

Podstawy Języka Javascript!

Ten ebook to miejsce, w którym zebrałem wartościową wiedzę z podstaw JS i trochę wskazówek jak uczyć się programowania na początku swojej drogi

Przemocny

Rozdział 0 - Podejście do nauki programowania!

0.1 Jak uczyć się programowania?

Oto jest pytanie! Czy powinniśmy katować się nauką codziennie po naście godzin? W żadnym wypadku, jeżeli poświęcisz co najmniej 3h dziennie na sprawną naukę bez żadnego rozproszenia to możesz uznać dzień za udany! W programowaniu nie chodzi o katowanie się nauką, bo uczyć trzeba się mądrze, a nie na ilość czy kuć na pamięć.

- <https://gladiators-of-javascript.com/blog/spinaj-poslady-i-do-kodowania/>
<https://levelup.gitconnected.com/here-are-20-pieces-of-programming-advice-you-never-asked-for-302b44c1d31c>

0.2 Zawitaj do Polskiej społeczności JS

Polska społeczność Javascriptu to mega pomocni ludzie, zawsze otwarci i skorzy do pomocy oraz chętni do współdzielenia się wiedzą. Często starsi stażem koledzy wrzucają ciekawe tematy dookoła Js-a zarówno po stronie frontendu jak i backendu.

- Grupy związane z Javascript
- <https://www.facebook.com/groups/742940452405327/>
- <https://www.facebook.com/groups/js.pierwsze.kroki/>
- <https://www.facebook.com/groups/jspolska/>
- Fanpage warte śledzenia
- <https://www.facebook.com/academylocalhost/>
- <https://www.facebook.com/javascript4u/>
- <https://www.facebook.com/nodejstutorial4u/>
- <https://www.facebook.com/moriohdotcom/>

0.3 Ile realnie w 2021 będziesz się uczyć na juniora i o jakie pieniądze możesz powalczyć?

Branża IT jest branżą dynamiczną, tutaj nigdy nie dopadnie nas stagnacja czy nuda. Języki programistyczne się rozwijają, dochodzą nowe udogodnienia przez co odstawiamy stare rozwiązania, a za tym idzie ciągła nauka, która dla niektórych może być przytłaczająca, a dla innych ekscytująca. A co Ty jako przyszły junior będziesz musiał umieć w 2021 roku?

- <https://github.com/kamranahmedse/developer-roadmap>
- https://dev.to/ender_minyard/full-stack-developer-s-roadmap-2k12
- <https://www.goskills.com/Development/Resources/How-long-does-it-take-to-learn-how-to-code>
- <https://www.youtube.com/watch?v=SATXBG4WOwo>

Rozdział 1 - Solidne wprowadzenie!

1.0A Konfiguracja środowiska - instalacja node.js, popularne wtyczki do VScode oraz jak poruszać się po terminalu!

VS code jest niezwykle popularny w świecie programistów, a co najlepsze jest całkowicie darmowy (tutaj firma microsoft nas rozpieściła). Do zautomatyzowania pracy pomagają nam wtyczki takie jak: Prettier, Bracket Pair Colorizer, Path Intellisense czy Live Server. Nie zapomnijmy o dodaniu do naszego arsenału środowiska uruchomieniowego czyli node.js.

- <https://www.youtube.com/watch?v=VqCgcpAypFQ>
- <https://www.youtube.com/watch?v=om4HQYXgZog>

1.0B Terminal i jego zastosowanie w programowaniu!

Może teraz Ci po głowie chodzić myśl "Po co mi wiedza o terminalu?". Już spieszę z wyjaśnieniem, w pracy będziemy wykorzystywać system kontroli wersji, który jest obsługiwany przez terminal. Możemy tworzyć, usuwać, przełączać się między folderami w terminalu co usprawnia pracę. Umiejętność obsługi terminalu również pomoże Ci łatwiej zarządzać bazami danych takimi jak np PostgreSQL.

- <https://www.hongkiat.com/blog/developers-command-line/>
- https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Understanding_client-side_tools/Command_line

<https://www.idesignpixels.com/posts/10-things-a-front-end-developer-should-know-in-terminal>

1.1 Czym tak właściwie jest Javascript?

Dlaczego Javascript jest tak popularny w Web Development-cie? (Od kilku lat utrzymuje się w TOP3) oraz jakie duże i popularne aplikacje zostały zbudowane w tym języku.

