

Zajęcia 2

Wstęp do programowania webowego

Po poznaniu gita przejdziemy do podstawowych zagadnień związanych z programowaniem aplikacji internetowych. Przed nami 4 języki do poznania. Na razie zajmujemy się tzw. front endem, czyli terminami odpowiadającymi za wygląd aplikacji, i interakcję z użytkownikiem. Do tego celu wykorzystamy język znaczników HTML i styli, tj. CSS. Trzeci składnik naszej "mieszanki", javascript, omówimy w innym czasie.

1. Pojęcia

Przed zabraniem się do kodowania, warto rozszerzyć wiedzę związaną z tą dziedziną. Wraz ze szczyptą znajomości angielskiego (tudzież pomocą tłumacza) będzie to banalnie proste.

a) Tworzenie programów komputerowych – od problemu do rozwiązania

Jak dobrze wiemy, wszystko co tworzymy jako programiści, w ostatecznym rozrachunku służy jednemu konkretnemu celowi – rozwiązaniu danego problemu. Upraszczając do granic maksimum proces myślowy stojący między pomysłem a programem, wygląda on następująco.

Problem -> Sposób jego rozwiązania -> Lista kroków postępowania (algorytm) -> implementacja (przepisanie czynności na kod zrozumiały dla komputera)

Nawet jeżeli nie będziemy rozpisywać na kartkach dziesiątek kroków które musimy wykonać, sam plan postępowania który wyimaginujemy w głowie można będzie przypasować do 3 pierwszych punktów przedstawionego „sznurka”. Oznacza to w skrócie, że proces ten zachodzi zawsze w programowaniu, nie ma od niego ucieczki. Więcej dowiedz się z zamieszczonego pod spodem przypisu, dokument ten zawiera między innymi pojęcia takie jak definicja algorytmu czy pseudokod. (**LEKTURA ZALECANA**)

Algorytm – pojęcia podstawowe - <http://home.agh.edu.pl/~byrska/src/5.Algorytmy.pdf>

b) Programowanie deklaratywne i imperatywne

Jako że będziemy zajmować się html'em i css'em, które są doskonałym przykładem deklaratywności, wytłumaczę pojęcia dwóch „metod” programowania. Każdy napisany przez nas kod (**pseudokod \neq kod**) przyjmie jedną z tych postaci. Są to jedno z tzw. *Paradygmatów programowania* (gr. Paradeigma – wzorec, przykład). Przekładając definicję książkową, wyobraź sobie programowanie deklaratywne jako takie, w którym podajesz komuś listę poleceń, jednak nie interesuje cię sposób ich wykonania, a efekt całej procedury. Programowanie imperatywne natomiast to takie, w którym każdą czynność musimy wcześniej zdefiniować słowo w słowo, a następnie wykonać. Zilustrujemy to przykładem.

DEKLARATYWNIE – Wojtku, wyjdź do sklepu, i kup pomarańcze, tutaj są pieniądze.

IMPERATYWNIE – Pójdź do Wojtka , zacznij z nim rozmawiać, poproś go o zakup produktu, w tym wypadku pomarańczy, wskaż mu sklep do którego ma iść, opcjonalnie podaj ilość produktu, przekaz mu pieniądze i wyjaśnij co ma zrobić z resztą.

Oczywiście mógłbym to rozszerzyć o wyższy poziom abstrakcji, jednak możemy przyjąć że Wojtek ma zakodowaną część czynności w głowie, i wie jak kupować pomarańcze w sklepie (co po części też można by podpisać pod gotowy zestaw instrukcji, który wywołujemy DEKLARATYWNIE). Na razie takie zrozumienie tych pojęć wystarczy nam w zupełności.

***Paradygmat programowania** - zestaw typowych dla danej grupy języków mechanizmów dostępnych programistom oraz sposobów ich interpretacji prze semantykę języka*

Więcej o paradygmatach programowania (LEKTURA OPCJONALNA) -

<http://ux.ap.krakow.pl/~jmp/pdf/Paradygmaty%20programowania-Byliny.pdf>

b) Poziomy języków programowania

Ostatnim z przytaczanych dzisiaj terminów będą poziomy języków programowania. Każdy z istniejących języków musi przynależeć do jednego z nich: wysokiego lub niskiego (**jest to klasyfikacja uproszczona !**). Na zajęciach będziemy zajmować się językami wysokiego poziomu, czyli takimi, które składnią są podobne do języka naturalnego (przykładowo: print „kup pomarańcze”);). Musimy wiedzieć jednak, że komputer zawsze ostatecznie będzie korzystał z tzw. kodu maszynowego, czyli standardowej kombinacji jedynek i zer (kodu binarnego), do którego wszystkie inne będą się kompilować.

