

Temat: Co to jest aplikacja internetowa? Technologie projektowania aplikacji internetowych – porównanie, charakterystyka.

Profil: Technik Informatyk (Technikum Informatyczne)

Klasa: 1

Przedmiot: E14.3

Spis treści

1 Co to jest aplikacja internetowa	2
1.1 Architektura warstwowa	2
1.2 Wzorce projektowe	4
1.2.1 Wzorzec MVC. Model-View-Controller (pol. Model-Widok-Kontroler)	5
1.3 Framework – platforma programistyczna	6
1.4 6 zasad projektowania aplikacji webowych	6
2 Technologie projektowania aplikacji internetowych – opis, porównanie	8
2.1 Technologie po stronie klienta	8
2.1.1 HTML5	8
2.1.2 CSS	10
2.1.3 Adobe Flash/Flex.	10
2.2 Java	11
2.2.1 Java Script	12
2.2.2 AJAX	13
2.3 Technologie po stronie serwera	14
2.3.1 CGI (ang. Common Gateway Interface)	14
2.3.2 PERL	15
2.3.3 JSP (ang. JavaServer Pages)	15
2.3.4 ASP (ang. Active Server Pages)	16
2.3.5 ASP.NET	17
2.3.6 PHP	17
2.3.7 Ruby on Rails	19

1 Co to jest aplikacja internetowa

Aplikacja internetowa, (ang.) *web application* – zwana również aplikacją webową. Jest to oprogramowanie stworzone przy użyciu kilku technologii (najważniejsze to: HTML5, CSS, JavaScript), działające w architekturze klient-serwer (gdzie przeglądarka występuje w roli klienta), mogące komunikować się z innymi aplikacjami bądź serwerami, wykonująca określone operacje i działania w odpowiedzi na akcje użytkownika. Przykładami typowych aplikacji webowych są:

- webowe interfejsy portali społecznościowych (FB, Google+, Twitter, LinkedIn)
- klient poczty (GMail, Outlook Web Access, interfejsy poczty WP, Onet, O2 itp.),
- sklepy internetowe, transakcyjne systemy bankowe,
- webowe interfejsy komunikatorów, IRC, gry MMORPG.

Z tego względu można rozróżnić rozwiązania stosowane po stronie klienta (przeglądarka) i po stronie serwera. Należy zwrócić tutaj uwagę na to, że nie jest to zwykła statyczna strona internetowa. Głównym założeniem jest interakcja z użytkownikiem.

W pracy aplikacji internetowej musi pośredniczyć serwer WWW. Do przygotowania samej aplikacji używa się różnych mechanizmów (np. **CGI, JSP, ASP.NET**) i języków (np. **PHP, Java, C#**), jak również serwerów aplikacji. Mechanizm prezentacji danych w przeglądarce określa się czasem mianem cienkiego klienta.

Generalnie:

- Strony internetowe zorientowane są w większym stopniu na dostarczanie informacji.
- Aplikacje webowe stanowią przede wszystkim narzędzie realizacji konkretnych zadań służących określonemu celowi.

Język skryptowy – tzn. tworzone w nim programy są po prostu plikami tekstowymi, które następnie są wykonywane przez interpreter danego języka. Nie jest tu wymagana kompilacja kodu do postaci pliku wykonywalnego - wszystkie polecenia tłumaczone są i wykonywane "w locie". Przetwarzając skrypt, dany język czyta plik w całości, sprawdza poprawność składni, "kompiluje" go po czym wykonuje. Dzięki temu programista może modyfikować i wykonywać skrypty znacznie szybciej niż np. w przypadku języka C, gdzie po każdej zmianie konieczna jest dodatkowo kompilacja. Ma to jednak także i wady, a najpoważniejsza chyba to taka, iż o ile w przypadku C niektóre błędy mogą zostać wyłapane już na etapie kompilacji kodu, to tutaj nie ma takiej możliwości, przez co są one dużo trudniejsze do znalezienia i naprawienia.

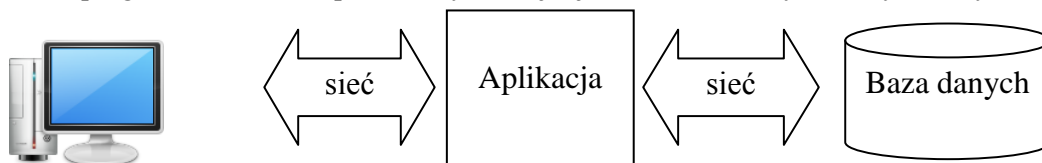
Źródło: <http://wieluk.republika.pl/s/operlu.html>

1.1 Architektura warstwowa

W zależności od przeznaczenia systemu należy dobrać odpowiedni model warstwowy przygotowywanej aplikacji. Model ten powinien zależeć od stawianych wymagań funkcjonalnych.

W ramach architektury aplikacji modele warstwowe są często tworzone w oparciu o wzorce projektowe.

Podstawowa architektura aplikacji internetowej zakłada, że na komputerze użytkownika funkcjonuje wyłącznie oprogramowanie obsługi interfejsu użytkownika, natomiast oprogramowanie właściwej aplikacji oraz oprogramowanie dostępu do danych znajdują się na oddzielnych, dedykowanych komputerach.



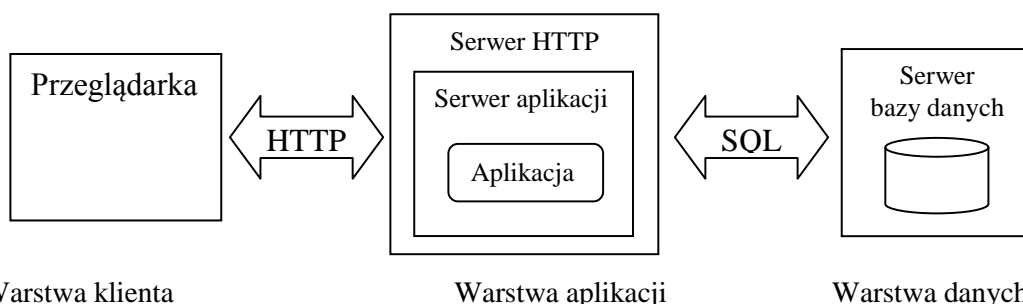
Interfejs użytkownika
(przeglądarka internetowa)

Aplikacje internetowe – porównanie



Rozszerzona architektura aplikacji internetowej zakłada, że w jej skład wchodzi trzy warstwy:

- Klienta – odpowiedzialna za interfejs użytkownika i interakcję z użytkownikiem, oparta na kliencie HTTP
- Aplikacji – która w odpowiedzi na żądania klientów tworzy dokumenty dynamiczne, składa się z serwera HTTP i serwera aplikacji będącego środowiskiem uruchomieniowym dla aplikacji generujących dokumenty dynamiczne
- Danych – odpowiada za udostępnianie informacji umieszczonych w dokumentach dynamicznych, składa się z serwera bazy danych, który odpowiada na żądania w języku SQL



Każda z tych warstw jest budowana z użyciem innych technologii. Kluczowy jest interfejs uruchamiany w przeglądarce. Oprogramowanie to jest tworzone głównie w JavaScript.

