/system działa poprawnie, wydajnie, wedle życzeń klienta, założeń projektu etc. Testy dzielą się na manualne, gdzie nie jest wymagana umiejętność programowania w żadnym języku, i automatyczne, wymagające znajomości konkretnych technologii do pisania skryptów.

Pamiętaj, że lista profesji w IT jest o wiele dłuższa!

Jak widzisz, nie wszystkie wyżej wymienione pojęcia dotyczą tylko wyłącznie pisania kodu. Już wspominaliśmy wyżej, że IT to nie tylko kodowanie, ale i szereg różnorodnych zadań i specjalizacji. W dalszej części artykułu skupimy się już stricte na programowaniu i związanymi z tym pojęciami, jednak chcielibyśmy pokazać Ci, że, naszym zdaniem wspomniały, świat IT jest o wiele bogatszy i ciekawszy. Domyślamy się, że zapewne o tym wszystkim wiesz i dlatego właśnie planujesz rozwijać swoją karierę w tym właśnie kierunku, jednak często umykają nam pewne pojęcia, gdy skupiamy się tylko na jednym temacie. O UX designerach i testerach opowiemy innym razem. Jeśli zaś wiesz, że to, co chcesz robić, to pisanie kodu, zapraszamy do kolejnej części artykułu i zarazem kolejnego kroku.



Krok 3. Wybierz język programowania

Jeżeli już wiesz, co ci się podoba i potrafisz to nazwać, o wiele łatwiej jest znaleźć język programowania, od którego najlepiej by było zacząć. Języki programowania są różne, bo służą innym celom, odpowiadają na różne potrzeby i mają zróżnicowane poziomy skomplikowania. Wiąże się to wszystko z ich zastosowaniem w rzeczywistych projektach. Gdy tworzysz prostą stronę internetową, użyjesz czegoś innego niż gdy potrzebujesz skomplikowanej aplikacji obsługującej niezliczone ilości danych. Inaczej też podejdziesz do wyboru języka, gdy Twoim celem jest stworzenie aplikacji mobilnej.

To tak, jak w ogrodnictwie – jeśli chcesz posadzić trzy kwiatki, będziesz potrzebować małej łopatkę i to powinno wystarczyć. Jeśli zaś masz w planach położenie instalacji nawadniającej trawnik potężnej rezydencji, lepszym rozwiązaniem będzie koparka i kilku ludzi do pomocy.

Wybierając pierwszy język, którego chcesz się nauczyć, warto spojrzeć nie tylko na cechy i zastosowanie, ale również na jego popularność. Im częściej jest używany, tym więcej znajdziesz materiałów do jego nauki i informacji o nim w ogóle, a w końcu pojawia się w większej liczbie ofert pracy, niż języki i technologie mniej znane.

Kolejną rzeczą, na którą koniecznie musisz zwrócić uwagę, to odpowiedź na poprzednie kroki – **co cię interesuje** i **co chcesz robić**, bo temu właśnie ma służyć język, którego zaczniesz się uczyć. To jak z wyborem nauki języka obcego (oprócz angielskiego, ten się zawsze przyda).

Dla podpowiedzi, czym możesz się zainteresować, podrzucamy listę popularniejszych języków programowania i nie tylko:

HTML przeważnie łączony z **CSS**

To podstawowe języki Internetu. Właśnie przy użyciu HTML i CSS najczęściej powstają proste strony internetowe i szkielety większych aplikacji. To dobry punkt wyjścia, jeśli myślisz o drodze front-endowca. Z HTMLem ułożysz elementy na stronie, natomiast CSSem nadasz im pożądany wygląd*.

uwaga: HTML nie jest językiem programowania, to hipertekstowy język znaczników. Języki, w których nie da się realizować obliczeń, w tym języki znaczników, nie są uznawane za języki programowania

JavaScript (*nie mylić z Javą!*)

Głównie język front-endowy, choć, dzięki NODE.JS, coraz szerzej wykorzystywany jest w back-endzie. Wszędzie tam, gdzie macie do czynienia z interaktywnymi aplikacjami webowymi (internetowymi), przyda się JavaScript.

