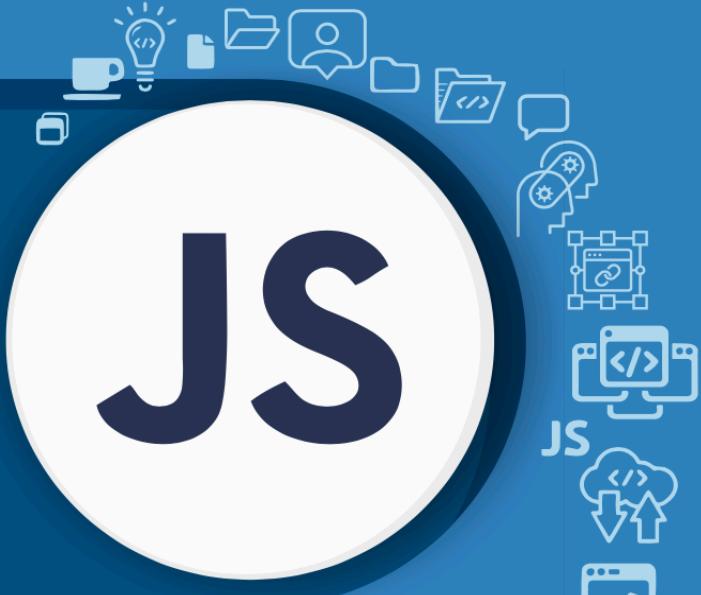


Marcin Moskała

JavaScript

od podstaw zacznij swoją przygodę
z programowaniem

ON



nauczmy się podstaw języka
i napiszemy razem grę



Jak nauczyć się programowania?

Nauka programowania pod wieloma względami przypomina naukę języka obcego. Trzeba poznać słówka (słowa kluczowe, nazwy funkcji), gramatykę (składnia języka) oraz pewne praktyki. Podobnie jak podczas uczenia się języka obcego, potrzeba przede wszystkim pracy własnej, a także wielu godzin poświęconych na przyswajanie materiału, praktykę, praktykę i jeszcze raz...praktykę.

Powstało sporo narzędzi, które mogą nam pomóc na tym polu. Najważniejszym z nich, choć nie jedynym, są kursy internetowe. Można korzystać też z książek, kursów na żywo, gier programistycznych, aplikacji na telefon i wielu innych. Każda z tych opcji ma swoje mocne i słabe strony. Zróbmy więc przegląd najważniejszych dostępnych opcji.

Kursy internetowe

Za najistotniejsze narzędzie do nauki programowania uważam interaktywne kursy internetowe. Te najlepsze zaczynają się od wprowadzenia do teorii, najczęściej w postaci filmu lub tekstu objaśniającego całą potrzebną wiedzę. Po tym często następują pytania sprawdzające, które zmuszają nas do odświeżenia i uporządkowania informacji. Następnie zaczyna się najważniejsze, czyli praktyka.

Dobre ćwiczenie to wyzwanie na odpowiednim poziomie. Zbyt łatwe niewiele nauczy, zbyt trudne jeszcze mniej, a do tego potrafi przyłknąć. Sztuka polega na ułożeniu odpowiednich ćwiczeń. Do tego dochodzi ich liczba — musi być ich wystarczająco, aby wiedza została dobrze zrozumiana i utrwalona, ale jednocześnie nie za dużo, żeby nie znudzić i nie zniechęcić.

Dzięki temu, że kursy łączą w sobie wiele technik przekazywania wiedzy, okazują się bardzo skuteczne. Oczywiście nie wszystkie, ale w internecie znajdziesz dużo dobrej jakości darmowych kursów dotyczących programowania w wielu różnych językach. Czasami możesz też znaleźć kurs warty inwestycji.

```
// Setup
var a;
// Only change code below this line
myVariable = 5;

This assigns the Number value 5 to myVariable.

If there are any calculations to the right of the = operator, those are performed before the value is assigned to the variable on the left of the operator.

var myVar;
myVar = 5;

First, this code creates a variable named myVar. Then, the code assigns 5 to myVar. Now, if myVar appears again in the code, the program will treat it as if it is 5.

Assign the value 7 to variable a.
```

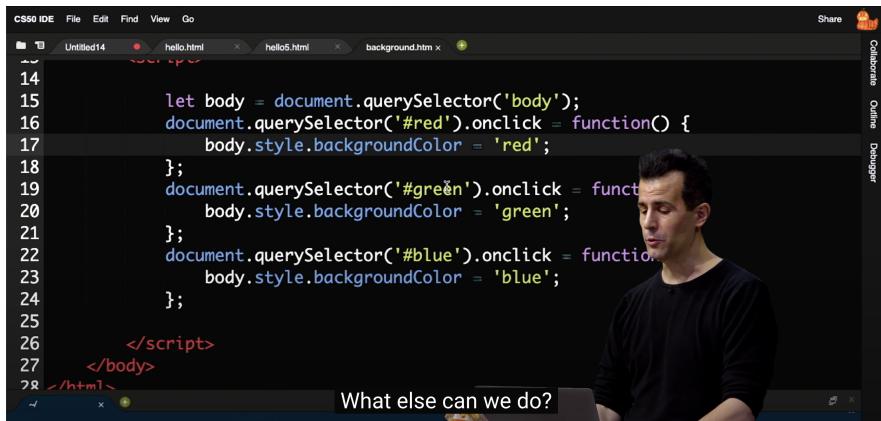
Zrzut ekranu z darmowego kursu na freecodecamp.org. To typowa formuła przy interaktywnych kursach - po lewej stronie widzimy teorię, a po prawej mamy zadanie do wykonania.

Jak zacząć? W internecie roi się od mnóstwa zarówno darmowych, jak i płatnych kursów. Szukaj ich pod hasłem “top free courses js”. Z JavaScript polecić mogę freeCodeCamp, a z płatnych opcji Frontend Masters oraz Pluralsight. Przed kupnem kursu na stronie Udemy polecam poczytać o nim opinie, zamieszczone na innych stronach (są tam zarówno wartościowe kursy, jak i te, w mojej opinii, wątpliwej jakości).

Akademickie kursy internetowe

Innym rodzajem kursów internetowych są te typowo akademickie. Mają one specyfczną formułę — bazują przede wszystkim na wykładach, pomiędzy którymi wykonuje się zadania lub prace domowe. Jest to cięższa metoda nauki, chociaż wiele z nich cechuje fenomenalna jakość.