Komponentowe aplikacje www

Aplikacje internetowe buduje się najczęściej w ten sposób, że poszczególne funkcje systemowe w celu ich separacji są rozdzielone pomiędzy różne komponenty.

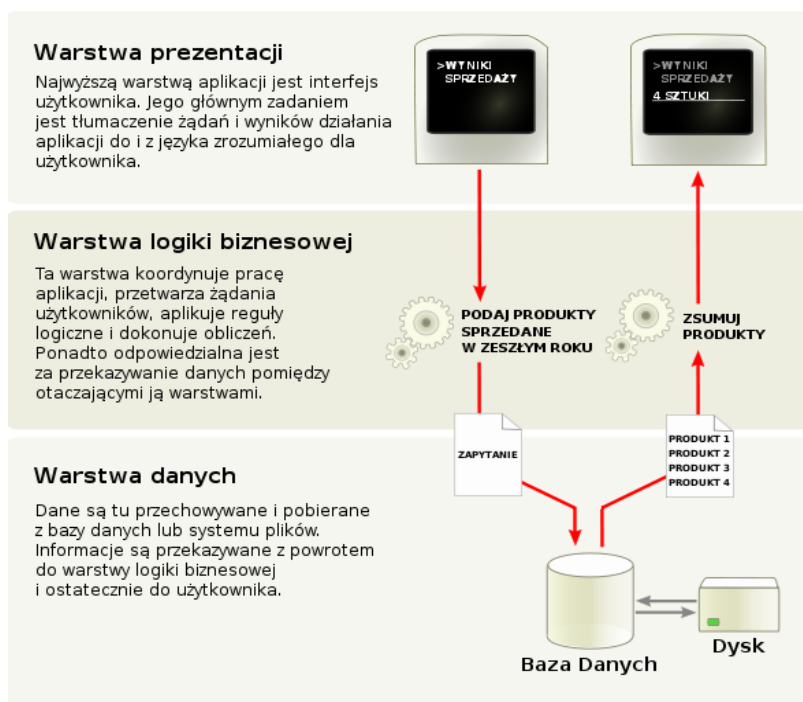
Stosowane są dwie metody:

- Architektura 4-warstwowa:
 - warstwa klienta
 - logiki prezentacji (przyjmowanie żądań od klientów)
 - logiki biznesowej (między innymi komunikacja z bazą danych)
 - danych
- Architektura Model-View-Controller



Składniki aplikacji podzielone są na składniki:

- sterujące (controller) : przyjmują żądania od klientów oraz koordynują ich obsługę
- prezentacji (view) : tworzą dokumenty dynamiczne i wypełniają je danymi dostarczonymi przez składniki modelu.
- modelu (model) : odpowiadają za realizację procesów biznesowych i komunikację z bazą danych.



1.2 Wzorce projektowe

Wzorec projektowy (ang. *design pattern*) – w inżynierii oprogramowania, uniwersalne, sprawdzone w praktyce rozwiązanie często pojawiających się, powtarzalnych problemów projektowych. Pokazuje powiązania i zależności pomiędzy klasami oraz obiektami i ułatwia tworzenie, modyfikację oraz pielęgnację kodu źródłowego. Jest opisem rozwiązania, a nie jego implementacją. Wzorce projektowe stosowane są w projektach wykorzystujących programowanie obiektowe.

Dokumentacja wzorca projektowego powinna zawierać następujące informacje:

- Nazwa wzorca oraz klasyfikacja: opisowa oraz unikalna nazwa, umożliwiająca identyfikację oraz odwoływanie się do wzorca; klasyfikacja według jednego ze schematów.
- Przeznaczenie: opis celu, który stoi za wzorcem oraz powody, jakimi należy się kierować podczas jego wyboru.
- Inne nazwy: jeżeli istnieją inne, dobrze znane nazwy wzorca, należy je podać.
- Motywacja: scenariusz zawierający problem powiązany z kontekstem, w którym wzorec może być stosowany.
- Stosowalność: sytuacje, w których wzorec może być użyteczny.
- Struktura: graficzna reprezentacja wzorca, zwykle jako diagram klas lub diagram interakcji.
- Uczestnicy: lista klas i obiektów stosowanych w tym wzorcem oraz ich zobowiązania.
- Współpraca: opis wzajemnej interakcji klas i obiektów wykorzystywanych we wzorcem.
- Konsekwencje: wykaz wyników, efektów ubocznych oraz kompromisów jakie występują podczas użycia wzorca.
- Implementacja: wskazówki dotyczące implementacji wzorca; zwrócenie uwagi na specyficzne kwestie.
- Przykładowy kod: przykład zastosowania wzorca z wykorzystaniem jednego z języków programowania.
- Przykłady zastosowania: znane przykłady zastosowania wzorca w rzeczywistych programach.
- Pokrewne wzorce: odniesienie wzorca do innych, z którymi wiąże się przez wspólne stosowanie lub można go z nimi zamienić oraz przedstawienie różnic w stosunku do podobnych wzorców.

1.2.1 Wzorzec MVC. Model-View-Controller (pol. Model-Widok-Kontroler)

Jest jednym z najczęściej stosowanych wzorców projektowych w aplikacjach internetowych. Służący do organizowania struktury aplikacji posiadających graficzne interfejsy użytkownika.

Z reguły proste aplikacje w PHP wyglądają tak, że sprawdzany jest adres wpisany do przeglądarki (na podstawie `$_GET`, bądź innych właściwości z `$_SERVER`), potem includowane są potrzebne pliki w zależności od pozyskanych danych. Absolutne podstawy podstaw.

Stosując MVC nie możemy opierać się, jak to zwykle bywa w tradycyjnym podejściu, tylko na jednym pliku, np. `index.php`, w którym mieszamy kod prezentacji (np. HTML) i logikę biznesową (dane np. z MySQL).

Działa to zupełnie inaczej, a mianowicie mamy 3 główne elementy aplikacji, gdzie każda odpowiedzialna jest tylko za jedną rzecz. Są nimi:

- Model zajmuje się tylko i wyłącznie pracą na danych.
- Widok (View) zajmuje się tylko i wyłącznie wyświetlaniem danych.
- Controller przyjmuje dane wejściowe od użytkownika i reaguje na jego poczynania, zarządzając aktualizacje modelu oraz odświeżenie widoków. Steruje wszystkim tym, co dzieje się podczas uruchamiania danej akcji. Np. podczas wywołania `index.php?controller=news&id=55`.

Przykładowo, jeśli dodasz np. nowego usera do bazy, to kontroler zdecyduje, że po tej czynności przeniesiony zostaniesz np. do edycji profilu. Kontroler to mózg sterujący całym wywołaniem strony. W pliku kontrolera wybierasz, z którego modelu (a więc pliku odpowiedzialnego za pobieranie danych – może to być baza danych, plik, cokolwiek) weźmiesz treść dla strony, a także jak ją wyświetlisz (czyli który widok Cię interesuje).