- <https://javascript.info/intro>
- <https://tweakyourbiz.com/technology/mobile-technology/built-using-javascript>
- <https://medium.com/@hackreactor/what-is-javascript-used-for-d438f3e9ca39>

1.2 Niezbędne źródło wiedzy czyli dokumentacja!

W dokumentacji programistycznej są opisane i wyjaśnione aspekty wybranego języka oraz jego składnia. Jest to również miejsce pierwszej pomocy gdy trafiamy na mur podczas rozwiązywania zadań z danego języka.

- <https://devdocs.io/javascript/>
- <https://developer.mozilla.org/pl/docs/Web/JavaScript>

1.3 GIT oraz Github, a co to takiego?

Github czyli jedna z największych społeczności programistycznych, gdzie możemy podzielić się kodem lub przeczytać świetne artykuły napisane przez ludzi z branży IT. GIT czyli system kontroli wersji, który śledzi nasze poczynania w kodzie i zapisuje poszczególne wersje, dzięki temu możemy śledzić nasze poczynania względem poprzednich wersji.

Dodatkowo świetnie się sprawuje przy pracy w zespole, ale o tym dowiesz się empirycznie!

- <https://www.flynerd.pl/2018/02/github-dla-zielonych-pierwsze-repozytorium.html>
- <https://www.git-scm.com/docs>
- <https://www.youtube.com/watch?v=RG0j5yH7evk>
- <https://www.youtube.com/playlist?list=PLjHmWifVUNMKIGHmaGPVqSD-L6i1Zw-MH>

Rozdział 2 - Podstawy składni!

2.0 Słowa wstępu przed wyciskiem

Przed wkroczeniem do walki ze składnią Javascriptu chciałbym abyś pracował również z kursem wideo który pozostawię w linku poniżej. Wizualizacja szczególnie pomoże wzrokowcom, ale nie tylko im, czasem każdy z nas potrzebuje zobaczyć przykład zrobiony przez doświadczonego programistę by na jego podstawie rozwiązać swój problem. Pamiętaj, że kurs to uzupełnienie, a nie podstawa nauki, Twoją podstawą jest zrozumienie języka poprzez empiryczne doświadczenie, a nie poprzez oglądania masy kursów! Witam w świecie programistów i życzę Ci powodzenia!

- <https://www.youtube.com/watch?v=dOnAC2Rr-6A>

2.1 Zmienne oraz typy danych

Zmienne czyli nasz chleb powszedni. Dzięki zmiennym tworzymy elementy potrzebny do stworzenia logiki aplikacji, każda zmienna ma swój typ o czym dowiesz się w pracy z językiem, a javascript lubi zaskakiwać, a dlaczego?. Dlatego, że javascript to język "Dynamicznie Typowany", a to znacznie utrudnia kontrolę.

- <https://kursjs.pl/kurs/super-podstawy/zmienne.php>
- <https://kursjs.pl/kurs/super-podstawy/typy-danych.php>
- <https://javascript.info/variables>
- <https://javascript.info/types>
- <https://javascript.info/type-conversions>

2.2 Operacje na liczbach oraz zapoznanie z obiektem Math.

Podstawowe operacje matematyczne na liczbach są powszechnie używane do budowania np - Kalkulatorów matematycznych, kalkulatorów BMI lub do obliczania pól figur geometrycznych. Do wielu z tych operacji, użyjemy obiektu Math. Jest niezwykle pomocny przy rozwiązywaniu działań matematycznych.

- <https://kursjs.pl/kurs/super-podstawy/number-math.php>
- <https://javascript.info/operators>
- <https://javascript.info/comparison>

2.3 Operacje na tekstach czyli zmienne typu "string"

Ciągi znaków czyli "stringi". Nauczysz się jak manipulować wybranymi przez Ciebie tekstami. Ucinanie tekstów, dynamiczna zmiana tekstu, wyciągnięcie poszczególnych zdań i wiele wiele innych.