Jak zacząć? Dobre kursy tego typu można znaleźć na platformach edX, Coursera oraz na YouTube. Szczególnie polecam kursy prowadzone na Uniwersytecie Harwarda przez David J. Malan (dostępny na YouTube lub edX) oraz, dla zainteresowanych sztuczną inteligencją, kursy Andrew Ng na platformie Coursera.



Fragment nagrania z kursu CS50 przeprowadzanego na Uniwersytecie Harwarda przez prof David J. Malan. Udostępniane na YouTube oraz na platformie edX.

Nauka programowania na YouTube

Codziennie na YouTube wgrywanych jest wiele godzin treści wspierających naukę programowania. Jak wszędzie, można trafić na filmy różnej jakości — od takich, których nie da się oglądać, po te wciągające i bardzo przydatne.

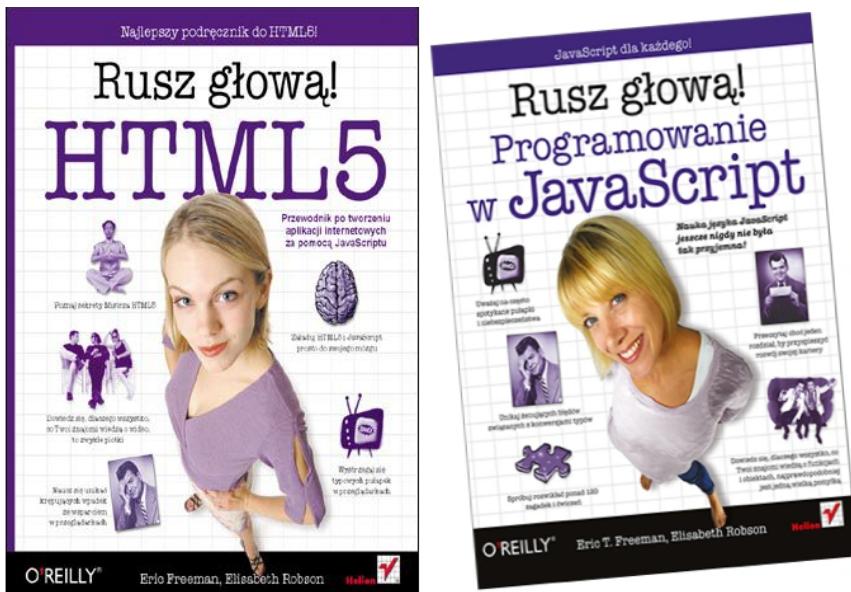
Przewagą kursów na YouTubie jest to, że można tam znaleźć dobryj jakości materiał w języku polskim. Mamy wielu zaangażowanych polskich twórców, którzy poświęcili mnóstwo czasu, aby dostarczyć jakościowe kursy programowania. Na przykład kanały takie jak [Pasia informatyki](#), [helloroman](#), [overment](#), [Jak nauczyć się programowania](#) czy [Samuraj Programowania](#). To jednak dopiero początek listy bohaterów YouTube'a, polecam więc po prostu poszukać, czego chcesz się nauczyć, a jest spora szansa, że znajdziesz wartościowy film na dany temat.

Kolejna mocna strona YouTube'a to ogromna liczba tutoriali pokazujących, jak stworzyć projekt krok po kroku. Takie rozwiązywanie zazwyczaj okazuje się bardzo dobrym wsparciem dla początkujących, ponieważ znacznie łatwiej robić projekt podążając za kimś. Jest to więc świetna ścieżka zarówno dla pasjonatów, jak i dla osób, które chcą wzbogacić swoje CV o nowe projekty.

Jak zacząć? Zaczynaj wyszukiwać interesujące Cię tematy w serwisie YouTube i sprawdzaj uzyskane propozycje. Z czasem znajdziesz twórców, których nagrania najbardziej do Ciebie przemawiają. Na YouTubie, podobnie jak w Google, polecam używać słów kluczowych, czyli "JavaScript kurs gra" zamiast "Jak napisać grę w JavaScript?".

Książki

Niejednokrotnie osoby uczące się programowania skarżyły mi się, że wiele polskich książek o tej tematyce zostało napisanych nadmiernie skomplikowanym językiem, co bardzo utrudnia zrozumienie i sam proces pozyskiwania wiedzy. Na szczęście są też książki łatwe, tłumaczące wszystko krok po kroku, bez gubienia czytelnika po drodze (mam nadzieję, że jestem autorem jednej z nich). Rekomenduję serię dla początkujących zatytułowaną "Rusz Głową". Wyjaśnia ona wszystko jasno i od zupełnych podstaw.



Przewagą książek nad kursami online jest poręczność oraz brak konieczności siedzenia przed monitorem. Możesz korzystać z nich przed snem albo gdy jedziesz metrem. Z drugiej jednak strony są mniej interaktywne i angażujące niż kursy internetowe. Dlatego traktowałbym je raczej jako uzupełnienie niż podstawę nauki.

Podcasty

Jedną z dodatkowych metod nauki mogą być podcasty dotyczące programowania. Nie nauczmy się z nich programować, ale mogą pomóc nam we wzbogaceniu słownictwa oraz pogłębieniu zrozumienia na temat konceptów, trendów i

tego, jak działa branża. Przewagą podcastów jest to, że można ich słuchać jak radia — w czasie prowadzenia samochodu, biegania, czy porządków domowych.

Bootcampy i kursy na żywo

Kolejną popularną opcją zrobiły się tzw. bootcampy, czyli intensywne kursy programistyczne. Największe korzyści z nich płynące to:

- Motywacja — króluje tutaj aspekt psychologiczny posiadania zaplanowanego czasu na naukę oraz działania razem z grupą.
- Uporządkowany program nauki — dużym ułatwieniem jest sytuacja, gdy wiesz, czego warto się uczyć i w jakiej kolejności.
- Prowadzący odpowiadający na pytania uczestników — to pomaga wyjaśnić niezrozumiałe kwestie i naprawić błędne rozumienie na wcześniejszym etapie.