Dobłą praktyką jest umieszczanie kontrolerów w folderze np. `/controllers`, widoków w `/views`, a modeli w `/models`. W ten sposób mamy jasny i prosty podział naszej aplikacji.

Jeśli więc uruchomimy przykładowo stronę `index.php?controller=news¶m=55`, według MVC powinny działać się następujące rzeczy:

- Funkcja/Klasa odpowiedzialna za parsowanie (analizie) adresu (np. Router) pobiera go i na podstawie konfiguracji stwierdza, które słowo/zmienna w adresie oznacza kontroler, a które parametry doń przekazywane. Posiłkując się powyższym URL, kontrolerem będzie słówko **news**, a parametrem **55**.

Dzięki takiemu parsowaniu adresów możemy łatwo zarządzać i dołączać kontrolery, które będą stanowić serce aplikacji. Po co stosować `switch` czy `if` do sprawdzania, co zostało wpisane w pole adresu, skoro możemy od razu, jeśli takowy istnieje, dołączyć stosowny plik.

Następnie, na podstawie uzyskanych danych, includujemy potrzebne pliki. Jak już się przekonał, potrzebny nam jest kontroler. Sprawdzamy więc, czy w folderze, który umownie wybraliśmy (`controllers`) znajduje się plik `news.php` (tak, `news` pochodzi z adresu strony).

- Następnie, mając załadowany kontroler, uruchamiamy jego defaultową metodę/funkcję, bo oprócz nazwy kontrolera nie mamy niestety nazwy żadnej jego akcji, którą moglibyśmy wywołać (gdyby adres miał postać `index.php?controller=news&action=delete&id=55`, można by posłużyć się zmienną `$_GET['action']` i uruchomić coś takiego:

```
$controller->$_GET['action']($_GET['id'])
```

W takim razie używamy `$controller->index()`, bądź coś podobnego – wszystko zależy od konfiguracji.

UWAGA! Dane \$_GET należy przefiltrować pod kątem złośliwych ataków. Z reguły robi to klasa odpowiedzialna za parsowanie adresu, bądź specjalna klasa obsługująca dany request (wywołanie adresu) pod kątem danych przekazanych przez użytkownika.

Źródło:

<http://ferrante.pl/frontend/php/wzorzec-mvc-w-php/>

1.3 Framework – platforma programistyczna

Do tworzenia aplikacji internetowych często wykorzystuje się platformy programistyczne, które definiują strukturę aplikacji, określają mechanizm ich działania oraz dostarczają bibliotek umożliwiających wykonywanie określonych zadań.

Framework posiada zdefiniowaną podstawową strukturę aplikacji, czyli elementy, które pozostają niezmiennie we wszystkich utworzonych przy jego pomocy aplikacjach. Programista tworzy aplikację, dostosowując poszczególne komponenty do wymagań realizowanego projektu. Ma on również możliwość tworzenia nowych elementów niezbędnych w projektowanej strukturze aplikacji.

Tworzenie aplikacji z użyciem Framework wymaga mniejszej ilości kodu do napisania. Aplikacja posiada dobrą wewnętrzną strukturę, jest dobrze zaprojektowana i przetestowana. Może występować niższa wydajność tworzonego oprogramowania. Frameworki mogą być stosowane jako szkielety kompletnych aplikacji jak również pojedynczych komponentów.

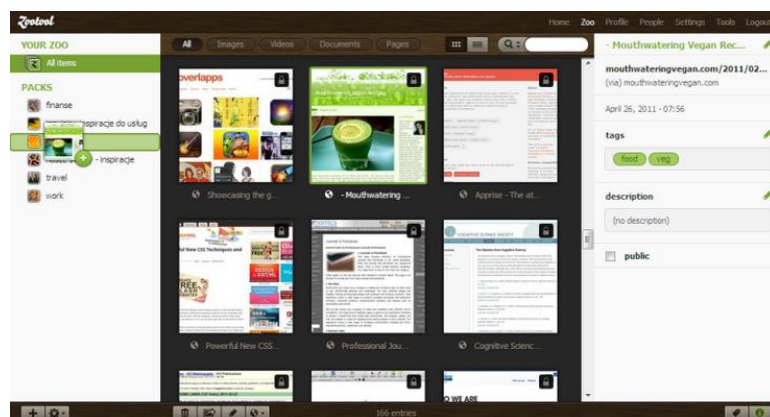
1.4 6 zasad projektowania aplikacji webowych

Autor: Ewa Sobula (<http://uxbite.com/2011/07/projektowanie-aplikacji-webowych-dla-poczatkujacych/>)

1. Umożliwaj bezpośrednią interakcję z aplikacją

Jeśli użytkownik może edytować pewne elementy, należy pozwolić mu na to w ich obrębie, bez przenoszenia go do osobnej strony. Prezentacja i edycja obiektu powinny się odbywać w jednym miejscu (np. poprzez drag & drop). Dzięki temu skracamy czas potrzebny na wykonanie tej czynności.

Np. Przenoszenie elementów do katalogów za pomocą funkcji drag & drop na przykładzie ZooTool.com



2. Minimalizuj wysiłek potrzebny do jej obsługi

Nie należy wymagać, aby użytkownik uczył się, domyślał i pamiętał, jak obsługiwać dany interfejs. W zamian powinien on dostarczać mu kontekstowe wskazówki, dzięki którym w każdym momencie korzystania z aplikacji będzie wiedział, co zrobić. W szczególności dotyczy to udostępniania narzędzi w zależności od kontekstu. Np. narzędzia do edycji tekstu powinny być udostępnione użytkownikowi w momencie i w miejscu, w którym edytuje dany element.

Dodatkowe opcje dostępne są dopiero po wybraniu zadania z listy - RemeberTheMilk.com



3. Utrzymuj użytkownika w obrębie strony

Przechodzenie od strony do strony czy przeglądowanie jej treści zaburza swobodne korzystanie z aplikacji. Zamiast tego lepiej wykorzystać sposoby prezentacji zawartości w obrębie strony, takie jak np. modale, doczytywanie treści wraz z postępem przewijania strony czy dynamiczne wyświetlanie treści.

Facebook pozwala na doczytywanie starszych postów, lajkowanie oraz komentowanie ich w obrębie jednego widoku.



4. Pomagaj użytkownikowi odkrywać funkcjonalności

Jednym z podstawowych wyzwań w projektowaniu aplikacji jest kwestia przekazania użytkownikom wiedzy o dostępnych funkcjonalnościach, nauczenia ich i zachęcenia do ich wykorzystywania. Użytkownik nie będzie korzystał z funkcji, o której istnieniu nie ma pojęcia. Ta zasada dotyczy dostarczania użytkownikom informacji jeszcze przed interakcją.

