



Zaawansowane Techniki WWW (HTML, CSS i JavaScript)

Dr inż. Marcin Zieliński

Środa 15:30 - 17:00 sala: A-1-04

WYKŁAD 2

Wykład dla kierunku: **Informatyka Stosowana II rok**

Rok akademicki: **2015/2016 - semestr zimowy**

Przypomnienie

Wprowadzenie do technologii internetowych

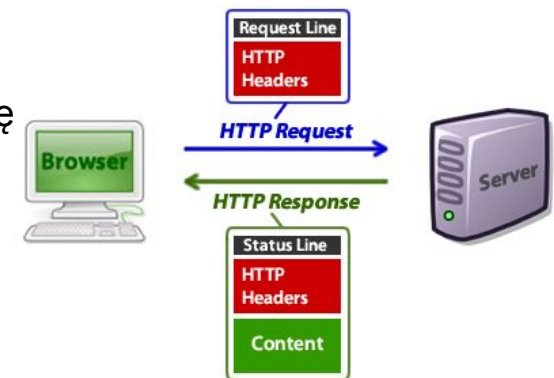
Historia internetu stron WWW

Protokoły komunikacyjne HTTP/HTTPS

Protokół HTTP

Hyper Text Transfer Protocol (HTTP/1.1) [Protokół przesyłania dokumentów Hiper-Tekstowych]

- W ogólności protokół zapewnia komunikację pomiędzy usługodawcą (serwerem) a klientem (hostem).
- W protokole zawarte są metody (*funkcje*) umożliwiające obsługę żądań (*zapytań*) wysyłanych przez klienta oraz odpowiedzi serwera.
- Cechą charakterystyczną protokołu HTTP jest jego “bezstanowość” (*ang. stateless*), co oznacza że nigdzie nie istnieje zapis stanu poprzednio wykonanych operacji, a kolejne transakcje są wykonywane niezależnie.
- Protokół HTTP operuje standardowo na porcie 80.



Przykład:

Kliknięcie na *hiperpołączenie* (np. w przeglądarce) powoduje, że komputer-klient (host) wysyła do serwera *żądanie*, po otrzymaniu żądania i po przeszukaniu swoich zasobów, wysyła do klienta – jeśli to możliwe – stosowną stronę WWW (lub w ogólności dowolne zasoby).

Definicja i opis standardu HTTP/1.1

(opis standardu HTTP/2.0 - w przygotowaniu)

RFC 2616
(czerwiec 1999)

RFC (Request for Comments)

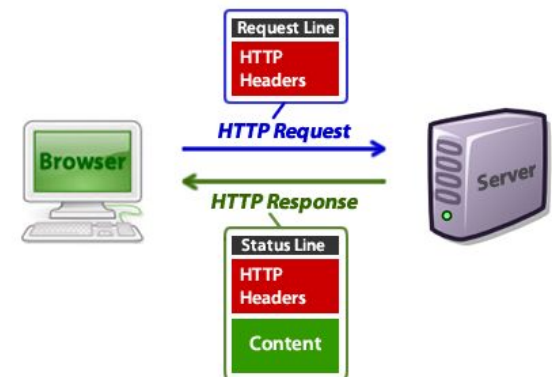
Protokół HTTP

Hyper Text Transfer Protocol (HTTP/1.1) [Protokół przesyłania dokumentów Hiper-Tekstowych]

Najważniejsze metody protokołu HTTP:

- **HEAD** - wysyła żądanie przesłania nagłówka zawierającego meta-dane (informację), bez przesyłania samego zasobu.
- **GET** - wysyła żądanie pobrania konkretnego zasobu URI (np. strony internetowej napisanej w języku HTML).
- **POST** - wysyła żądanie do serwera akceptacji zasobu dołączonego do żądania.

pozostałe: **PUT, DELETE, TRACE, OPTIONS, CONNECT.**



Przykład 1:

Żądanie (klient)

GET HTTP/1.1

Odpowiedź (serwer)

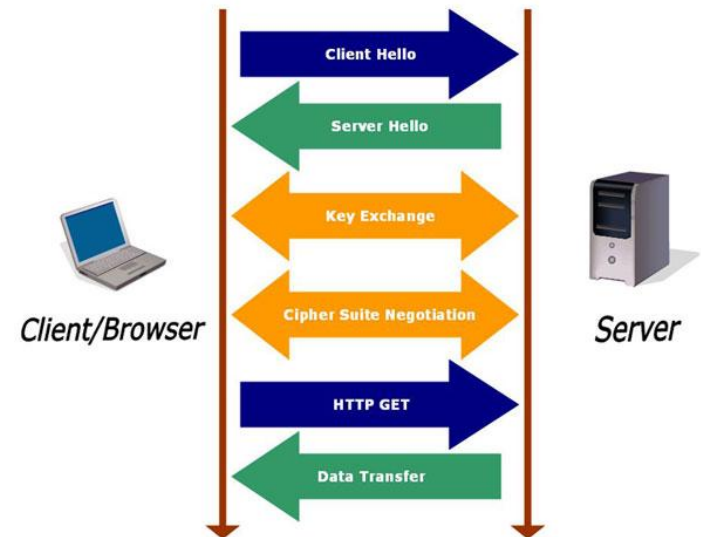
HTTP/1.1 200 OK

Kod odpowiedzi (stanu)

Protokół HTTPS

Hyper Text Transfer Protocol Secure (HTTPS) [Zabezpieczony Protokół przesyłania dokumentów Hiper-Tekstowych]

- Jest to wersja szyfrowana protokołu HTTP, oparta o protokół szyfrujący SSL (Secure Socket Layer).
- SSL działa warstwę niżej w modelu OSI od samego HTTPS.
- Działanie protokołu HTTPS polega na tym samym co HTTP, natomiast zanim zostanie ustanowiona komunikacja “klient-serwer”, wymieniane są klucze szyfrujące SSL.
- Wymiana danych na bazie protokołu HTTPS, zapobiega przechwyceniu oraz zmianie danych w trakcie komunikacji klient-serwer.
- Domyślnym portem na których serwer nasłuchuje żądań HTTPS to 443.