***Kompilacja** – proces tłumaczenia kodu źródłowego na kod wynikowy (echo „kup pomarańczę” -> 110110101);*

***Transpilacja** – proces podobny do kompilacji, tłumaczenia kodu źródłowego również na źródłowy, jednak w innym języku programowania.*

***Poziomy języków programowania** - http://www.play-hookey.com/computers/language_levels.html*

2. HTML

Na początek zajmiemy się html'em, gdyż to w nim umieszczać będziemy pozostały kod naszych aplikacji internetowych.

HTML – hypertext markup language, umożliwia nam tworzenie elementów wyświetlanych w przeglądarce, w prosty, deklaracyjny sposób. Tworzymy go używając specjalnych znaczników, które są potem interpretowane w przeglądarce, i wyświetlane w postaci gotowej strony. Jest prosty w odbiorze, gdyż czytając kod orientujemy się jakie jest jego działanie, a także nie musimy pisać skomplikowanego kodu by uzyskać zadowalający nas efekt. Dlatego też HTML jest coraz szerzej używany nie tylko jako język przeglądarkowy, ale także jako domyślny przy budowie różnych interfejsów (występują jego odmiany, np. w androidzie interfejs użytkownika tworzy się w XML – extended HTML). Aktualnie używaną wersją jest HTML5. Na początek polecam przerobienie podstaw HTML, w artykule przygotowanym przez fundację mozilla. Wprowadzi on wszystkie podstawowe elementy, potrzebne nam w pracy na kolejnych zajęciach.

***KURS** - <https://developer.mozilla.org/pl/docs/Web/Guide/HTML/Wprowadzenie>*

3. CSS

Nie warto długo zwlekać. Choć ciekawy, HTML bez CSS jest jak głowa bez ciała – gotowa do działania, ale nie może bez niego funkcjonować. Szybko opiszę do czego służy, oraz podam ciekawe źródła, z których rozpoczniemy naukę.

CSS– Cascade Stylesheets – pozwala nam na nadanie stylu komponentom HTML, takich jak kolory, rozmiary, pozycjonowanie itd. Po jego poznaniu, będziemy gotowi do napisania pierwszych stron, takich jak wizytówki czy ogłoszenia, nie wymagających zaawansowanych interakcji z serwerem (treści dynamicznej, zmieniającej się w zależności od danych). Przygodę z jego użyciem rozpoczniemy od zasad elementarnych omówionych w kursie, na kolejnych zajęciach powinniśmy być jedynie rozeznani w podstawowej składni języka, wszystkie szczególne przypadki jego używania przedstawię i wytłumaczę na następnych zajęciach.

CSS na początek - https://developer.mozilla.org/pl/docs/Web/CSS/Na_pocz%C4%85tek

4. Warte zobaczenia.

Sekcja ta pojawia się pierwszy raz. Będę tu umieszczał ciekawe źródła, z których będziecie mogli dowiadywać się nowych informacji ze świata programowania (w tym wypadku Web developmentu, wybaczenie, nie jestem zorientowany w innych dziedzinach, nie zajmuję się nimi). Myślę że to dobry zbiór na start (mam tego trochę w zanadrzu ^^)

Smashing magazine – chyba najpopularniejsza strona z informacjami ze świata developerów. Artykuły na najwyższym poziomie, wspaniale napisane, wnoszą bardzo dużo do naszych umiejętności. Szczególnie polecam cotygodniowy felieton, „Web development reading list” zawierający nowinki ze świata programowania –

<http://smashingmagazine.com>

Sitepoint – strona poświęcona poradnikom na każdy temat z programowania, zawsze można znaleźć coś ciekawego. Podobny do smashingmagazine, większa ilość artykułów, jednak nie tak długich i szczegółowych. –

<http://sitepoint.com>

Stack overflow – Biblia, ucieczka grzesznych, ratunek udręczonych programistów. Możecie tam zadawać pytania dotyczące problemów związanych z programowaniem, i otrzymywać odpowiedź innych programistów z całego świata. Baza pytań wyrażana w dziesiątkach milionów to ostatnia deska ratunku developerów. Jeżeli czegoś nie ma na stacku, to albo jesteś geniuszem i stworzyłeś coś nowego, albo jest to niewykonalne. - <http://stackoverflow.com>

W3Schools– Strona z darmowymi kursami programowania, prawie każdy zaczynał tu swą przygodę z programowaniem webowym, teraz dalej jest świetnym źródłem informacji, aczkolwiek twierdzą, że jest teraz dużo ciekawszych źródeł uczących tego samego. Nie mniej na początek jest genialnym źródłem, tyle że w języku angielskim (ojczystym programistów, musicie go umieć) – <https://w3schools.com>

MDN – Mozilla developer Network – zawiera aktualnie praktycznie całą dokumentację związaną z językami webowymi : HTML, CSS, Javascript . Używaj jako podręcznika ! - <https://developer.mozilla.org/pl/>