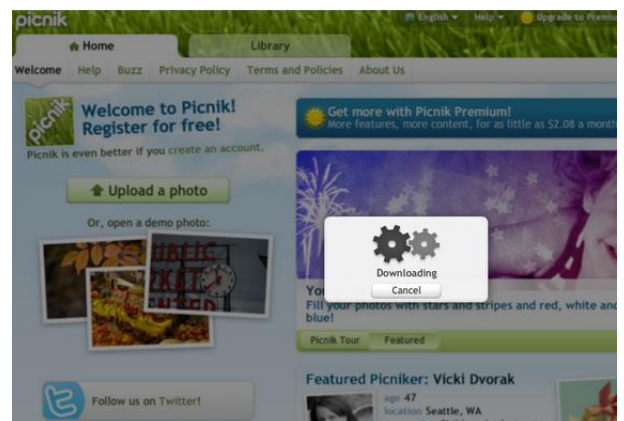
Grooveshark już na stronie głównej wyjaśnia, jak z niego korzystać.



5. Wykorzystuj efekty przejścia (rozsądnie)

W aplikacjach szeroko wykorzystywane są efekty przejścia, takie jak rozjaśnianie i przyciemnianie, rozwijanie i zwijanie, podświetlanie czy inne animacje. Służą one prezentacji informacji zwrotnej podczas trwania interakcji. Nawet niewielki ruch znakomicie przyciąga uwagę użytkownika. I właśnie dlatego stosowanie tych rozwiązań powinno być przemyślane i uargumentowane, aby faktycznie użytkownikowi pomagać, a nie przeszkadzać.

Efekt przyciemnienia tła - Picnic.com



Instapaper pokazuje okienko zapisu artykułu do późniejszego przeczytania i znika zaraz po zakończeniu zapisu.



6. Reaguj na działania użytkownika w czasie rzeczywistym

Aplikacja powinna szybko reagować na to, co robi użytkownik i wychodzić mu naprzeciw – czyli dostarczać mu informacji zwrotnej także po interakcji. Ten feedback powinien być odpowiedni oraz szybki. Przykładem jest autouzupełnianie, podgląd zmian w czasie rzeczywistym czy pasek postępu.

Podpowiadanie wyszukiwanych fraz w odpowiedzi na wpisywaną przez użytkownika treść.



2 Technologie projektowania aplikacji internetowych – opis, porównanie

Przeglądarki internetowe można traktować jako środowisko działania aplikacji tak samo jak system operacyjny, pełnią one funkcje interfejsu użytkownika. Do pisania aplikacji w takim środowisku może służyć język JavaScript (jest on odpowiednikiem assemblera) lub przy użyciu narzędzi pozwalających na generowanie obiektów osadzanych na stronie internetowej.

Interfejs użytkownika

Zasadnicza różnica między różnymi rozwiązaniami dotyczy rodzaju przesyłanych struktur danych:

1. Przygotowanie wyglądu interfejsu po stronie serwera. Jest to najstarsze rozwiązanie. Aplikacja na serwerze (PHP Python/Django etc) odczytuje dane i generuje wygląd strony do wyświetlenia (rendering). Strony w języku HTML są przekazywane poprzez serwer www. Modyfikację wyglądu poszczególnych elementów strony uzyskuje się dzięki arkuszom stylu (CSS).
2. Wykorzystanie XML. W arkuszach XML są wysyłane tylko dane. Wygląd interfejsu uzyskuje się dzięki zastosowaniu arkuszy XSL.
3. Dynamiczna wymiana danych między GUI a serwerem. Może to odbywać się przy użyciu tradycyjnych stron HTML dzięki zastosowaniu technologii Ajax, albo poprzez klasyczną aplikację działającą w środowisku przeglądarki.

Na rynku jest obecnie wiele różnych technologii tworzenia aplikacji internetowych. Stworzenie kompletnej i aktualnej listy jest trudne (a niewiadomo, czy w ogóle możliwe), ponieważ dziedzina ta stale się rozwija, a liczba pakietów oprogramowania szybko rośnie.

2.1 Technologie po stronie klienta

Są związane z rodzajem oprogramowania klienckiego, czyli przeglądarki int. i zainstalowanych wtyczek.

2.1.1 HTML5

HTML (ang. *HyperText Markup Language*) – hipertekstowy język znaczników, wykorzystywany do tworzenia i prezentowania stron internetowych www.



Zgodnie z zapowiedziami prace nad specyfikacją HTML5 zostaną ukończone w 2014 roku - wtedy też nowy standard zostanie oficjalnie ogłoszony i zacznie obowiązywać. Dwa lata później - w 2016 roku - ma być gotowy HTML5.1; wówczas ma być także zaprezentowany szkic standardu HTML5.2.

Główna różnica pomiędzy HTML 5 a XHTML 2 polega na tym, że HTML 5 jest pisane z myślą o kompatybilności wstecznej, podczas gdy XHTML 2 zrywało z kompatybilnością.

Jakie nowe możliwości przynosi HTML5

Większość zmian we wzorcu HTML (w porównaniu z wersją 4) odnosi się do sposobu obsługi grafiki, zdjęć, wideo oraz muzyki. HTML5 znacznie zwiększa też możliwości modyfikowania wyglądu i zachowania tekstów. Oto najważniejsze zmiany w nowej wersji języka stron WWW:

- **Wideo i audio:** Odtwarzanie filmów wideo i utworów audio bezpośrednio w przeglądarce nie wymaga dodatkowych programów (na przykład Flash). HTML5 zawiera również elementy, za pomocą których można sterować odtwarzaniem.
- **Fotogalerie:** W HTML5 zdjęcia mogą być prezentowane na stronie w wygodniejszy i atrakcyjniejszy niż dotąd sposób. Nowa wersja HTML dostarcza wielu efektów, na przykład przejść między zdjęciami, co pozwala na tworzenie ciekawych galerii fotografii. Do tej pory stworzenie atrakcyjnego pokazu wymagało użycia wtyczek.
- **Integracja tekstu i obrazu:** W HTML5 przybyło mnóstwo opcji dotyczących wyświetlania na stronach grafiki i tekstu. Dzięki temu napisy mogą mieć nie tylko charakter informacyjny, ale również dekoracyjny.
- **Wprowadzanie danych:** Wprowadzanie danych w wyświetlanych na stronie formularzach ma być znacznie łatwiejsze i bezpieczniejsze. Dzięki HTML5 przeglądarka natychmiast rozpoznaje wpisaną w pole wartość i na przykład ostrzeże, że wpisany w pole dotyczące miasta tekst jest adresem e-mail. W HTML4 kontrola formularza może być przeprowadzona dopiero po przesłaniu go do serwera - to dość niewygodne, bo informacja o błędzie w polu formularza wyświetla się z dużym opóźnieniem.
- **Grafika i animacje:** Grafika ze zmieniającą się treścią (na przykład wyświetlane na bieżąco wykresy czy kompozycje różnych fotografii), a także animacje i proste gry są w HTML5 zintegrowane ze stroną WWW. Wszystko dzięki nowemu elementowi o nazwie *Canvas*, który pozwala na dynamiczne renderowanie kształtów i obrazów.