<https://www.extranet.uj.edu.pl>

Protokół HTTP i HTTPS

- Do sprawnego posługiwania się protokołem HTTP/HTTPS konieczna jest znajomość podstawowych kodów oznaczających stany serwera zwracana w odpowiedzi na żądanie klienta.
- Kod stanu (odpowiedź) serwera jest podawana w nagłówku HTTP/HTTPS w postaci liczby trzycyfrowej.

GET HTTP/1.1

HTTP/1.1 200 OK

Klasyfikacja kodów stanów (odpowiedzi serwera):

Zakres kodów	Znaczenie
100 - 199	Informacyjne.
200 - 299	Żądanie (od klienta) powiodło się
300 - 399	Żądanie klienta zostało przekazane, wymagane są dalsze działania.
400 - 499	Żądanie klienta nie powiodło się.
500 - 599	Błąd serwera.

Serwery obsługujące protokoły HTTP/HTTPS (Web-Serwery)

WebServer - komputer (serwer) - a najczęściej klasa komputerowy - obsługujący żądania HTTP/HTTPS, za pomocą odpowiedniego oprogramowania.

Główną a jednocześnie podstawową funkcjonalnością WebServerów jest przechowywanie, przetwarzanie i dostarczanie stron internetowych zapisanych za pomocą języka HTML, oraz obrazów, multimediów i skryptów, o które dodatkowo wzbogacone są strony internetowe.

Serwery obsługujące protokół HTTP/HTTPS (Web-Serwery)

WebServer - komputer (serwer), a najczęściej klasa komputerowa, obsługująca żądania HTTP/HTTPS, za pomocą odpowiedniego oprogramowania.

Główną a jednocześnie podstawową funkcjonalnością WebServerów jest przechowywanie, przetwarzanie i dostarczanie stron internetowych zapisanych za pomocą języka HTML, oraz obrazów, multimediiów i skryptów, o które dodatkowo wzbogacone są strony internetowe.

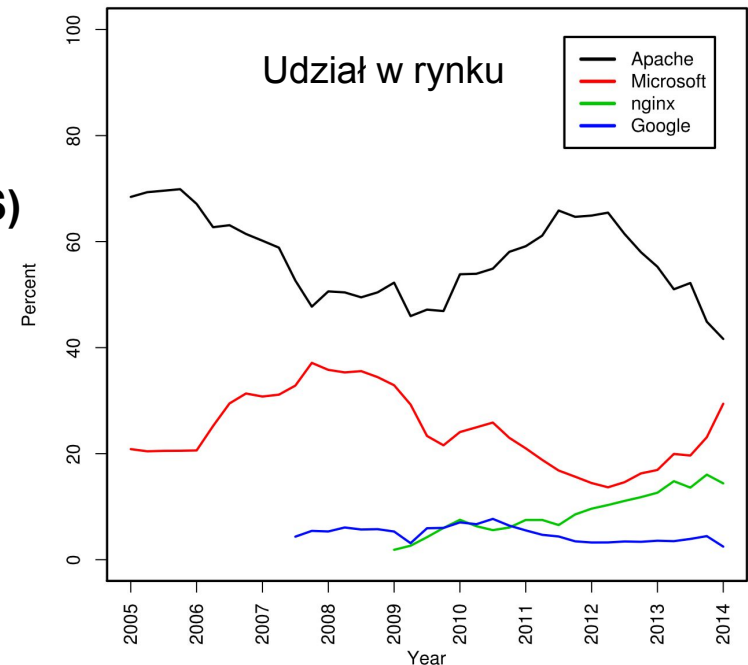
Najpopularniejsze oprogramowanie WebServerowe:

- **APACHE**
- **Microsoft Internet Information Services (IIS)**
- **Nginx**
- **Google Web Server (GWS)**



NGINX

Google



(dane Netcraft, maj 2014)

Serwery obsługujące protokół HTTP/HTTPS (Web-Serwery)

WebServer - komputer (serwer), a najczęściej klasa komputerowa, obsługująca żądania HTTP/HTTPS, za pomocą odpowiedniego oprogramowania.

Główną a jednocześnie podstawową funkcjonalnością WebServerów jest przechowywanie, przetwarzanie i dostarczanie stron internetowych zapisanych za pomocą języka HTML, oraz obrazów, multimediiów i skryptów, o które dodatkowo wzbogacone są strony internetowe.