Z drugiej strony, bootcampy nie należą do tanich alternatyw. Narzucają też pewien rygor, a więc zmuszają nas do dostosowania planów i zaangażowania się w określonym czasie. Nie są konieczne — widziałem wiele osób, które nauczyły się programowania bez nich. Jeśli potrafimy sami się zmotywować, to wystarczą nam sporadyczne konsultacje z nauczycielem programowania (najlepiej praktykującym programistą). Znając plusy i minusy, sami musimy zdecydować, czy warto się wybrać na taki intensywny kurs, czy nie.

Mentoring

Rozwój umiejętności programistycznych znacznie przyspiesza posiadanie dobrego mentora. Może to być bliski znajomy programista, trener na kursie, wybrana osoba czy też współpracownik. Dobry mentor powinien mieć doświadczenie, cierpliwość, czas i chęci, by pomóc Ci w nauce. Jego rolą jest ukierunkowanie procesu nauki, udzielenie pomocy, odpowiedzenie na pytania, wspieranie i motywowanie.

Zgadzisz się chyba, że nie brakuje możliwości jak efektywnie uczyć się programowania. Na powyższych opcjach lista się nie kończy. Niejedna osoba głowi się, jak pomóc w tym zadaniu, a w efekcie regularnie powstają nowe metody. Efektywnie można się uczyć nawet [na telefonie](#), czy [grając w gry programistyczne](#).

Gry w służbie programowania

Nauka programowania to przede wszystkim praktyka, czyli nic innego jak godziny poświęcone na czytanie i pisanie kodu. Zarówno jedno, jak i drugie powinno stanowić odpowiednie wyzwanie: zbyt łatwe zadanie niewiele Cię nauczy, zbyt trudne będzie nieproporcjonalnym kosztem do wysiłku, a co więcej, może Cię zniechęcić. Dobre kursy podsuwają więc krok po kroku coraz to bardziej wymagające zadania. Jest jednak sposób by uczyć ten proces ciekawszym: zamiast mieć za zadanie wyświetlanie konkretnych tekstów, możemy pokierować robota działającego na stacji kosmicznej albo poprowadzić armię do walki. Taka właśnie idea przywieca grom programistycznym.



Zrzut ekranu z gry CodeCombat, z jednego z początkowych poziomów. Po prawej stronie piszemy kod sterujący bohaterem, a po lewej możemy uruchomić ten kod i zobaczyć jaki uzyskamy efekt.

Grom programistycznym przywieca cel łączenia dwóch światów: zabawy i nauki. Osobę z odpowiednim podejściem potrafią wciągnąć niczym najlepszą "strzelankę", a przy tym nauczyć niewiele mniej niż dobry kurs.

Dla mnie osobiście programowanie lub praca programisty sama w sobie przypomina grę strategiczną albo RPG. Ma się jasno zdefiniowany problem, poznaje się narzędzia i możliwości, a na koniec widoczny jest sukces. Trochę tak, jakby dostało się misję, następnie krążyło po świecie w poszukiwaniu podpowiedzi jak ją rozwiązać, aż w końcu uzyskiwało pożądany rezultat.

Odpowiednie podejście to sprawa kluczowa, gdyż osoby wysoko zmotywowane takie rzeczy mogą nie bawić, a wręcz przeszkadzać. Wybiorą one kurs

programistyczny bez elementów graficznych i gamifikacyjnych⁴⁷. Słusznie, nie należy się przecież zmuszać do zabawy. Z drugiej strony warto też docenić istnienie takich gier i poszukać czegoś, co mogłoby nas wciągnąć. Sam spędziłem na nich dużo czasu w młodości. Widziałem też wiele osób, które nauczyły się programować dzięki grom. Dostrzegam jakąś magię w tym, że człowiek wciagnięty w zabawę potrafi spędzić na nauce cały dzień, nawet nie zważając na upływający czas.

Warto pamiętać o czerpaniu przyjemności z nauki programowania. Programowanie uważam za coś naprawdę fantastycznego. Nie znam innej branży, w której z taką pasją podchodzi się do tego, co się robi. To pasja, której trzeba się nauczyć — i uwierz mi, warto. Przede wszystkim pomoże Ci w nauce, ale także rekruterzy zwracają na nią uwagę. **Największym wrogiem nauki nie jest to, że zajmie ona trochę więcej czasu, ale to, że stracisz motywację i przestaniesz się uczyć.** Tak, robienie i dostrzeganie postępów postrzegam jako bardzo istotne, ale frajda z samej nauki czyni ją łatwiejszą. Warto więc o nią zadbać. Jakkolwiek może to abstrakcyjnie brzmieć dla osoby chcącej jak najszybciej się przebranżolić — trzeba nauczyć się kochać programowanie. Bez tego ciężko zostać programistą, a jeszcze ciężej stać się dobrym w swoim fachu.

Przykładowe gry programistyczne

Gier programistycznych znamy mnóstwo. Jak wpiszesz w Google “programming games”, to łatwo znajdziesz liczne zestawienia. Żeby jednak dać pewien przedsmak, z czego można wybierać, przedstawię te, w które sam grałem.

Historia zaczyna się od bardzo starych gier: **Colobot** oraz jego kontynuacja **CeeBot**. Są to już dość sędziwe pozycje, więc przedstawiam je raczej jako ciekawostkę i wprowadzenie. Steruje się w nich robotami na odległej planecie. W różnych częściach zajmujemy się kolonizacją oraz obroną planet, lataniem statkiem, a nawet wyścigami czy grą w kosmiczną piłkę nożną. W grze operuje się wymyślonym na jej potrzeby językiem podobnym do C++. Co ciekawe, Colobot uzyskał rekomendację Ministerstwa Edukacji i Nauki jako środek dydaktyczny na poziomie gimnazjum i szkół ponadgimnazjalnych w zakresie nauki tworzenia algorytmów oraz opanowania reguł programowania i składni typowych dla języków C++ i Java. Była to rzeczywiście pierwsza gra programistyczna, w którą grałem. Wiele się przy niej nauczyłem. Dziś już bym jej

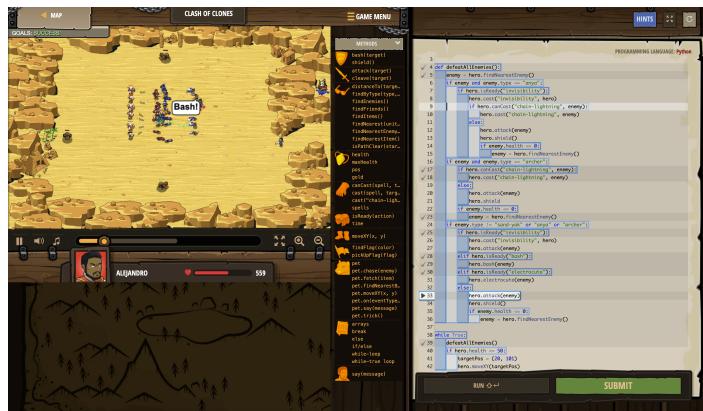
⁴⁷Gamifikacja lub grywalizacja to celowe wykorzystanie mechanizmów stosowanych przy projektowaniu gier w celu aktywizowania, motywowania i utrzymania użytkowników.

nie rekomendował, głównie dlatego, że mocno się zastarzała, wymaga instalacji, a poza tym lepiej przy nauce operować na języku, z którego potem planujemy korzystać.