- <https://kursjs.pl/kurs/super-podstawy/string.php>
- <https://javascript.info/string>
- https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Useful_string_methods

2.4 Tablice oraz metody tablicowe.

Tablice to bardzo ważny koncept w Js. W tablicach możemy przechowywać wszelkie zmienne, a nawet inne tablice z innymi zmiennymi, mamy pełną swobodę nad tym co się w niej znajduje. Do manipulowania danymi w środku tablicy służą nam metody! Możemy sortować dane, usuwać, dodawać, filtrować według swoich potrzeb lub wyszukiwać tylko ważnego dla nas elementu

- <https://kursjs.pl/kurs/super-podstawy/tablice.php>
- <https://javascript.info/array>
- <https://javascript.info/array-methods>
- <https://prntscr.com/vbeic1>

2.5 Funkcje.

Funkcje czyli mini programy napisane według naszych potrzeb dla naszych potrzeb lub projektu. Są reużywalne co oznacza, że możemy użyć jednej funkcji w wielu przypadkach o ile ma to sens, jest to istotny temat gdyż bez funkcji nie napiszemy żadnego programu czy aplikacji.

- <https://kursjs.pl/kurs/super-podstawy/funkcje.php>
- <https://javascript.info/function-basics>
- <https://javascript.info/function-expressions>
- <https://javascript.info/arrow-functions-basics>

2.6 Instrukcje warunkowe oraz operatory logiczne AND i OR

Instrukcje służą nam jako przełącznik, a dlaczego jako przełącznik? Bo możemy określić przypadki w których dany kawałek kodu ma się wykonać, a kiedy ma się wykonać inny, jak w systemie binarny (czyli zero/jedynkowy).

- <https://kursjs.pl/kurs/super-podstawy/instrukcje-warunkowe.php>
- <https://javascript.info/ifelse>
- <https://javascript.info/switch>
- <https://javascript.info/logical-operators>

2.7 Pętle - Istotny aspekt w każdym języku programowania.

Pętlę głównie używamy do iterowania po tablicach, a co to oznacza iterowanie? Jest to możliwość zrobienia dowolnej akcji na elementach tablicy z zachowaniem matematycznej kolejności. Czyli od pierwszego elementu kolejno do ostatniego. Możemy iterować również po obiektach, ale o tym dowiesz się w dalszej części kursu, więc miej to już na uwadze!

- <https://kursjs.pl/kurs/super-podstawy/petle.php>
- <https://javascript.info/while-for>
- <https://blog.bitsrc.io/understanding-the-for-of-loop-in-javascript-8aded97d7ef8>
- <https://www.youtube.com/watch?v=x7Xzvm0iLCI>

2.8 Obiekty - Javascript mówi "W Js wszystko jest obiektem".

Obiekty to wymagający temat, mamy tutaj słowo kluczowe THIS, konstruktory, dziedziczenie, klasy i kilka innych tematów, które na pewno nie będą łatwe. Lecz zrozumienie działania obiektów jest, niezbędne do opanowania tego języka. Jak wspomniałem wcześniej w Js wszystko jest obiektem.

- <https://kursjs.pl/kurs/obiekty/obiekty.php>
- <https://javascript.info/object-basics>
- <https://www.youtube.com/playlist?list=PLiqmKUYV2ZSorXNZVlkb7Acl7M80hhpdg>

2.9 Coś pomiędzy tablicą, a obiektem - czyli Mapy i Sety.

Map. Jest to prosty obiekt w którym możemy przechowywać tak zwane pary key/value, które już pewnie poznałeś przy okazji nauki o obiektach. Ale istnieje pewna różnica między Map, a Obiektem. Różnica jest taka, że klucz w map może przyjąć dowolny typ, a w obiektach klucz zawsze jest konwertowany na "string". Dodatkowo po mapach możemy iterować takimi metodami jak entries, keys, values

Set. Set niezastąpionym typem kolekcji dlatego, że przechowuje tylko unikalne wartości(set nie przechowuje kluczy).