PHP

Język wykorzystywany po stronie serwera, a więc w back-endzie, służący do budowania serwisów internetowych. Obecnie mówi się, że staje się coraz mniej popularny, jednak w związku z jego szerokim zastosowaniem w przeszłości, jeszcze długo będzie należał do tych częściej używanych.

JAVA

Jeden z najpopularniejszych (obok C#) i najwszechstronniej wykorzystywanych języków programowania. W Javie możemy tworzyć aplikacje zarówno serwerowe, desktopowe, jak i mobilne. Spotkacie ją nawet w swojej pralce czy mikrofalówce! Używany jest głównie przez duże korporacje. Należy do języków, które mają zarówno swoją stronę front-endową, jak i back-endową, jednak specjalizacja w tym przypadku nie przychodzi na początku nauki, a dopiero długo po jej rozpoczęciu.

C#

Drugi obok Javy język, któremu zaufały duże korporacje. Podobnie jak Java, jest używany i we front-endzie, i w back-endzie. Rozwijany i używany we frameworku Microsoftu .NET służy do tworzenia aplikacji na platformę Windows, jak również gier komputerowych i aplikacji desktopowych. **Java i C#** są dobrym początkiem, jeśli w przyszłości chcesz tworzyć aplikacje mobilne.

Python

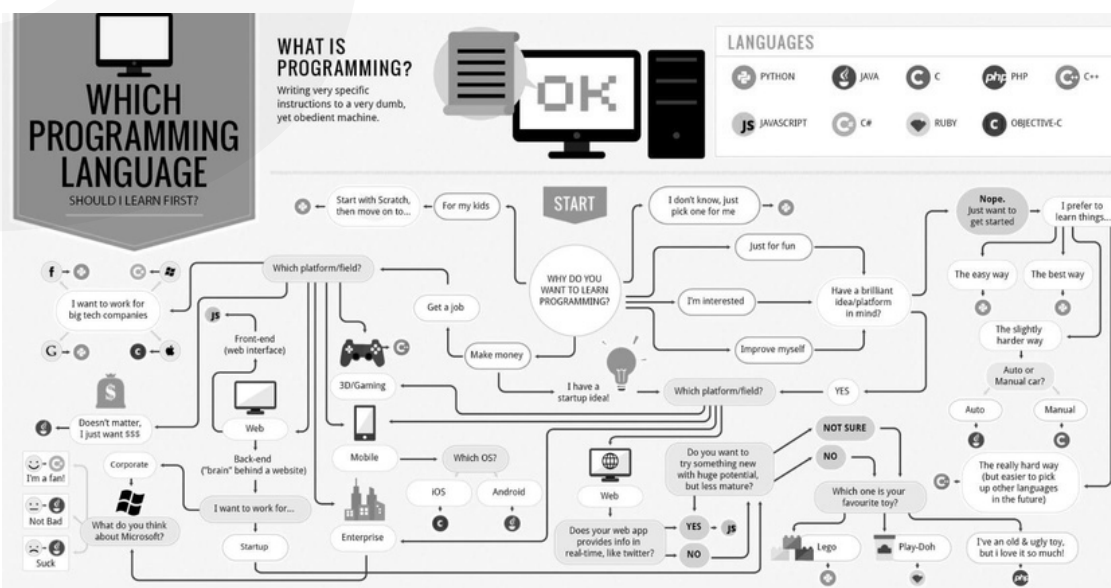
Jeden z ulubionych języków akademickiej społeczności. Wykorzystywany zarówno w back-endzie, jak i front-endzie. Dobry wszędzie tam, gdzie zależy nam na sprawnym przetwarzaniu i analizowaniu danych. Jest również szeroko wykorzystywany do tworzenia aplikacji webowych i sieciowych. Inne popularne, za to trudniejsze do nauki języki programowania, używane do zadań specjalnych:

C

Powszechnie uważany za podstawowy język programowania, bardzo szybki, jednak stosunkowo trudny. Wykorzystywany m.in. przy tworzeniu systemów operacyjnych i wszędzie tam, gdzie liczy się szybkość wykonywania..