HTML 5 znacznie ułatwia życie użytkownikom internetu:

- **Grafika i animacje:** HTML5 sprawia, że nie trzeba instalować w przeglądarce tylu wtyczek, co teraz. To nie tylko zwiększa wygodę, ale też poprawia bezpieczeństwo, bo nie brak w sieci stron z dodatkami zagrażającymi na przykład naszej prywatności. Zastosowanie dodatkowych programów w HTML 5 konieczne jest tylko w wyjątkowych przypadkach.
- **Uniwersalne zastosowanie:** HTML5 jest znacznie lepiej od swojego poprzednika przygotowany do pracy na urządzeniach innych niż komputery biurkowe i laptopy. Strony napisane w nowej wersji HTML będą mogły być bez problemów obsługiwane przez tablety, smartfony, a także telewizory z funkcjami internetowymi. Na przykład wykorzystująca HTML4 i Flasha strona YouTube nie jest obsługiwana przez niektóre urządzenia mobilne, a wersja YouTube wykonana w HTML5 działa na nich bez żadnych problemów. W urządzeniach przenośnych nie da się też często zainstalować niezbędnych do poprawnego wyświetlenia witryny wtyczek - w HTML 5 ten problem w większości wypadków znika.
- **Identyczny wygląd strony:** Obecnie ta sama strona otwarta w różnych przeglądarkach wygląda nieco inaczej (różnice są przede wszystkim w szczegółach, ale to głównie dlatego, że webmasterzy za pomocą różnych trików starają się je maksymalnie zniwelować). Strony wykonane w HTML 5 będą w każdej przeglądarce wyglądać tak samo.

Obecnie tylko nieliczne strony wykonane są w HTML5. Jednak przeglądarki internetowe poprawnie wyświetlają wiele elementów tego języka. Na stronie www.html5test.com można sprawdzić, jak w teście obsługi HTML5 wypada nasza przeglądarka

Przykład:

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="utf-8" />
    <script src="skrypt.js"></script>
    <link href="styl.css" rel="stylesheet">
  </head>
  <body>

  </body>
</html>
```

2.1.2 CSS



Kaskadowe arkusze stylów (ang. *Cascading Style Sheets*, w skrócie CSS) to język służący do opisu formy prezentacji (wyświetlania) stron WWW. CSS został opracowany przez organizację W3C w 1996 r. Pierwszy szkic CSS zaproponował w 1994 r. Håkon Wium Lie.

Arkusze stylów CSS to lista dyrektyw (tzw. reguł) ustalających w jaki sposób ma zostać wyświetlana przez przeglądarkę internetową zawartość wybranego elementu (lub elementów) (X)HTML lub XML. Można w ten sposób opisać wszystkie pojęcia odpowiedzialne za prezentację elementów dokumentów internetowych, takie jak rodzina czcionek, kolor tekstu, marginesy, odstęp międzywierszowy lub nawet pozycja danego elementu względem innych elementów bądź okna przeglądarki. Wykorzystanie arkuszy stylów daje znacznie większe możliwości pozycjonowania elementów na stronie, niż oferuje sam (X)HTML.

CSS został stworzony w celu odseparowania struktury dokumentu od formy jego prezentacji. Separacja ta zwiększa zakres dostępności witryny, zmniejsza złożoność dokumentu, ułatwia wprowadzanie zmian w strukturze dokumentu. CSS ułatwia także zmiany w renderowaniu strony w zależności od obsługiwane medium (ekran, palmtop, dokument w druku, czytnik ekranowy). Stosowanie zewnętrznych arkuszy CSS daje możliwość zmiany wyglądu wielu stron naraz bez ingerowania w sam kod (X)HTML, ponieważ arkusze mogą być wspólne dla wielu dokumentów.

Przykład:

```
body /* Nazwa znacznika BODY */
{
    background-color: #0f0;
}

p
{
    color: #00f;
}

.pogrubienie
{
    font-weight: bold;
}
```

2.1.3 Adobe Flash/Flex.



Adobe Flash to najstarsza i chyba najlepsza technologia, stworzona przez firmę Macromedia służąca do tworzenia dynamicznych prezentacji w internecie. Flash to głównie animacja i efektowna grafika, to także możliwość oglądania filmów w oknie przeglądarki.

Strona we Flashu to nie tylko nowoczesny wygląd, ale także możliwości interaktywne. Zbudowanie strony we Flashu jest droższe niż w HTML-u. Technologia ta jest bardziej skomplikowana, choć w prosty sposób możliwe jest osiągnięcie rzeczy niemożliwych do zrobienia w HTML.

Technologia Flash jest też trochę bardziej pracochłonna. Właściwie każdy projekt powstawać musi od podstaw. Często istnieje także konieczność zbudowania alternatywnej wersji w HTML (dla grupy starszych internautów), co zwiększa koszty całego projektu strony.

Na dodatek efekty wynikające z technologii Flash nie są dostępne dla wszystkich. Konieczne jest tutaj zainstalowanie odpowiedniej wtyczki. Problemem jest także zapanowanie nad tak zbudowaną stroną internetową w niektórych przeglądarkach.

Wady i zalety Flash

Zalety:

- możliwość tworzenia multimedialnych gadżetów, podnoszących atrakcyjność strony (np. gier online),
- możliwość lepszego prezentowania nagrań audio/wideo bezpośrednio na stronie,
- możliwość prezentowania dynamicznych wykresów,
- łatwiejsze osiągnięcie ciekawych efektów wizualny strony.

Wady:

- konieczność zainstalowania specjalnego oprogramowania w celu pełnego korzystania z możliwości Flasha,
- wolniejsze wczytywanie się stron zbudowanych we Flashu, spowalnianie działania przeglądarki,
- wyższe wymagania sprzętowe,
- niedostępność lub ograniczona dostępność technologii Flash w urządzeniach mobilnych,
- utrudniona możliwość pozycjonowania, potrzeba większych nakładów pracy,
- wysokie koszty stworzenia od podstaw strony internetowej.

Wydaje się, że Flash powoli ustępuje technologii AJAX, HTML5 dzięki której między innymi FlashPlayer nie jest już potrzebny.

Rozszerzeniem technologii Flash jest Flex. Rozszerzenie polega na lepszym wsparciu dla budowy interfejsów "okienkowych" oraz powiązaniu z serwerami aplikacji.

2.2 Java



Obiektowy język programowania stworzony przez grupę roboczą pod kierunkiem Jamesa Goslinga z firmy Sun Microsystems. Java jest językiem tworzenia programów źródłowych kompilowanych do kodu bajtowego, czyli postaci wykonywanej przez maszynę wirtualną. Jego podstawowe koncepcje zostały przejęte z języka Smalltalk (maszyna wirtualna, zarządzanie pamięcią) oraz z języka C++ (duża część składni i słów kluczowych).

Jawy nie należy mylić ze skryptowym językiem JavaScript, z którym wspólną ma jedynie składnię podstawowych instrukcji.

Projekt Java od początku zakładał rozwiązania sieciowe i wielowarstwowe. Większość przeglądarek internetowych integruje się z tym rozwiązaniem, dzięki czemu aplikacje mogą być osadzone na stronach internetowych. W praktyce stosuje się w aplikacjach internetowych w Javie wyłącznie serwery aplikacji (Tomcat, Glassfish lub JBoss).

Dla uatrakcyjnienia tego rozwiązania i sprostania konkurencji Adobe stworzono projekt JavaFX. Umożliwia on tworzenie prezentacji multimedialnych takich jak w przypadku Flash'a.

Klasą dla siebie w tej dziedzinie jest darmowy produkt Google: GWT.

GWT umożliwia programowania aplikacji internetowych całkowicie w języku Java! Interfejs jest następnie tłumaczony na język JavaScript. Dobre wprowadzenie do GWT można znaleźć na stronie

Przykład:

```
public class HelloWorld
{
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

2.2.1 Java Script



JS – skryptowy język programowania, stworzony przez firmę Netscape, najczęściej stosowany na stronach internetowych.

Najczęściej spotykanym zastosowaniem języka JavaScript są strony WWW. Skrypty służą najczęściej do zapewnienia interaktywności poprzez reagowanie na zdarzenia, sprawdzania poprawności formularzy lub budowania elementów nawigacyjnych. Skrypty JavaScriptu uruchamiane przez strony internetowe mają znacznie ograniczony dostęp do komputera użytkownika.

Obecna wersja to 1.8, trwają prace nad wersją 2.0, przykładowe zmiany w stosunku do wersji 1.x

- Przyjrzyjmy się zatem, jak wyglądało tworzenie „klasy” w JS 1.x, a jak będziemy tworzyli klasy w JS 2.0:

```
/* JavaScript 1.x definicja "klasy" */

function Foo() {
    this.a = "a";
    this.b = "b";
}
var myFoo = new Foo(); // tworzenie instancji klasy

/* JavaScript 2.0 definicja klasy */

class Bar {
    this.a = "a";
    this.b = "b";
}
var myBar = new Bar(); // tworzenie instancji klasy
```

- Definiowanie stałych będzie wyglądało tak:

```
/* stała w JavaScript 1.x */
var FOO = 'bar'; // jestem zapisana wielkimi literami,
proszę mnie nie ruszać...

/* stała w JavaScript 2.0 */
const FOO = 'bar'; // spróbuj mnie ruszyć.
```

Pełny przegląd możliwości JS 2.0 ma obecnie ponad 40 stron. Zainteresowani mogą zobaczyć go jako plik PDF po linku:

<http://www.ecmascript.org/es4/spec/overview.pdf>

2.2.2 AJAX



AJAX (ang. *Asynchronous JavaScript and XML*, asynchroniczny JavaScript i XML) – technika tworzenia aplikacji internetowych, w której interakcja użytkownika z serwerem odbywa się bez przeladowywania całego dokumentu, w sposób asynchroniczny. Ma to umożliwić bardziej dynamiczną interakcję z użytkownikiem niż w tradycyjnym modelu, w którym każde żądanie nowych danych wiąże się z przesłaniem całej strony HTML.

Przykład:

```
<script type="text/javascript">
function ajaxFunction()
{
var xmlhttp;
if (window.XMLHttpRequest)
    {
    // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
    }
else if (window.ActiveXObject)
    {
    // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
else
    {
    alert("Your browser does not support XMLHttpRequest!");
    }
}
</script>
```

Popularne biblioteki JavaScript powiązane z AJAX-em

Prototype biblioteka ułatwiająca korzystanie z możliwości oferowanych przez AJAX

jQuery lekka biblioteka programistyczna dla języka JavaScript, ułatwiająca korzystanie z JavaScriptu (w tym manipulację drzewem DOM).



jQuery pozwala w wygodny i zrozumiały sposób korzystać z następujących funkcjonalności:

- selektory – umożliwiają wybranie dowolnego podzbioru węzłów modelu DOM
- atrybuty – jQuery pozwala przetwarzać atrybuty węzłów dokumentu
- manipulowanie modelem DOM
- zmiana i przypisywanie stylu do elementów
- rozbudowana obsługa zdarzeń, możliwość definiowania własnych
- efekty – animacje
- AJAX – prosty interfejs realizujący asynchroniczne zapytania

Cechy:

- Niezależność od przeglądarki – eliminuje konieczność dostosowywania kodu do różnych przeglądarek WWW
- pełna funkcjonalność w: IE 6.0+ (oprócz interfejsu AJAX), Firefox 2+, Safari 3.0+, Opera 9.0+, Chrome
- Obsługa selektorów zgodna z CSS3
- Małe rozmiary – produkcyjna wersja to tylko ok 80 kB
- Wygoda tworzenia wtyczek

Script.aculo.us rozszerza Prototype ułatwiając tworzenie animacji i interfejsów

MooTools modułowa biblioteka AJAX zawierająca również ułatwienia do tworzenia efektów wizualnych

2.3 Technologie po stronie serwera

Klient otrzymuje gotową stronę internetową wygenerowaną z uwzględnieniem kryteriów określonych przez użytkownika.

Dany język skryptowy nie jest interpretowany (przetwarzany) przez przeglądarkę, lecz przez specjalny program na serwerze

2.3.1 CGI (ang. Common Gateway Interface)

Zazwyczaj program serwera WWW wysyła do przeglądarki statyczne dokumenty HTML. Za pomocą programów CGI można dynamicznie (na żądanie klienta) generować dokumenty HTML uzupełniając je np. treścią pobieraną z bazy danych przed wysłaniem ich do przeglądarki.

Programy CGI są często pisane w językach interpretowalnych takich jak Perl, przez co nazywa się je także skryptami CGI.

Przykład (Linux):

```
#!/usr/bin/perl -w
print "Hello, World!\n";
```

Uwaga w przypadku Windows:

Źródło: http://www.poradnik-webmastera.com/artykuly/cgi_perl/jak_uruchomic_skrypty CGI napisane w perl_u_pod_windows.php

Serwer nie potrafił odnaleźć interpretera Perla. Szukał go tam gdzie wskazywała pierwsza linia skryptu CGI, czyli w `/usr/bin/perl`. Domyślnie takiej ścieżki nie ma pod Windows

Pierwsze rozwiązanie:

Zamiana ścieżki Unixowej na Windowsową - zamiast `/usr/bin/perl` ścieżka do katalogu gdzie na Windowsie jest zainstalowany Perl. Rozwiązanie okazało się skuteczne - skrypt się uruchomił.

Na szczęście w systemie Windows można też używać w ścieżkach do plików slash ("`/`"). Możliwe jest korzystanie ze ścieżek zaczynających się od znaku slash - ścieżka taka odnosi się do katalogu głównego bieżącego dysku twardego. W przypadku skryptów CGI uruchamianych na serwerze Apache dyskiem bieżącym jest ten na którym jest zainstalowany ten serwer.