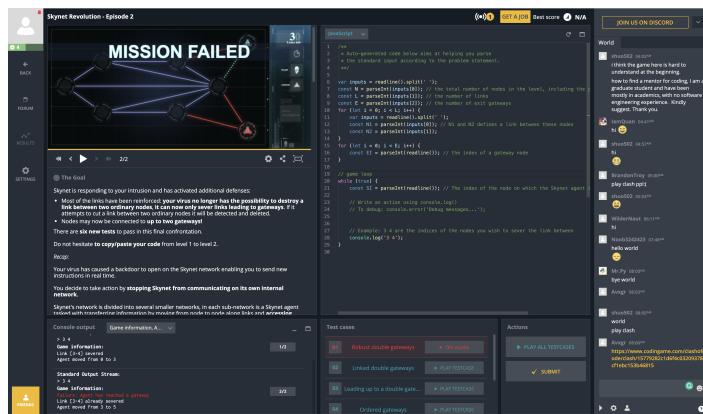
Zrzut ekranu z gry CeeBot.

Bardziej współczesną opcją, a przy tym moim osobistym faworytem, jest **CodeCombat**. Gra opiera się na sterowaniu bohaterem w świecie RPG. Aktualnie możesz do tego wybrać język Python, JavaScript, CoffeeScript lub C++. W kolejnych planszach mierzysz się z różnorodnymi wyzwaniami, od najprostsze- go chodzenia i walka- czenia, po wydawanie rozkazów całym armiom. Grę wspo- minam jako dobrze skonstruowaną i wciągającą. Warto ją traktować jak uzu- pełnienie w procesie nauki. Zakładając, że zaczynasz już z jakimś doświadcze- niem, pierwsze poziomy mogą się wydać proste. Wystarczy przejść je szybko, aby dojść do większych wyzwań. Większym problemem będzie, gdy stopień trudności przekroczy twoje aktualne umiejętności. Nie przejmuj się, po prostu poćwicz na innym kursie i wróć do gry później. Podczas grania polecam robić notatki z dostępnych funkcji.



Zrzut ekranu z gry CodeCombat.

Kolejna popularna propozycja to **CodinGame**. Tutaj już bezpośrednio zajmujemy się rozwiązywaniem wyzwań programistycznych. Mamy do wyboru zarówno ogromną liczbę misji, jak i języków programowania. Do tego liczne elementy współzawodnictwa z innymi graczami, które sprawiają, że potrafi być ona naprawdę wciągająca. Jako ciekawostkę dodam, że zyskała popularność nie tylko wśród niedoświadczonych programistów, ale także wśród osób ćwiczących się w konkurowaniu na zawodach programistycznych (Competitive Programming)⁴⁸.



Zrzut ekranu z gry CodinGame.

Warto wspomnieć o grach, które powstały przede wszystkim, aby dostarczyć rozrywki z samej gry, a nauka programowania lub konceptów programistycz-

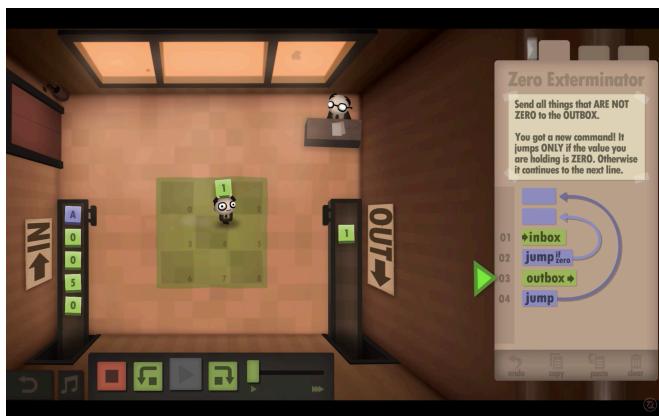
⁴⁸Ażkolwiek są do tego dedykowane platformy jak HackerRank, CodeChef czy CodeForces.

nych jest raczej dodatkiem. Przykładowo Hack ‘n’ Slash, który na początku przypomina zwykłą grę RPG, ale szybko odkrywamy, że mechanika pozwala na hakowanie gry, w którą gramy, a z czasem możemy nawet modyfikować jej kod źródłowy.



Zrzut ekranu z gry Hack ‘n’ Slash.

Powstał też szereg gier, w których w zasadzie nie pisze się kodu jako takiego, ale przy użyciu różnego rodzaju abstrakcji uczy się myślenia i algorytmiki wykorzystywanej w programowaniu. Taką pozycją jest na przykład Human Resource Machine. Więcej takich gier przedstawię w następnym rozdziale, o nauce programowania na telefonie.



Zrzut ekranu z gry Human Resource Machine.

Zakończenie

Nauka programowania to wspaniały proces. Zbyt często widzę, jak ludzie nie potrafią go docenić. Najpierw gnają, ciągle niezadowoleni jak wiele jeszcze nie wiedzą, a jak już osiągną pewien poziom i coraz mniej jest wiedzy do zdobycia, wspominają tamten czas z rozrzenieniem. Wtedy programiści często zaczynają poszukiwać okazji do nauki czegoś nowego — kształcąc się w nowych językach i technologiach, uczęszczają na konferencje i hakafony, spędzają wolny czas na pisaniu open source. Nauka uskrzydla, daje poczucie rozwoju i dostarcza niesamowitej satysfakcji.