C++

Język ten, łącząc prędkość C z obiektowością i samodzielnym zarządzaniem pamięcią, jest narzędziem do zadań specjalnych, takich jak silniki gier komputerowych czy skomplikowane aplikacje mobilne i desktopowe.



Źródło: <https://carlcheo.com/startcoding>

W Internecie znajdziesz mnóstwo porad i infografik, które mogą ci pomóc wybrać język programowania, od którego możesz zacząć naukę, jednak bądź ostrożny/a! Infografika po lewej w zabawny sposób pokazuje, jakie pytania możesz sobie zadać podczas rozważań o tym, czym chcesz się zajmować.

Pamiętaj, żeby z dystansem traktować zawarte w zagranicznych materiałach informacje o zarobkach, a czasem nawet popularności danego języka. Jeśli planujesz na początek karierę w Polsce, poszukaj możliwie jak najnowszych statystyk dotyczących naszego kraju i sprawdź ogłoszenia o prace.

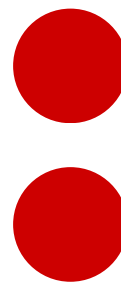
Krok 4. Trzymaj się swojego wyboru

Wykonując uczciwie wszystkie poprzednie kroki, powinieneś/aś mieć już za sobą decyzję, od którego języka zaczniesz swoją przygodę z programowaniem. Teraz najważniejsze cokolwiek wybrałeś/aś, trzymaj się tego. Najpierw naucz się dobrze jednego języka, potem próbuj swoich sił w kolejnych, a wierz nam, że będzie dużo łatwiej niż z tym pierwszym. Konsekwencja w nauce jednego języka programowania naraz i poznanie go naprawdę dobrze, jest bardzo ważna.

To jak z nauką języków obcych. Jeśli poznasz podstawy francuskiego, potem przeczucisz się na rosyjski, by wreszcie poznać kilka zwrotów po arabsku – jedyne, co będziesz potrafić, to powiedzenie „Cześć, jestem Ania, lubię ziemniaki”, ale w żadnym z tych języków swobodnie nie porozmawiasz. W programowaniu podobnie – podstawy dadzą ci możliwość napisania kilku funkcji, ale by naprawdę pisać aplikacje, musisz znać ten język.

Nawet jeśli przyjdą chwile zwątpienia, czy podjęty wybór był słuszny, zignoruj je i rób dalej swoje. Jeśli poznasz ten jeden język bardzo dobrze, nauka każdego kolejnego przyjdzie ci o wiele łatwiej, a przeskakiwanie z jednego języka na inny sprawi, że nie skupisz się dobrze na żadnym.





Podsumowując!

Jeśli chcesz nauczyć się programować, musisz pamiętać, że to nauka, jak każda inna – będziesz potrzebować czasu, samozaparć i konsekwencji w działaniu. Nie da się wszystkiego od razu, niestety. Sam wybór języka na początek może wydawać się na początku skomplikowany, ale mamy nadzieję, że tych kilka rad i wytłumaczonych pojęć pomoże ci lepiej się do tego zabrać.

Lepiej dla ciebie, bo pamiętaj, że w tym wszystkim najważniejsze jest to, co ty chcesz robić w życiu.

Jeśli zaś dalej nie wiesz, czym się zająć, nie możesz się zdecydować lub w otchłani Internetu znalazłeś/aś tematy, które bardzo cię zainteresowały, ale nie masz pojęcia, jak się zabrać za naukę tych zagadnień – **napisz do nas**, a spróbujemy rozwiązać twoje wątpliwości i podpowiedzieć ci najlepszą dla ciebie drogę nauki.