Rozwiązaniem problemu było zatem utworzenie na dysku gdzie jest zainstalowany serwer katalogów `/usr/bin` i skopiowanie tam pliku `perl.exe` z katalogu ActiveState Perl'a. Po wykonaniu tej prostej czynności można korzystać ze skryptów CGI napisanych w Perlu także pod Windows, bez konieczności ich modyfikacji.

Przykład (Windows):

```
#!C:\perl\bin\perl.exe -wT
use strict;
use CGI;

my $query = new CGI;

print $query->header( "text/html" );

print <<END_HERE;
<html>
  <head>
    <title>My First CGI Script</title>
  </head>
  <body bgcolor="#FFFFCC">
    <h1>This is a pretty lame Web page</h1>
```

```
<p>Who is this Ovid guy, anyway?</p>
</body>
</html>
END_HERE
```

2.3.2 PERL

Interpretowany język programowania autorstwa Larry'ego Walla początkowo przeznaczony głównie do pracy z danymi tekstowymi (przeszukiwanie i wydobywanie informacji z plików tekstowych a następnie generowanie raportów na podstawie zebranych danych), obecnie używany do wielu innych zastosowań. Wzorowany na takich językach jak C, skrypcy: sed, awk i sh oraz na wielu innych.

Perl to wolne oprogramowanie, dostępne pod licencjami GPL i artystyczną. Jest dostępny dla wielu systemów operacyjnych, lecz jego naturalne środowisko to Unix i jego pochodne. Rośnie popularność Perla w środowisku Microsoft Windows. Jako przykład praktycznego zastosowania Perla można podać oprogramowanie Wikipedii, które aż do stycznia 2002 było skrypcem CGI napisanym w Perlu.

Dużą zaletą Perla jest jego niezależność od platformy, na której tworzone skrypty są wykonywane. Mimo nieznaczących różnic, przeniesienie kodu z jednej platformy na inną odbywa się bez jakichkolwiek modyfikacji - bez konieczności mozolnego przystosowywania do nowego systemu operacyjnego czy też rekompilacji kodu źródłowego - wystarczy jedynie na obu środowiskach posiadać zgodną wersję interpretera i to załatwia wszystko.

Oprócz programów wykonywanych "wiersz po wierszu" pozwala on konstruować skomplikowane struktury danych i programować w stylu podobnym do obiektowego. Dzięki elastyczności, każde zadanie można wykonać na więcej niż jeden sposób. Powoduje to, iż programista może opracować swój własny koncept myślenia i własny styl programowania. Daje to swobodę działania i możliwość wybrania drogi, jaka najlepiej pasuje do wykonania zadania.

Aby móc na swoim komputerze uruchamiać skrypty Perla, trzeba mieć go zainstalowanego. Ponieważ jest on dostępny za darmo, pobranie go z Sieci i instalacja nie stanowi żadnego problemu.

Przykład:

```
$_ = "Magiczna liczba to 150\n";
print;
s/\d+/-1/;
print;
```

3 wiersz to wyrażenie regularne.

Wynik

```
Magiczna liczba to 150
Magiczna liczba to -1
```

Tłumaczenie na polski:

```
Weź napis "Magiczna liczba to 150\n";
Wydrukuj go;
Zmień w nim pierwszy ciąg cyfr na "-1";
Wydrukuj go;
```

2.3.3 JSP (ang. JavaServer Pages)

Technologia umożliwiająca tworzenie dynamicznych dokumentów WWW w formatach HTML, XHTML, DHTML oraz XML z wykorzystaniem języka Java, wpleczonego w kod HTML danej strony. W tym aspekcie, jest to rozwiązanie podobne do PHP.



Strona JSP w procesie translacji jest zamieniana na *serwlet* (z reguły mała aplikacja napisana w Javie uruchamiana po stronie serwera w kontenerze serwetów). Każde wywołanie strony JSP z poziomu klienta (przeglądarki) wykonywane jest przez skompilowany serwlet

Serwlet – klasa Javy działająca po stronie serwera WWW w modelu żądanie-odpowiedź, rozszerzająca jego możliwości. Nazwa Serwlet powstała na wzór nazwy aplet, przez zastąpienie sylaby ap- sylabą serw-, wskazującą na wykonywanie programu po stronie serwera.

Elementy strony JSP

- treść statyczna - przepisywana bez modyfikacji do generowanego dokumentu
- dyrektywy JSP - informacje kontrolujące proces generowania dokumentu
- elementy skryptowe - skryplety (kod w języku Java kontrolujący proces generowania dokumentu) oraz elementy składniowe tzw. Expression Language
- akcje JSP - tagi XML wywołujące określone metody serwerowe

Przykładowa strona JSP

Dyrektywa ustawiająca typ zawartości, kodowanie itp	<code><%@ page language="java" contentType="text/html; charset=ISO-8859-2" pageEncoding="ISO-8859-2"%></code>
Czysty kod HTML	<code><!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>
skryplet deklarujący zmienną języka Java	<code><%! int k=5; %></code>
Czysty kod HTML	<code><html> <head> <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-2"/> <title>Przykładowa strona JSP</title> </head> <body></code>
Skryplet zwracający wartość zmiennej języka Java	Aktualny czas: <code><%=java.util.Calendar.getInstance().getTime()%></code>
Skryplety sterujące generowaniem dokumentu	<code><% for (int i=0; i<k; ++i) { %> Liczba: <%=i%>
</code> <code><% } %></code>
Czysty kod HTML	<code></body> </html></code>

2.3.4 ASP (ang. Active Server Pages)



Technika firmy Microsoft służąca tworzeniu dynamicznych stron WWW wykonywanych po stronie serwera, oparta na jednym z dostępnych języków skryptowych.

Dokumenty ASP składają się z dwóch elementów:

- kodu programu
- kodu HTML lub XHTML

Kod programu może być napisany w jednym z akceptowanych przez ASP języków skryptowych (domyślnie wspierane oznaczone czcionką wytłuszczoną) **VBScript**, **C#**, Jscript, PerlScript, Python, Ruby, Delphi

Szereg dodatkowych można uzyskać przez doinstalowanie niezależnych silników skryptowych dostarczanych w postaci rozszerzeń Active Scripting.

Większość stron ASP jest tworzona przy wykorzystaniu języka VBScript, skutecznie promowanego przez Microsoft jako najlepszego do tego celu. Strony ASP mają domyślnie skojarzone rozszerzenie .asp.