Możesz przyrównać ten proces do wyprawy w góry. Tak, każdy chce zdobyć szczyt, ale przecież najprzyjemniejsza jest sama wspinaczka. Kiedy alpinista wejdzie na wymarzoną górę, przeważnie nie siedzi zbyt długo na jej szczytce, a już szuka następnego wyzwania. Chodzi o proces. Wątpię, by ktoś został dobrym alpinistą, przy każdym kroku będąc na siebie złym, że jeszcze nie dotarł do celu albo że zbyt wolno tam podąża. Tak samo w programowaniu – istota opiera się na czerpaniu radości z procesu samej nauki.

Jako jedną z możliwości urozmaicenia tej ekscytującej ścieżki polecam sięgnięcie po gry programistyczne. Pełnią różne funkcje, od uczenia podstaw, stawiania wyzwań programistycznych, aż po przybliżanie algorytmiki. Warto popatrzeć na aktualnie popularne gry programistyczne i przekonać się, czy jakaś Cię nie zainteresuje, bo programowaniem warto się bawić.

Nauka programowania na telefonie

Niemal każdy z nas codziennie nosi w kieszeni potężne urządzenie do nauki, zdolne do wyświetlania wciągających filmów, stawiania nam wyzwań programistycznych, czy nawet odpytywania z przerobionego materiału. Tak, mówię o smartfonie.

Oczywiście sam telefon to za mało by nauczyć się programowania. W końcu wymaga ono środowiska programistycznego, klawiatury itp. Jeśli masz wybór – ucz się na komputerze — to środowisko zdecydowanie bliższe temu, na którym się pracuje. Niemniej smartfon potrafi być świetnym wsparciem. Przede wszystkim dlatego, że prawie zawsze mamy go pod ręką. Pozwala więc na naukę, gdy siedzimy w poczekalni, jedziemy autobusem lub podczas krótkiej przerwy w pracy.

Nie zachęcam tutaj do bycia przyklejonym do telefonu. Jeśli nie masz tego nawyku, to nie rekomenduję go sobie wyrabiać. Jeśli natomiast i tak spędzasz długie godziny na Facebooku, Instagramie, czy na konsumowaniu newsów sugeruję, by dla odmiany wykorzystać część z tego czasu na naukę programowania.

Na telefonie można uczyć się na przynajmniej kilka sposobów:

- możemy wykorzystać jeden z kursów do nauki programowania na telefonie,
- możemy uczyć się przy pomocy kursów na platformach MOOC,
- możemy grać w gry programistyczne na telefonie...

Opcji powstało o wiele więcej, ale omówmy te wspomniane powyżej.

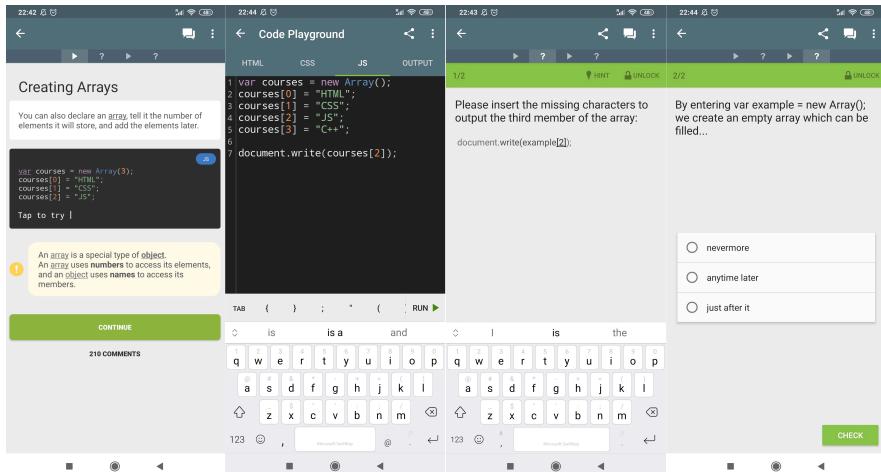
Kursy do nauki programowania na telefonie

Niby nie brakuje aplikacji do nauki programowania, zwłaszcza na platformę Android, ale mniej ich zostanie, gdy zostawimy tylko te dobre i darmowe.

Nowe aplikacje pojawiają się regularnie, a pierwotne wersje albo się starzeją, albo ich twórcy każą sobie za nie niemało zapłacić. Gdy więc rozpoczniesz poszukiwania dla siebie, sprawdź co można znaleźć i jakie aplikacje rekomendują inni użytkownicy. Aby jednak dać Ci pogląd na to, czego można się spodziewać, przedstawiam listę moich faworytów (stan na początek 2021 roku).

SoloLearn

Pierwszą aplikacją, którą mogę polecić, jest SoloLearn. Bawiłem się z nią już kilka dobrych lat temu. To aplikacja solidna, z dobrze skonstruowanymi lekcjami. Każda z nich składa się naprzemiennie z teorii oraz wyzwań. Dostarcza także playground, czyli miejsce, gdzie na telefonie możesz edytować i uruchamiać kod. Nie brakuje w niej wsparcia dla wielu języków, w tym JavaScript, HTML, CSS, Java, Python czy Kotlin. Wizualnie nie należy może do najpiękniejszych, ale wygląd przecież nie jest najważniejszy.

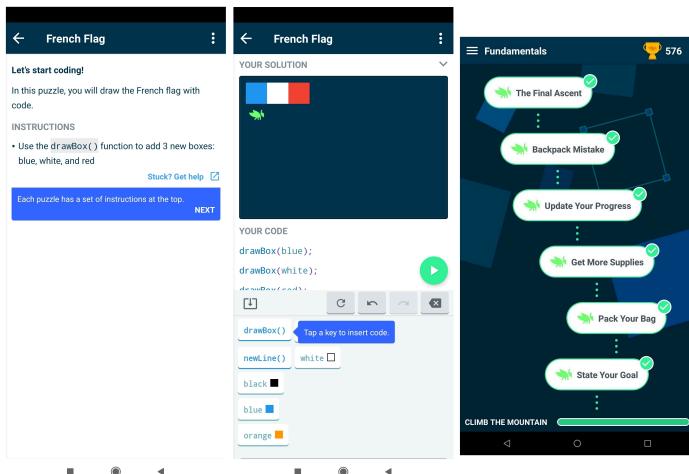


Zrzuty ekranu z aplikacji SoloLearn do nauki programowania na telefonie.