Pierwsze kroki w IT

Jeżeli nadal masz pewne obawy, czy zdecydować się na kurs u nas, czy odważyć się i zacząć zmiany w swoim życiu, mamy coś ekstra- cykl wywiadów z naszymi Absolwentami:



Pierwsze kroki w IT

"Osobom, które jeszcze nie zdecydowały się iść na kurs z prawdziwego zdarzenia, radzę żeby przestały się zastanawiać i to zrobić. Nie będą żałować. Inwestycja w siebie to najlepsza inwestycja."

Wojtek Wiśniewski
absolwent kursu:
Podstawy testowania w praktyce
w CODE:ME

CODE:ME
UCZYMY PROGRAMOWANIA

[Link do artykułu - kliknij aby przenieść się do artykułu](#) ✨



Pierwsze kroki w IT

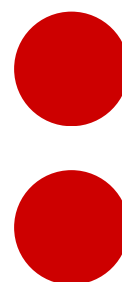
"...w Internecie mamy ogrom wiedzy o różnych technologiach, wiele kursów, blogów, poradników, ćwiczeń ale w pewnym momencie warto jednak trafić na mentora, który zbierze ważne i sensowne informacje w jedno i podpowie, co jest przydatne w realnej pracy, a czym sobie nie warto zaprzętać głowy..."

Katarzyna Sołdańska
absolwentka kursu:
Data Science w CODE:ME

CODE:ME
UCZYMY PROGRAMOWANIA

[Link do artykułu - kliknij aby przenieść się do artykułu](#) ✨

Mini słowniczek



404 - jeden z najpopularniejszych komunikatów o błędzie. Często oznacza on, że dany link jest uszkodzony, nieaktywny lub niepoprawny

Atrybut - element składni języka programowania, który określa konkretną właściwość (znaczenie), nadaną wybranemu elementowi (obiektowi)

Bazy danych - służą do przechowywania danych np. użytkowników, mogą być również spisem wszystkich produktów w sprzedaży

Bug – błąd, który może spowodować niepoprawne działanie programu

Code review – etap, w którym inny programista przegląda Twój kod i sugeruje ewentualne poprawki

DevOps - skrót od Development Operations – nazwa ta oznacza jedną z metod tworzenia oprogramowania, kładącą nacisk na ścisłą współpracę administratorów oraz programistów.

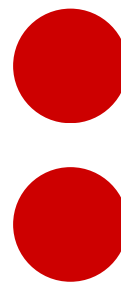
Favicon - inaczej favourite icon / ulubiona ikona, która pojawia się w zakładce przeglądarki internetowej, gdy otworzymy jakąś stronę.

GIT - miejsce w którym są przechowywane wszystkie wersje Twojego kodu. Dzięki temu jeśli coś się popsuje w kolejnej wersji, można bezpiecznie wrócić do poprzedniej edycji

Jira/TFS itd.– różne narzędzia służące do zarządzania zadaniami w zespołach projektowych

Meta Tag - dodatkowe informacje o stronach www lub elementach – przykładowo może to być sposób wyświetlania fragmentów treści w wynikach wyszukiwarki.


SCRUM - metodyka wytwarzania oprogramowania



CODE:ME

UCZYMY PROGRAMOWANIA

Od wielu lat uczymy programować i testować jest to nasza pasja! Pomagamy stawiać pierwsze kroki w świecie IT osobom początkującym, ale również dzięki kursom zaawansowanym, pomagamy wskoczyć na wyższy level wtajemniczenia.

Podnosimy kompetencje w teamach programistów i pokazujemy, między innymi dzięki prężnie działającej społeczności Hackerspace Trójmiasto, że branża IT i nowe technologie w ogóle są naprawdę  ciekawe i niezwykle różnorodne.

Fundacja CODE:ME
al. Wojska Polskiego 41,
Gdańsk
telefon: +48 537 492 774
e-mail: kontakt@codeme.pl
facebook: @FundacjaCODEME 