- <https://javascript.info/map-set#set>

- <https://kursjs.pl/kurs/obiekty/map-set.php>
- <https://medium.com/front-end-weekly/es6-map-vs-object-what-and-when-b80621932373>
- <https://medium.com/free-code-camp/lets-learn-about-set-and-its-unique-functionalities-in-javascript-5654c5c03de2>

2.10 Event Loop oraz jak działa asynchroniczność względem synchroniczności w JS

Asynchroniczność pozwala nam na wykonywanie kilku rzeczy jednocześnie. Wyobraź sobie, że Twoja przeglądarka w momencie samego startu wykonała dla Ciebie zadanie, a Ty w tym samym czasie jeszcze otwierasz kilka innych zakładek. Bez asynchronicznego działania nie udałooby się to dlaczego? O tym dowiesz się sam z przygotowanych materiałów(a ten o asynchroniczności jest szczególnie genialny!).

<https://amangupta6121998.medium.com/asynchronous-concept-in-javascript-10dce818281d>

<https://medium.com/front-end-weekly/javascript-event-loop-explained-4cd26af121d4>

- <https://www.youtube.com/watch?v=Ysp1rh5d5ps>

2.11 Promise! Czyli obietnice w Js

Obietnice w Javascript możemy przyrównać do obietnic z prawdziwego życia. Przykładowo - "Jeśli dobrze zrozumieć promisy w js, to rozwiąże Twoje zadanie!". Czyli najpierw składamy obietnice(że coś zrobimy), a potem mamy reakcję, czyli wynik spełnienia naszej obietnicy. Promisy działają na tej samej zasadzie, najpierw deklarujemy obietnicy, a potem dostajemy jej wynik.

<https://www.youtube.com/playlist?list=PLjHmWifVUNMLOfSdeSSEGgSXebPXL7Z3Q>

- <https://kursjs.pl/kurs/ajax/promise.php>
- <https://javascript.info/promise-basics>
- <https://javascript.info/promise-api>

Rozdział 3 Ciekawostki oraz tematy uzupełniające!

3.0 Motywacja to również podstawa!

Pamiętaj, każde Twoje działanie ma swoje konsekwencje w przyszłości. Dlatego motywacja i wiara w siebie jest bardzo ważna! Na początku zawsze jest ciężko, ale

mądrym podejściem do tematu nauki czy sportu jesteśmy w stanie osiągnąć każdy wymarzony cel! Dlatego nie szukaj drogi na skróty tylko pokaż sam sobie że potrafisz

<https://gladiators-of-javascript.com/blog/framework-mentalny-czyli-7-wskazowek-mentora-ulatwiajacych-nauke-programowania/>

<https://www.codingdojo.com/blog/7-tips-learn-programming-faster>

3.1 Błędy podczas nauki programowania(unikaj ich!)

Błędy to rzecz całkowicie ludzka w końcu każde je popełnia, ale najlepiej uczyć się na czyichś błędach, a nie na swoich. Do nauki trzeba podejść z szacunkiem, nie rzucać się na najgłębszą wodę, a konsekwentnie zawyżać poprzeczkę. Jak zwykle z pomocą przychodzą do nas starsi koledzy z branży, którzy podzielili się swoimi błędami podczas nauki, żeby pomóc Tobie nowy wojownikowi ich nie popełniać!

<https://www.dummies.com/programming/cpp/the-10-most-common-beginner-programming-mistakes/>

- <https://blog.nerdjfb.com/17-beginner-programming-mistakes/>

3.2 Programowanie funkcyjne

W javascript alternatywą dla programowania zorientowanego obiektowo(OOP) jest tytułowe programowanie funkcyjne.

- <https://bmilewski.pl/javascript/javascript-programowanie-funkcyjne/>

- <http://www.algosmart.pl/programowanie-funkcyjne-powtorka-reactjs-11/>

- <https://www.youtube.com/playlist?list=PLhXZp00uXBk4ejkUZiDnv3R0AERA7Z4xp>

3.3 Programowanie zorientowane obiektowo czyli OOP

Programowanie obiektowe służy nam do modularyzowania kodu, a co oznacza modularyzowanie? Jest to nic innego jak rozbijanie naszego kodu na mniejsze uniwersalne "klocki", które możemy używać w wielu miejscach. Z takich elementów, możemy zbudować co tylko zechcemy, tutaj niebo nie jest limitem!