Najpopularniejsze oprogramowanie WebServerowe:

- **APACHE**
- **Microsoft Internet Information Services (IIS)**
- **Nginx**
- **Google Web Server (GWS)**



NGINX

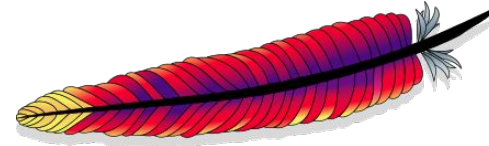
Google



WebServer	Udział w rynku
Apache	38%
MS IIS	33%
Nginx	15%
GWS	2%

(dane Netcraft, maj 2014)

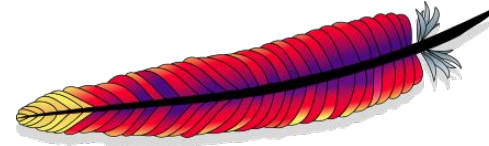
Serwer APACHE



Apache HTTP Server (Apache Software Foundation) - początki serwera sięgają roku 1995 kiedy w National Center for Supercomputing Applications Robert McCool stworzył aplikację NCSA Httpd.

- Serwer posiada wiele funkcjonalności zaimplementowanych w postaci modułów które rozszerzają jego podstawowe funkcje.
- Zapewnia wsparcie dla wielu języków skryptowych które mogą być wykorzystane przy tworzeniu aplikacji internetowych m.in. : PHP, Perl, Python, TCL.
- Współpracuje z najpopularniejszymi silnikami baz danych: MySQL, PostgreSQL.
- Apache obsługuje protokół HTTP jak i jego szyfrowaną wersję HTTPS (SSL).
- Jedną z zalet Apache jest moduł “*mod_gzip*” umożliwiający kompresję danych przesyłanych przez protokół HTTP, co znacząco wpływa na prędkość podawania danych.
- W ramach jednego zainstalowanego serwera APACHE, dzięki tzw. “wirtualnym hostom” można obsługiwać wiele stron internetowych alokowanych w ramach różnych domen.
- Jest dostępny bezpłatnie na większość systemów operacyjnych, w szczególności jest dostarczony domyślnie z wieloma dystrybucjami systemu LINUX/UNIX.

Serwer APACHE



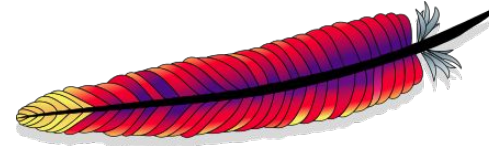
Apache HTTP Server (Apache Software Foundation) - początki serwera sięgają roku 1995 kiedy w National Center for Supercomputing Applications Robert McCool stworzył aplikację NCSA Httpd.

- Serwer posiada wiele funkcjonalności zaimplementowanych w postaci modułów które rozszerzają jego podstawowe funkcje.
- Zapewnia wsparcie dla wielu języków skryptowych które mogą być wykorzystane przy tworzeniu aplikacji internetowych m.in. : PHP, Perl, Python, TCL.
- Współpracuje z najpopularniejszymi silnikami baz danych: MySQL, PostgreSQL.
- Apache obsługuje protokół HTTP jak i jego szyfrowaną wersję HTTPS (SSL).
- Jedną z zalet Apache jest moduł “*mod_gzip*” umożliwiający kompresję danych przesyłanych przez protokół HTTP, co znacząco wpływa na prędkość podawania danych.
- W ramach jednego zainstalowanego serwera APACHE, dzięki tzw. “wirtualnym hostom” można obsługiwać wiele stron internetowych alokowanych w ramach różnych domen.
- Jest dostępny bezpłatnie na większość systemów operacyjnych, w szczególności jest dostarczony domyślnie z wieloma dystrybucjami systemu LINUX/UNIX.

Uwaga:

Na liście procesów (np. w systemie Linux) pracujący serwer Apache oznaczony jest nazwą **httpd (http-daemon)**

Serwer APACHE



Apache HTTP Server (Apache Software Foundation) - początki serwera sięgają roku 1995 kiedy w National Center for Supercomputing Applications Robert McCool stworzył aplikację NCSA Httpd.

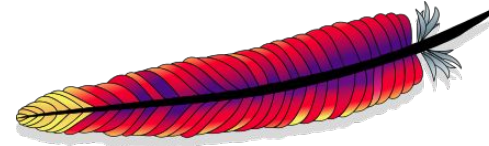
- Serwer posiada wiele funkcjonalności zaimplementowanych w postaci modułów które rozszerzają jego podstawowe funkcje.
- Zapewnia wsparcie dla wielu języków skryptowych które mogą być wykorzystane przy tworzeniu aplikacji internetowych m.in. : PHP, Perl, Python, TCL.
- Współpracuje z najpopularniejszymi silnikami baz danych: MySQL, PostgreSQL.
- Apache obsługuje protokół HTTP jak i jego szyfrowaną wersję HTTPS (SSL).
- Jedną z zalet Apache jest moduł "*mod_gzip*" umożliwiający kompresję danych przesyłanych przez protokół HTTP, co znacząco wpływa na prędkość podawania danych.
- W ramach jednego zainstalowanego serwera APACHE, dzięki tzw. "wirtualnym hostom" można obsługiwać wiele stron internetowych alokowanych w ramach różnych domen.
- Jest dostępny bezpłatnie na większość systemów operacyjnych, w szczególności jest dostarczony domyślnie z wieloma dystrybucjami systemu LINUX/UNIX.

Uwaga:

Na liście procesów (np. w systemie Linux) pracujący serwer Apache oznaczony jest nazwą **httpd (http-daemon)**

Najnowsza wersja: **2.4.10** do pobrania z **<https://httpd.apache.org/>**

Serwer APACHE



Apache HTTP Server (Apache Software Foundation) - początki serwera sięgają roku 1995 kiedy w National Center for Supercomputing Applications Robert McCool stworzył aplikację NCSA Httpd.