W czasie pisania tej książki nauka z SoloLearn jest darmowa. Można zapłacić za konto Pro, ale wszystkie lekcje i tak są dostępne bez niego.

Grasshopper

Kolejna propozycja to Grasshopper, napisana przez programistów Google. Zgodnie ze stroną, współdzieli ona misję z moją książką, czyli została stworzona, by nauka programowania stała się przystępniejsza i bardziej dostępna dla każdego. Wyróżnia ją bardzo przydatna funkcjonalność, czyli środowisko do programowania, gdzie funkcje i wartości, zamiast pisać, przeciąga się jak bloki. Lekcje zostały ciekawie napisane i można się z niej sporo dowiedzieć.



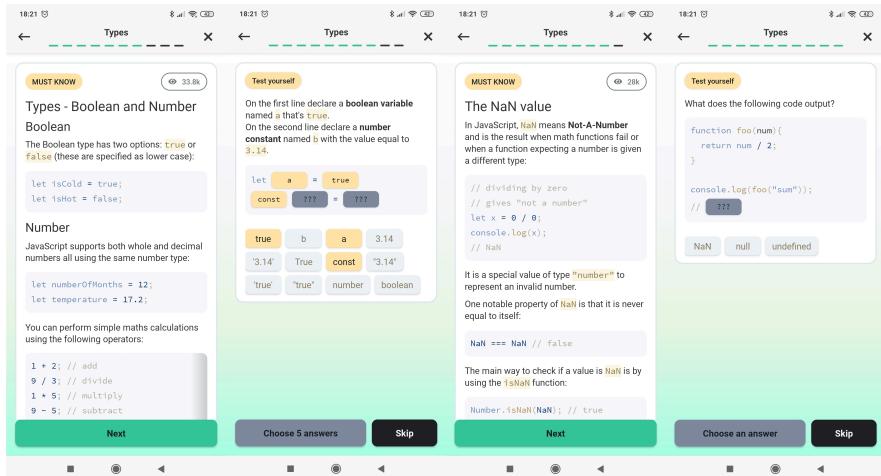
Zrzuty ekranu z aplikacji Grasshopper do nauki programowania na telefonie.

W czasie pisania tej książki aplikacja Grasshopper jest darmowa, podobnie jak wiele innych produktów stworzonych przez Google.

Enki

Kolejną aplikacją, którą mógłbym zarekomendować, jest Enki. Ona również uczy programowania poprzez lekcje i wyzwania. Zarówno jedne, jak i drugie są dobrze przemyślane i zaprojektowane. Spodobały mi się w niej wyzwania typu “zupełnij lukę”, gdyż zmuszają nas do myślenia, a jednocześnie łatwo je rozwiązywać na telefonie.

W czasie pisania tej książki większość lekcji w Enki mamy dostępnych za darmo i tylko niektóre z nich wymagają płatnego konta, więc wykupienie go jest opcjonalne.



Screen z aplikacji Enki do nauki programowania na telefonie.

Płatne aplikacje

Stworzenie dobrej aplikacji niestety kosztuje i niewiele z nich jest dostępnych całkiem za darmo. Jeśli jednak ten typ nauki wydaje Ci się interesujący, możesz również znaleźć wartościowe płatne aplikacje w racjonalnych cenach.

Platformy do kursów dostępne na telefonie

W rozdziale *Jak nauczyć się programowania?* wspominaliśmy o platformach z kursami o charakterze raczej akademickim. Najpopularniejsze mają swoje aplikacje mobilne, pozwalające na naukę na telefonie. Umożliwiają one często na naukę w trybie offline czy odtwarzanie kursu na słuchawkach przy wyłączonym ekranie. Jest to spora wygoda, na przykład gdy jesteśmy w podróży (wiele kursów ukończyłem w pociągach i samolotach). Ponownie poleczę platformy edX i Coursera, gdzie znajdziemy wiele darmowych kursów prowadzonych często przez światowej sławy autorytetę⁴⁹.

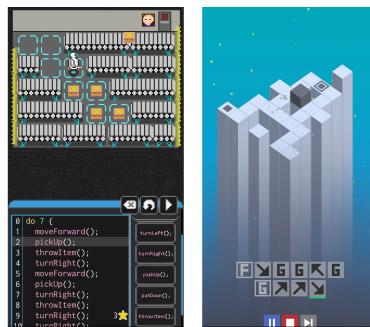
⁴⁹Dla przykładu, na Coursera kurs z języka Scala prowadzony jest przez samego twórcę, Martina Oderskyego. Podobnie kurs z języka Kotlin współtworzony przez Andrey Breslava. Znaleźć tam można także kursy wydane przez czołowe uczelnie techniczne z całego świata.

The screenshot shows a Coursera course page for 'HTML, CSS, and Javascript for Web Developers' offered by Johns Hopkins University. The course has a rating of 4.8 (10k) and is described as being offered online. It includes sections for 'About this Course', '100% online', 'Flexible deadlines', 'Approx. 29 hours to complete', and 'angielki'. A blue button at the bottom says 'Enroll For Free (Starts mar 30)'.

Zrzuty ekranu z aplikacji Coursera służące do nauki na telefonie. Większość wykładów ma format akademicki i jest prowadzona przez profesorów z dużych uczelni całego świata.

Gry programistyczne na telefonie

Tak jak mówiliśmy o grach programistycznych na komputer, tak też można znaleźć gry na telefon, które ćwiczą myślenie programistyczne. Warto potraktować je jako dodatek do zgłębiania programowania, tak jak łamigłówki przy nauce matematyki. Poniżej screeny z gier [Automaton](#) oraz [Cursivity](#).



Zrzuty ekranu z gier Automaton oraz Cursivity ćwiczących myślenie programistyczne.

Ilęż to powstało narzędzi do nauki programowania... od kursów, książek, gier, aż po aplikacje na telefon. To niesamowite jak wiele osób poświęciło się temu, by uczyć i pomagać innym w nauce. Nic, tylko korzystać.

Jak zostać programistą?