<https://www.youtube.com/playlist?list=PL4cUxeGkcC9i5yvDkJgt60vNVWffpbIB7>

<https://medium.com/free-code-camp/an-introduction-to-object-oriented-programming-in-javascript-8900124e316a>

- <https://javascript.info/classes>

- <https://kursjs.pl/kurs/obiekty/obiekty.php>

Dodatkowo zapoznaj się z wzorcami projektowymi!

- <https://refactoring.guru/>

3.4 Co ten Javascript?! Czyli dziwne aspekty Js-a

Playlista skupiająca uwagę wszystkim dziwnym sprawą związanym z programowaniem w javascriptcie.

- <https://www.youtube.com/playlist?list=PLkFVVqINj4rP8YemOQ-pBlyL0AXjmiPR9>

3.5 Clean Code. Czyli dobre praktyki pisania kodu!

Pisanie czytelnego kodu jest niezbędne aby zostać dobrym programistą. Poprawiając jakość kodu nie tylko sprawiamy że łatwiej jest go zrozumieć innym programistą, ale również kod staje się łatwiejszy w zarządzaniu nim. Dodanie nowego ficzera, czy przebudowa nie stanowi żadnego wyzwania właśnie dzięki zasadą "Clean Code". Do tego tematu również możemy dorzucić powszechnie stosowane koncepty takie jak - KISS, DRY, YAGNI czy SCA!

https://www.youtube.com/playlist?list=PLmmYSbUCWJ4x1GO839azG_BBw8rkh-zOj
<https://github.com/ryanmcdermott/clean-code-javascript>
<http://itcraftsman.pl/uzyteczne-koncepty-projektowe-kiss-dry-yagni-tda-oraz-separati-on-of-concerns/>
<https://gladiators-of-javascript.com/blog/jak-pisac-lepszy-kod-clean-code-w-js-i-w-ts/>

3.6 Zbiór dobrodziejstw jakie przyniósł ze sobą standard ES6

Standard ECMAScript 6(ES6) wprowadził wiele zmian do języka javascript dzięki którym życie programistów stało się o wiele łatwiejsze. Kluczowe w tytułowym standardzie było wprowadzenie nowych zmiennych (let, const), nowy sposób budowania funkcji (arrow function), nowy sposób tworzenia aplikacji opartych na programowaniu zorientowanym obiektowo(wprowadzenie Class) i wiele wiele innych

- <https://codeburst.io/es6-tutorial-for-beginners-5f3c4e7960be>
- <https://codeburst.io/es6-destructuring-the-complete-guide-7f842d08b98f>
- <https://codeburst.io/javascript-es6-the-spread-syntax-f5c35525f754>
<https://medium.com/javascript-scene/javascript-es6-var-let-or-const-ba58b8dcde75>
<https://medium.com/free-code-camp/write-less-do-more-with-javascript-es6-5fd4a8e50ee2>
- <https://github.com/lukehoban/es6features>

3.7 Co nowego dodano w ES2020

Świat z dnia na dzień prężnie się rozwija, a javascript razem z nim. Trzeba pamiętać, że aby coś wdrożyć na światło dzienne, trzeba najpierw solidnego pomysłu, potem realizacji, a na koniec dogłębnie to przetestować. Dlatego nowości nie wychodzą codziennie lecz raz na jakiś czas. A co przyniósł nam rok 2020?

<https://medium.com/javascript-in-plain-english/new-features-in-es2020-you-should-check-b4974d9d7edc>

3.8 Jak dobrze szukać rozwiązań dla naszych problemów

Nikt nie jest doskonały i w ludzkiej naturze leży fakt popełniania błędów, ale sztuką jest przyjąć feedback na klatę i sukcesywnie brnąć w rozwiązanie problemu! Nie zostawiaj go na potem, nie on się sam nie rozwiąże i tak będzie go widać, on nie jest niewidzialny! Całe szczęście programiści to pomocni ludzie i stworzyli kilka miejsc gdzie możemy dzielić się problemami jak i ciekawostkami! Oto kilka z nich

- <https://stackoverflow.com/>
- <https://dev.to/>
- <https://medium.com/>
- <https://github.com/explore>