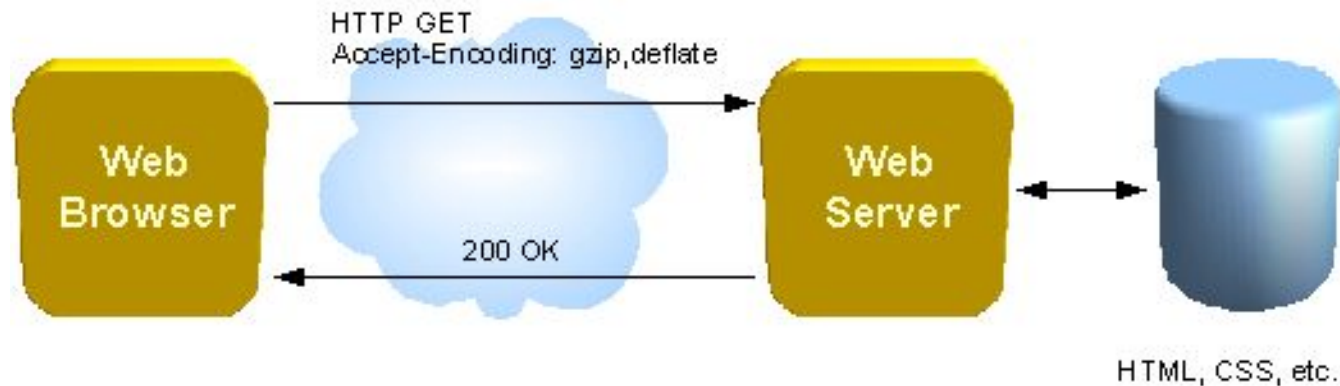
- Serwer posiada wiele funkcjonalności zaimplementowanych w postaci modułów które rozszerzają jego podstawowe funkcje.
- Zapewnia wsparcie dla wielu języków skryptowych które mogą być wykorzystane przy tworzeniu aplikacji internetowych m.in. : PHP, Perl, Python, TCL.
- Współpracuje z najpopularniejszymi silnikami baz danych: MySQL, PostgreSQL.
- Apache obsługuje protokół HTTP jak i jego szyfrowaną wersję HTTPS (SSL).
- Jedną z zalet Apache jest moduł "*mod_gzip*" umożliwiający kompresję danych przesyłanych przez protokół HTTP, co znacząco wpływa na prędkość podawania danych.
- W ramach jednego zainstalowanego serwera APACHE, dzięki tzw. "wirtualnym hostom" można obsługiwać wiele stron internetowych alokowanych w ramach różnych domen.
- Jest dostępny bezpłatnie na większość systemów operacyjnych, w szczególności jest dostarczony domyślnie z wieloma dystrybucjami systemu LINUX/UNIX.

Uwaga:

Dla wygody użytkowników którzy chcą uruchomić w pełni działający serwer WWW na własnym laptopie polecam rozwiązania integrujące w jednej aplikacji serwer **Apache**, **MySQL**, **PHP**. Jednym z tego typu rozwiązań jest darmowa wersja aplikacji **X-AMP** dostępną na większość popularnych systemów operacyjnych:

<https://www.apachefriends.org/pl/index.html>

Synchroniczne żądanie http

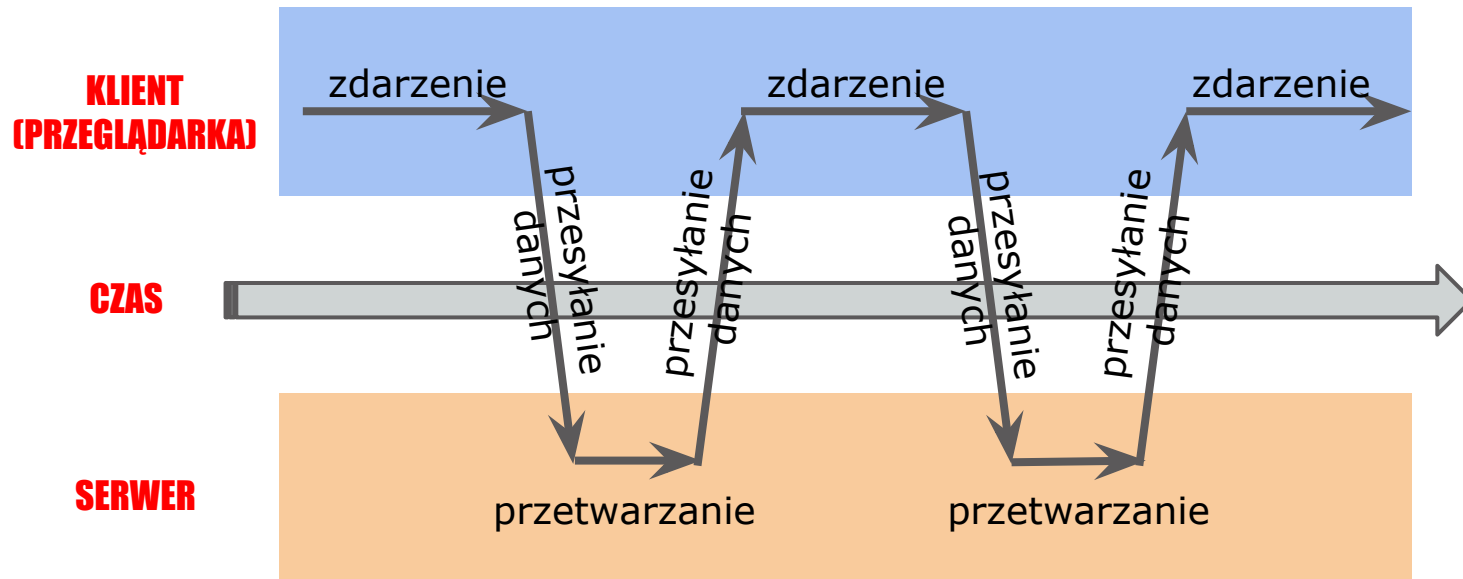


Jest to model synchronicznej komunikacji HTTP, gdzie klient wysyła żądanie, serwer je odbiera następnie przetwarza i na końcu generuje odpowiedź.