Jednym z najczęściej zadawanych mi pytań przez osoby z dalszego otoczenia to: Jak zostać programistką⁵⁰? Mam wobec niego mieszane odczucia. Doceniam osoby, które chcą się uczyć i wziąć los w swoje ręce. Moi znajomi, którzy zdecydowali się na przejście tej ścieżki, są bardzo zadowoleni ze swojej decyzji. Z drugiej strony widziałem, jak ciężka to była dla nich droga: ile czasu i emocji ich kosztowała. Jest to zwykle bardzo trudny czas — zarówno psychicznie, jak i pod kątem liczby rzeczy, z których trzeba zrezygnować, żeby poświęcić czas na naukę. Dlatego polecam rozłożyć go na raty — najpierw potraktować programowanie jako hobby, uczyć się i kodować dla zabawy. W ten sposób nawet nie zauważamy, jak potencjalna ścieżka do zmiany branży stanie się coraz bardziej osiągalna. Często jednak taka sugestia nie spotykała się z aprobatą wśród moich rozmówców. Słyszałem, że to niemożliwe, że chcieliby zmienić branżę jak najszybciejsz... Dlatego opracowałem plan jak przebranżować się i zostać programistą w możliwie najkrótszym czasie.

Geneza planu

Pierwszy plan nauki programowania powstał, gdy pierwsza osoba zapytała mnie jak możliwie szybko rozpocząć karierę programistyczną. Gdy pomagałem tej i kolejnym osobom, mój plan ulegał zmianom. Większość osób modyfikowała go wedle własnych potrzeb, ale ogólnie podążała za nim. Zdawał się im służyć. Jeśli więc nie masz profesjonalnego trenera programowania, który by poprowadził Cię, a chcesz zostać programistą jak najszybciej, to polecam Ci również zaplanować swoją naukę w oparciu o niego.

Dla kogo jest ten plan?

Ten plan powstał dla osób, które:

- znają język angielski na poziomie przynajmniej komunikatywnym,
- potrafią sprawnie operować komputerem, przeglądarką internetową oraz wyszukiwać informacje w internecie.

⁵⁰Nie udało mi się tutaj uniknąć rozróżnienia płci, więc naprzemiennie będę używać określenia “programista” i “programistka”.

Osoby, które nie spełniają tych warunków, mają niestety dłuższą drogę do tego zawodu. Im polecam rozpocząć od powyższych, wymaganych umiejętności.



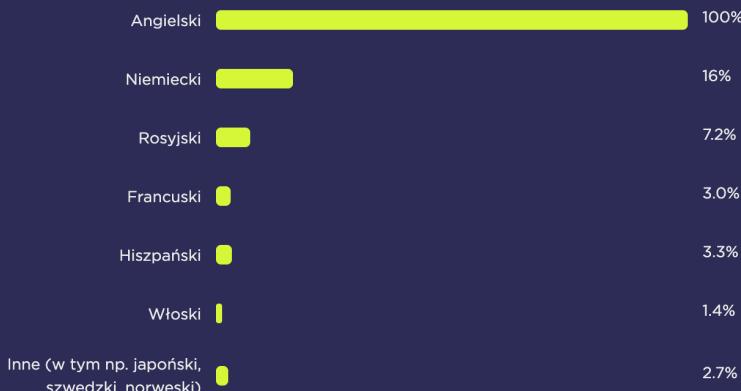
Język angielski jest najpopularniejszym językiem obcym w IT

Językiem, który opanowali prawie wszyscy specjalisi IT, jest język angielski (99,5%). Jest to podstawa, aby funkcjonować w branży, o czym wiedzą praktycznie wszyscy specjalisi. Innymi językami, którymi posługują się specjalisi IT, są m.in.: j. niemiecki (16%), j. rosyjski (7,2%), j. francuski (3%).

Blisko 27% specjalistów IT zna co najmniej dwa języki obce.

■ ZNAJOMOŚĆ JĘZKÓW OBCYCH

Możliwość wielokrotnego wyboru



Zgodnie z raportem Bulldogjob, niemal wszyscy wszyscy specjalisi IT w Polsce znają język angielski.

Plan ten jest także dla osób, które rzeczywiście chcą pracować jako programiści. Programowanie wiąże się z dużą ilością czasu spędzonego przy komputerze i ciągłego rozwiązywania problemów. Taki tryb pracy nie będzie odpowiadał każdemu.

Plan zakłada też dostęp do dużej liczby firm poszukujących specjalistów w tej dziedzinie. Tak sytuacja przedstawia się w dużych miastach. Niestety ciężko mi powiedzieć, jak ta ścieżka powinna wyglądać w mniejszych miejscowościach.

Ile zajmuje przejście tej ścieżki?

To kwestia bardzo indywidualna. Poświęcając **przynajmniej** 20 godzin tygodniowo na intensywną naukę (średnio zakładałbym bliżej 30 godzin), większość znanych mi osób była w stanie przekwalifikować się w 9 do 12 miesięcy, zaczynając od kariery zupełnie nieprogramistycznej (od inżynierów instalacji, aż po sprzedawców w sklepach) do znalezienia pierwszej pracy jako “Junior Developer” lub osoba na stażu. Sumarycznie można by oszacować ten wysiłek na 1000 do 1500 godzin intensywnej nauki. Wynika to przede wszystkim z tego, że do pracy potrzebujemy nabyc umiejętności o wiele szerszych niż tylko samo programowanie w określonym języku. Dla przykładu, jeśli ktoś chce zostać front-end developerem to na pewno powinien znać:

- HTML i CSS,
- JavaScript,
- wybrany framework (jak React, Vue, Angular),
- najważniejsze narzędzia (takie jak Git czy używane IDE),
- przynajmniej podstawowe wzorce i praktyki,
- mieć ogólną wiedzę techniczną,
- poruszać się w tym środowisku naprawdę dobrze (mieć ukończonych przynajmniej kilka projektów).

To dużo więcej niż jedynie język JavaScript, więc tym samym musi zająć więcej czasu.

Plan w 7 krokach

1. **Podejmij wstępную decyzję, co chcesz robić jako programista.** Skonsultuj się z kilkoma doświadczonymi programistami, poczytaj artykuły i pooglądaj porównania na temat różnic między programistą: front-end, back-end, Android, iOS, systemów wbudowanych, gier, big data itp. Zwróć uwagę na łatwość znalezienia pierwszej pracy i wysokość zarobków. Dla przykładu: tworzenie gier brzmi pasjonująco, ale raczej nie jest to łatwa droga. Na ten moment potrzebujesz podjąć decyzję, w jakim kierunku chcesz iść i jakiego języka będziesz używać. Robienie stron internetowych i JavaScript, albo tester automatyczny i Java, albo Data Scientist i Python, albo analityk i SQL, albo Android i Kotlin, albo iOS i Swift, albo programista systemów wbudowanych (ang. embedded systems) i C, albo... cóż, masz ogrom możliwości.