2.3.5 ASP.NET



Jest to produkt który ma być odpowiedzią firmy Microsoft na sukces Macromedia/Adobe. Microsoft prowadził od początku politykę polegającą na unikaniu standaryzacji celem uzyskania przewagi konkurencyjnej w postaci wyłącznego dostępu do pełnej informacji o produktach. Nawet jeśli jakieś standardy decydował się stosować, to modyfikował je w swych implementacjach tak by przestały one być zgodne z innymi implementacjami. To doprowadziło do konfliktu o standard Java. Microsoft zmuszony wyrokiem sądowym do stosowania takich praktyk "pogniewał się" i stworzył środowisko .NET. Mimo iż firma przyjęła do wiadomości to iż istnieją inne systemy poza Windows, to wsparcie dla konkurencyjnego Linux'a jest udzielane 'po cichu'. Powstał otwarty projekt Mono.

Następca ASP. Technika tworzenia dynamicznych stron internetowych po stronie serwera, działająca w oparciu o technikę .NET Framework i kod zarządzany.

Głównym celem było umożliwienie jak najszybszego i prostego tworzenia rozbudowanych serwisów internetowych.

Strony ASP.NET są uruchamiane przy użyciu serwera, który umożliwia wygenerowanie treści HTML(CSS), WML lub XML – rozpoznawanych oraz interpretowanych przez przeglądarki internetowe.

Aplikacje internetowe są pisane w języku C#. Uruchamiane są jako obiekty Silverlight (Microsoft dostarcza aplikację odtwarzającą je na podobieństwo Flash Playera). W systemie Linux istnieje odpowiednik tego odtwarzacza pod nazwą Moonlight.

Przykład:

```
<!DOCTYPE html>
<html>
<body>
  <h1>Hello Web Pages</h1>
  <p>The time is @DateTime.Now</p>
</body>
</html>
```

2.3.6 PHP



Skryptowy obiektowy język programowania zaprojektowany do generowania stron internetowych w czasie rzeczywistym. Jak działa interpreter PHP? To proste - przerabia instrukcje PHP na HTML.

PHP jest projektem open-source. Każdy może pobrać za darmo jego kopię, zainstalować i używać bez żadnych ograniczeń. Do kodu źródłowego zapewniony jest pełny dostęp - jeżeli programista ma odpowiednie zdolności, może go modyfikować do woli oraz nadsyłać własne propozycje zmian do osób nadzorujących projekt. Dzięki takiej wolności PHP rozwija się bardzo dynamicznie, a w Internecie można znaleźć setki modyfikacji oraz dodatkowych modułów.

Mimo otwartości kodu, nad rozwojem oficjalnej wersji projektu czuwa firma Zend Company z Izraela założona przez twórców PHP. Zapewnia ona dodatkowe narzędzia i opiekę prawną, a także wyznacza kierunki rozwoju projektu.

Zastosowanie

PHP znaleźć można wszędzie: od prywatnych stron domowych przez gry internetowe aż do potężnych witryn korporacji lub portali. Opanowanie języka jest proste także dlatego, iż Internet pełen jest przewodników oraz artykułów pokazujących, jak w praktyce wykorzystywać wiele z jego możliwości.

PHP jest najczęściej stosowany do tworzenia skryptów po stronie serwera WWW, ale może być on również używany do przetwarzania danych z poziomu wiersza poleceń, a nawet do pisania programów pracujących w trybie graficznym (np. za pomocą biblioteki GTK+, używając rozszerzenia PHP-GTK). Idealnie nadaje się do tworzenia złożonych aplikacji zarządzających dużymi ilościami danych: forami dyskusyjnymi, systemami zarządzania treścią (CMS), sklepami internetowymi (e-Commerce).

Implementacja PHP wraz z serwerem WWW Apache oraz serwerem baz danych MySQL, określana jest jako platforma AMP (w środowisku Linux – LAMP, w Windows – WAMP).

Ponadto PHP można uruchomić na serwerze IIS (Windows) oraz w chmurze Windows Azure.

Składnia PHP jest oparta na składni języków C, Java oraz Perl. Cały kod PHP musi zawierać się np. pomiędzy znacznikami: `<?php` oraz `?>`

Przykład:

```
<?php
$a = 5;
$b = 3;

function suma($zmienna1, $zmienna2)
{
    $wynik = $zmienna1 + $zmienna2;
    return $wynik; //Dodajemy argumenty i zwracamy wynik
}

echo suma($a, $b); // 8
?>
```

CMS

System zarządzania treścią (ang. *Content Management System*, CMS) – oprogramowanie pozwalające na łatwe utworzenie serwisu WWW oraz jego późniejszą aktualizację i rozbudowę przez redakcyjny personel nietechniczny.

Kształtowanie treści i sposobu ich prezentacji w serwisie internetowym zarządzanym przez CMS odbywa się za pomocą prostych w obsłudze interfejsów użytkownika, zazwyczaj w postaci stron WWW zawierających rozbudowane formularze i moduły.

Podstawowym zadaniem platform CMS jest oddzielenie treści (zawartości informacyjnej serwisu) od wyglądu (sposobu jej prezentacji). Po wprowadzeniu nowych informacji przez uprawnionego redaktora system zapisuje je w bazie danych. Następnie system CMS generuje dynamicznie strony internetowe na podstawie treści pochodzącej z bazy danych oraz odpowiednich szablonów. Pozwala to na bardziej elastyczne a przede wszystkim wygodniejsze zarządzanie treścią niż ma to miejsce w przypadku zastosowania statycznych plików HTML.

Wykorzystanie szablonów stron sprawia, że zmiana koncepcji graficznej całego serwisu sprowadza się do przygotowania i zamiany szablonu (tzw. skórki). Dzięki takiemu podejściu proces publikacji w Internecie staje się prostszy. Ponadto te same dane można prezentować jednocześnie w różnych formatach, np. jako pliki PDF.

W ostatnich latach platformy CMS znacznie ewoluowały, dodając m.in. możliwość elastycznej i dynamicznej modyfikacji już nie tylko treści, ale i struktury.

Systemy zarządzania treścią zazwyczaj oparte są na bazach danych i językach skryptowych (server-side), lub specjalistycznym oprogramowaniu uruchamianym po stronie serwera. CMS może być formą oprogramowania klienckiego, które do aktualizacji strony może używać takich protokołów jak FTP. Coraz częściej wykorzystywane są złożone techniki opisu struktur dokumentów (np. XML).

Najczęściej systemy CMS są napisane w językach takich jak: ASP, JSP, PHP, czy Python.

Przykłady CMS:

- Joomla
- WordPress
- Quick.CMS
- itp.

2.3.7 Ruby on Rails

Ruby on Rails (często nazywany RoR lub po prostu Rails – rubiny i szmaragdy) – framework open source do szybkiego tworzenia aplikacji webowych stworzony głównie przez duńskiego programistę Davida Heinemeiera Hanssona w ramach pracy nad oprogramowaniem Basecamp. RoR został napisany w języku Ruby z użyciem architektury MVC (ang. Model-View-Controller).



Rails znajduje zastosowanie jako narzędzie tworzenia praktycznie każdego rodzaju aplikacji internetowej, czy będzie to oprogramowanie do współpracy online, tworzenia społeczności, e-commerce, zarządzania treścią czy statystyk.

Popularność języków na podstawie Google Trends