W sytuacji takiej klient musi czekać z kolejnym żądaniem do momentu kiedy nie dostanie odpowiedzi od serwera.

W modelu synchronicznym mamy bardzo mały poziom aktywności oraz interaktywności strony, strona musi być przeładowana (pobrana) po każdym żądaniu klienta, jeśli strony są złożone to proces ten jest długi.

Synchroniczne żądanie http

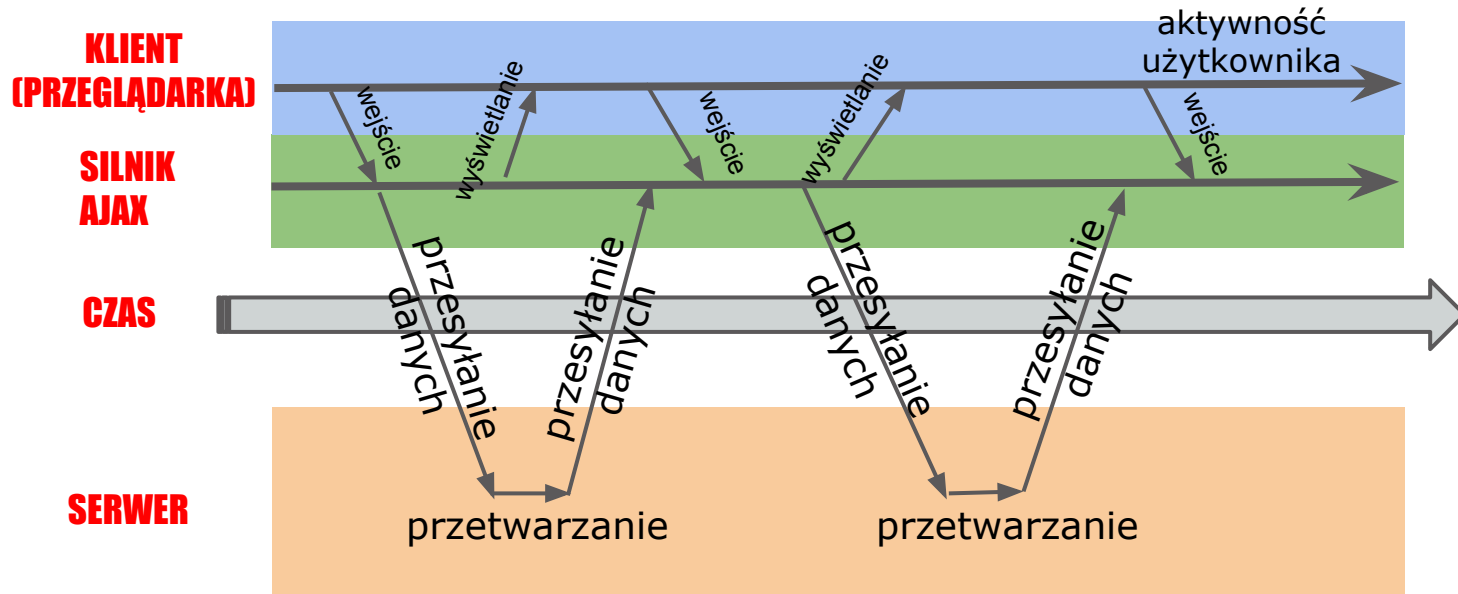


Jest to model synchronicznej komunikacji HTTP, gdzie klient wysyła żądanie, serwer je odbiera następnie przetwarza i na końcu generuje odpowiedź.

W sytuacji takiej klient musi czekać z kolejnym żądaniem do momentu kiedy nie dostanie odpowiedzi od serwera.

W modelu synchronicznym mamy bardzo mały poziom aktywności oraz interaktywności strony, strona musi być przeładowana (pobrana) po każdym żądaniu klienta, jeśli strony są złożone to proces ten jest długi.