2. **Naucz się języka programowania.** Ten krok zajmie wiele godzin, bo to nie tylko nauka, ale przede wszystkim oswajanie się z czytaniem i pisaniem kodu. Ukończ kilka kursów online od podstaw oraz spędź dość czasu na ćwiczeniach, by czuć się w tym języku komfortowo. Przydatne mogą się okazać gry programistyczne oraz aplikacje do nauki na telefonie.
3. **Wybierz framework.** Nadszedł czas by wybrać z jakiego framework'u⁵¹ będziesz korzystać. Na przykład do pisania stron internetowych popularne są aktualnie React, Vue i Angular. Poczytaj o różnicach i zasięgnij rady doświadczonych programistów.
4. **Naucz się frameworku.** Zrób przynajmniej kilka kursów dotyczących programowania, wykonuj wszystkie ćwiczenia, testuj to, co robi prowadzący, eksperymentuj sam z tym, co już wiesz.
5. **Rób projekty w wybranym frameworku.** Zaczni od kursów na YouTubie (albo innych platformach) jak zrobić różnego rodzaju projekty w wybranym przez Ciebie frameworku. Możesz wyszukać na przykład "Make game in React" czy "Make shop in Vue". Takich kursów nie brakuje na YouTubie. Podążaj za tym, co robią prowadzący i wykonuj ten projekt samodzielnie. To ważne, bo jednocześnie zdobywasz wiedzę jak takie projekty powstają i budujesz swoje portfolio. W przyszłości postaraj się je rozwijać — jak nauczysz się nowych rzeczy, to zastanów się, gdzie możesz ich użyć w napisanych już projektach. Pamiętaj o robieniu kopii zapasowych, najlepiej przy użyciu Git.
6. W międzyczasie **douczaj się reszty narzędzi oczekiwanych przez pracodawców.** Zróbj kurs z Git, z używania konsoli, ze środowiska, w którym będziesz programować, z popularnych bibliotek. Nie zaszkodzi doowiedzieć się, czym jest regex, HTTP i REST, programowanie funkcyjne i obiektowe, najważniejsze skróty klawiszowe itp. Wydaje się, że to niewielkie tematy, ale te umiejętności okażą się cenne zarówno przy programowaniu, jak i przy rekrutacji.
7. **Zaczni jąść CV.** W zasadzie im więcej tym lepiej. Warto wysyłać ich przynajmniej po 100 tygodniowo (nie pomyliłem się, w dużych miastach jest tyle ofert pracy dla programistów). Nie bój się aplikować do tej samej firmy, gdy otworzy się nowa rekrutacja. Nie wahaj się startować na pozycje nieco powyżej Twojego doświadczenia (jak firma nie znajdzie doświadczonego programisty, często zaczyna szukać wśród obiecujących nowicjuszy). Dobrą strategią wydaje się skupienie na tym przez jeden dzień w tygodniu, a przez resztę tygodnia koncentracja na dopracowywaniu swoich projektów oraz przygotowanie się do rozmów o pracę. Do tego procesu pomagają zestawienia typowych pytań padających podczas rozmów rekrutacyjnych. Postaraj się na jak najwięcej z nich mieć z góry przygotowaną odpowiedź.

⁵¹Najważniejsza biblioteka używana w projekcie, decydująca jak piszemy nasz kod. Patrz Słowniczek na końcu książki.

Na wszystkich etapach rób dużo notatek. Przydatne bywa też uczenie się pewnych rzeczy na pamięć przy użyciu fiszek (na przykład w programie Anki). Co prawda wiedza wykuta na pamięć mało przydaje się w samej pracy, ale pozwala zabłysnąć przed rekruterami.

Podczas całej ścieżki warto mieć przy sobie osobę bardziej doświadczoną, z którą możesz się przynajmniej raz w tygodniu skonsultować. Chodzi o kogoś, kto odpowiedziałby na Twoje pytania i wyłapał błędy, które mogłyby Cię blokować w dalszym rozwoju. Najlepiej by był to praktykujący programista.

Polecam nie bać się pójścia na staż lub praktyki, nawet darmowe. Zwykle firmy w młodych programistów i tak muszą inwestować sporo czasu bardziej doświadczonych osób. Dopiero z czasem zaczynają oni przynosić większy zysk, niż sami są wydatkiem. Zazwyczaj, jak tylko zaczniemy pierwszą pracę w programowaniu, tempo nauki niesamowicie przyspiesza, a nasza pozycja na rynku szybko rośnie. Doświadczenie zawodowe jest też bardzo cenne w CV. Tak więc nawet na darmowy staż czy praktykę można patrzeć jak na inwestycję.

Zakończenie

Zmiana branży to ścieżka ciężka, ale możliwa do osiągnięcia. Nie wymaga studiów, wymaga dużo mniej naturalnych zdolności niż się powszechnie wydaje⁵². Wymaga jednak wielu godzin. Gdy spędzamy je bez pośpiechu, w ramach hobby, to upływały szybko. Gdy jednak podchodzimy do nich jak do sprintu, by jak najszybciej dotrzeć do upragnionego nowego zawodu, może nas wykończyć emocjonalnie. Zwłaszcza ostatni krok, w którym siłą rzeczy dostaje się odmowy od wielu firm. Dlatego nikogo na tą drogę nie popycham na siłę. Zachęcam raczej, by zacząć od programowania z pasji. Jeśli jednak się na nią zdecydujesz albo życie Cię na nią skieruje, to mam nadzieję, że mój plan okaże się przydatny i skuteczny.

[1] Vicki L. Sauter, Predicting computer programming skill, Computers & Education, Volume 10, Issue 2, 1986, Pages 299-302, ISSN 0360-1315, <https://bit.ly/2UNQi9V>

[2] Shute, V. J. (1991). Who is Likely to Acquire Programming Skills? Journal of Educational Computing Research, 7(1), 1–24. <https://bit.ly/2U6l1Py>

⁵²Wiele osób rezygnuje, myśląc, że potrzebne są zdolności matematyczne, a tymczasem niektóre badania sugerują, że ważniejsze mogą być zdolności językowe[1][2].