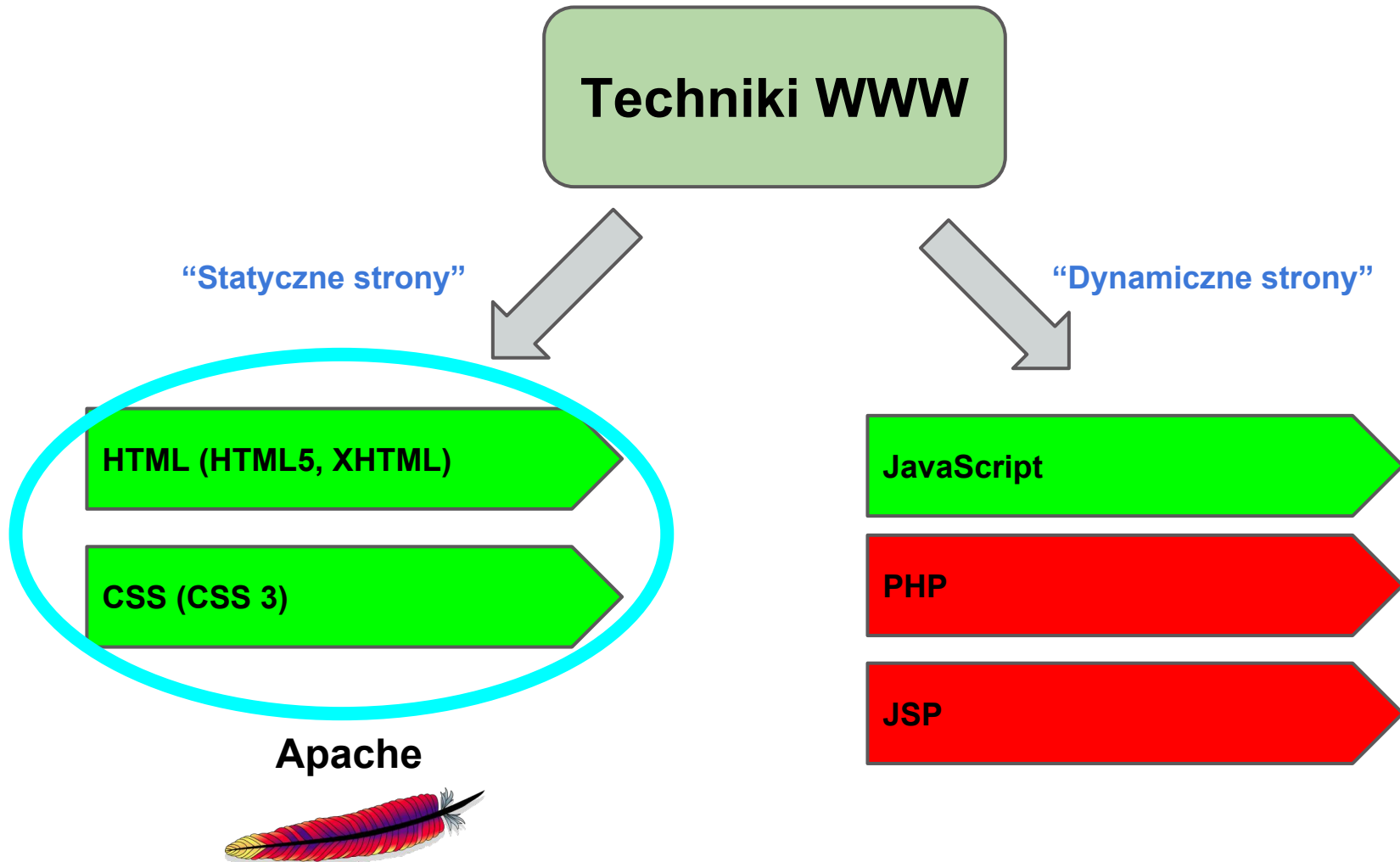
Asynchroniczne żądanie http



Jest to model asynchronicznej komunikacji HTTP, gdzie klient (przeglądarka) nie czeka na przyjęcie odpowiedzi na żądanie serwera, a wykonuje dalsze żądania.

W takim modelu nie ma konieczności przeładowania strony przy każdej operacji klienta, wystarczy, że zostaną doczytane brakujące dane, a dzięki odpowiednim narzędziom zmodyfikowana zostanie zawartość strony.

Technologie wykorzystywane na kursie

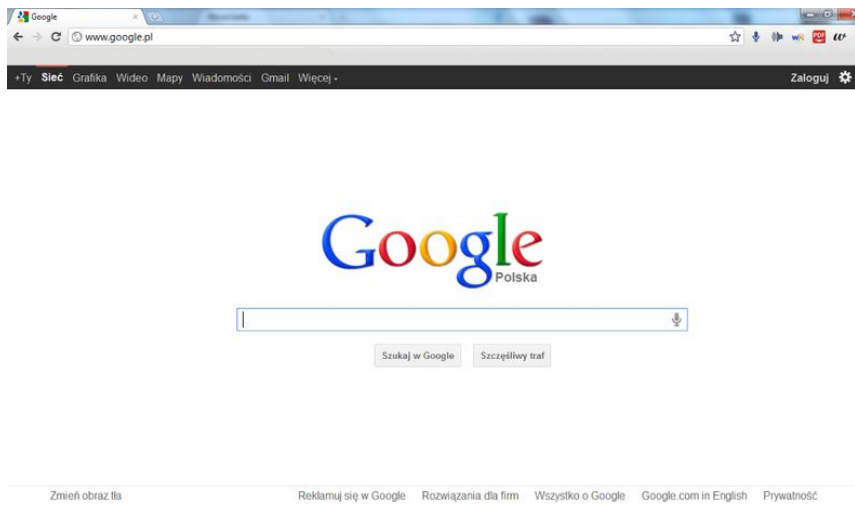


Strona internetowa WWW

Strona Internetowa WWW - kolekcja logicznie połączonych ze sobą zasobów (najczęściej plików napisanych w języku **HTML** oraz multimediiów), znajdujących się na jednym serwerze (w konkretnej kartotece) obsługującym żądania HTTP.

Strona internetowa WWW

Strona Internetowa WWW - kolekcja logicznie połączonych ze sobą zasobów (najczęściej plików napisanych w języku **HTML** oraz multimediów), znajdujących się na jednym serwerze (w konkretnej kartotece) obsługującym żądania HTTP.



<http://www.google.pl>



<http://www.onet.pl>

- Zasoby w ramach jednej strony internetowej są ze sobą połączone za pomocą tzw. hiperłączy które wskazują na URI konkretnego zasobu.
- Pojęcie "*strony internetowej*" (inaczej witryny internetowej) łączy w sobie: dane (treść strony), prezentację (formatowanie) oraz logikę (strukturę).

Strona internetowa WWW

Strona Internetowa WWW - kolekcja logicznie połączonych ze sobą zasobów (najczęściej plików napisanych w języku **HTML** oraz multimediów), znajdujących się w konkretnej kartotece na serwerze obsługującym żądania HTTP.

```
view-source:www.onet.pl

1 <!DOCTYPE html>
2 <html lang="pl">
3 <head>
4   <meta charset="utf-8" />
5   <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
6   <meta name="msapplication-starturl" content="http://www.onet.pl/" />
7   <meta name="msapplication-task" content="name=Wiadomości;action-uri=http://wiadomosci.onet.pl;icon-uri=http://ocdn.eu/static/template-engine/MjU7MDA_/f0d3c1307427ace61799402945edcd72/faviconn2.ico"/>
8   <meta name="msapplication-task" content="name=Biznes;action-uri=http://biznes.onet.pl;icon-uri=http://ocdn.eu/static/template-engine/MjU7MDA_/f0d3c1307427ace61799402945edcd72/faviconn1.ico"/>
9   <meta name="msapplication-task" content="name=Sport;action-uri=http://sport.onet.pl;icon-uri=http://ocdn.eu/static/template-engine/MjU7MDA_/f0d3c1307427ace61799402945edcd72/faviconn1.ico"/>
10  <meta name="msapplication-task" content="name=Film;action-uri=http://film.onet.pl;icon-uri=http://ocdn.eu/static/template-engine/MjU7MDA_/f0d3c1307427ace61799402945edcd72/faviconn1.ico"/>
11  <meta name="msapplication-task" content="name=Muzyka;action-uri=http://muzyka.onet.pl;icon-uri=http://ocdn.eu/static/template-engine/MjU7MDA_/f0d3c1307427ace61799402945edcd72/faviconn1.ico"/>
12  <meta name="msapplication-task" content="name=VOD;action-uri=http://vod.onet.pl;icon-uri=http://ocdn.eu/static/template-engine/MjU7MDA_/f0d3c1307427ace61799402945edcd72/faviconn1.ico"/>
13  <meta name="description" content="Dzisiaj w Onet.pl: wiadomości z kraju i ze świata; biznes, sport, rozrywka i pogoda. Sprawdź pocztę, bloguj, czatuj i umawiaj się na randki."/>
14  <title>Onet - informacje, rozrywka, emocje</title>
15  <link rel="canonical" href="http://www.onet.pl" />
16  <link rel="dns-prefetch" href="http://ocdn.eu" />
17  <link rel="dns-prefetch" href="http://m.ocdn.eu" />
18  <link rel="dns-prefetch" href="http://lib.onet.pl" />
19  <link rel="dns-prefetch" href="http://csr.onet.pl" />
20  <link rel="dns-prefetch" href="http://kropka.onet.pl" />
21  <link rel="dns-prefetch" href="http://google-analytics.com" />
22  <link rel="dns-prefetch" href="http://reklama.onet.pl" />
23  <link rel="shortcut icon" href="http://ocdn.eu/static/template-engine/MjU7MDA_/f0d3c1307427ace61799402945edcd72/faviconn1.ico" type="image/x-icon" />
24  <script type="text/javascript"> /*  */
25    window.suggestmeyes_loaded = true;
26  /* ]]] */&lt;/script&gt;
27
28  &lt;link href="http://ocdn.eu/resources/template-engine/mainpagev2templateengine.template-engine/resource/737d8fa66a9b72d454bbda926896b73d/11.55/nsg.css" type="text/css" rel="stylesheet"&gt;
29
30  &lt;link href="http://ocdn.eu/resources/template-engine/mainpagev2templateengine.template-engine/resource/008f95d4bad5c3563a6389dfa6f66f5/11.55/7.2/menu.css" type="text/css" rel="stylesheet"&gt;
31  &lt;!--[if lt IE 9]&gt;
32  &lt;link href="http://ocdn.eu/resources/template-engine/mainpagev2templateengine.template-engine/resource/737d8fa66a9b72d454bbda926896b73d/11.55/ieimg.css" type="text/css" rel="stylesheet"&gt; &lt;![endif]--&gt;
33  &lt;script src="http://ocdn.eu/resources/template-engine/mainpagev2templateengine.template-engine/resource/none/11.55/nsg-top_min.js" type="text/javascript"&gt;&lt;/script&gt;
34  &lt;script type="text/javascript"&gt;
35    if(NSGStream.requireJS){
36      document.write(&lt;link href="http://ocdn.eu/resources/template-engine/mainpagev2templateengine.template-engine/resource/none/11.55/nsg-stream.css" type="text/css" rel="stylesheet"&gt;);
37      document.write(&lt;script src="http://ocdn.eu/resources/template-engine/mainpagev2templateengine.template-engine/resource/none/11.55/nsg-stream.min.js"&gt;&lt;/script&gt;);
38    }
39  &lt;/script&gt;
40  &lt;!--[if lt IE 9]&gt;
41  &lt;script src="http://ocdn.eu/static/template-engine/MjU7MDA_/f0d3c1307427ace61799402945edcd72/ie-lt9.js" type="text/javascript"&gt;&lt;/script&gt;
42  &lt;![endif]--&gt;
43  &lt;script type="text/javascript"&gt; /* <![CDATA[ */
44    NSG_AUTOREFRESH = (typeof NSG_AUTOREFRESH == 'undefined' ? false : NSG_AUTOREFRESH);
45    if (NSG_AUTOREFRESH != true) {
46      pp_gemius_identifier = 'bPo6D0bzSxcue3osfkZZIJaE.10Rye0gSEhsufRyys3.W7';
47    }
48  /* ]]] */&lt;/script&gt;
49  &lt;script type="text/javascript" src="http://ocdn.eu/static/mast/xgemius.js"&gt;&lt;/script&gt;
50  &lt;script type="text/javascript"&gt; /* <![CDATA[ */
51    NSG_KEYWORDS = (typeof NSG_KEYWORDS == 'undefined' ? '' : NSG_KEYWORDS);
52    NSG_LOCAL = (typeof NSG_LOCAL == 'undefined' ? '' : NSG_LOCAL);
53    var adSlots = ['flat-search','flat-boks1','flat-link1','flat-link2','flat-link3','flat-link5','flat-link4','flat-link6','flat-link11','flat-boxright1','flat-boxright2','flat-boxright3','flat-boxright4'];
54    if(!promoTop &amp;&amp; !NSGStream.hashStream) {
55      adSlots.unshift('right');
56      adSlots.unshift('top');
57    }
58    if(NSGStream.hashStream) {
59      adSlots.unshift('flat-boxright5'); adSlots.unshift('flat-boxright6'); adSlots.unshift('flat-boxright9'); adSlots.unshift('flat-boxright10');
60      adSlots.unshift('flat-link7'); adSlots.unshift('flat-link8'); adSlots.unshift('flat-link12'); adSlots.unshift('flat-link13'); adSlots.unshift('flat-sidebarbox');
61    }
62    onetAds={</pre></div>
```

Technologie teraz i dawniej

Strony internetowe dawniej i obecnie są tworzone z wykorzystaniem języka

HTML (Hyper Text Markup Language)

[początki sięgają 1980 roku]

Z biegiem czasu sam język znaczników HTML przestał spełniać oczekiwania co do tworzenia serwisów WWW i w trakcie ostatnich 20 lat wokół technologii “webowych”, powstały różne narzędzia pozwalające na uatrakcyjnienie wyglądu stron internetowych.

Technologie teraz i dawniej

Strony internetowe dawniej i obecnie są tworzone z wykorzystaniem języka

HTML (Hyper Text Markup Language)

[początki sięgają 1980 roku]

Z biegiem czasu sam język znaczników HTML przestał spełniać oczekiwania co do tworzenia serwisów WWW i w trakcie ostatnich 20 lat wokół technologii “webowych”, powstały różne narzędzia pozwalające na uatrakcyjnienie wyglądu stron internetowych.

1

Pierwszym narzędziem, które pozwalało na zmianę wyglądu strony były Kaskadowe Arkusze Stylu (CSS), które powstały w roku 1994. Umożliwiały nadawanie odpowiednim elementom zapisanym w języku HTML nowych cech (np. zmiana koloru czcionki).

Technologie teraz i dawniej

Strony internetowe dawniej i obecnie są tworzone z wykorzystaniem języka

HTML (Hyper Text Markup Language)

[początki sięgają 1980 roku]

Z biegiem czasu sam język znaczników HTML przestał spełniać oczekiwania co do tworzenia serwisów WWW i w trakcie ostatnich 20 lat wokół technologii “webowych”, powstały różne narzędzia pozwalające na uatrakcyjnienie wyglądu stron internetowych.

1

Pierwszym narzędziem, które pozwalało na zmianę wyglądu strony były Kaskadowe Arkusze Stylu (CSS), które powstały w roku 1994. Umożliwiały nadawanie odpowiednim elementom zapisanym w języku HTML nowych cech (np. zmiana koloru czcionki).

2

W tym samym czasie, w roku 1994, rozpoczęto pracę nad nowym językiem PHP (Personal Home Page/Forms Interpreter), który pozwalał na generowanie w czasie rzeczywistym kodu w języku HTML bazując na stworzonych przez użytkownika skryptach.

Technologie teraz i dawniej

Strony internetowe dawniej i obecnie są tworzone z wykorzystaniem języka

HTML (Hyper Text Markup Language)

[początki sięgają 1980 roku]

Z biegiem czasu sam język znaczników HTML przestał spełniać oczekiwania co do tworzenia serwisów WWW i w trakcie ostatnich 20 lat wokół technologii “webowych”, powstały różne narzędzia pozwalające na uatrakcyjnienie wyglądu stron internetowych.

1

Pierwszym narzędziem, które pozwalało na zmianę wyglądu strony były Kaskadowe Arkusze Stylu (CSS), które powstały w roku 1994. Umożliwiały nadawanie odpowiednim elementom zapisanym w języku HTML nowych cech (np. zmiana koloru czcionki).

2

W tym samym czasie, w roku 1994, rozpoczęto pracę nad nowym językiem PHP (Personal Home Page/Forms Interpreter), który pozwalał na generowanie w czasie rzeczywistym kodu w języku HTML bazując na stworzonych przez użytkownika skryptach.

3

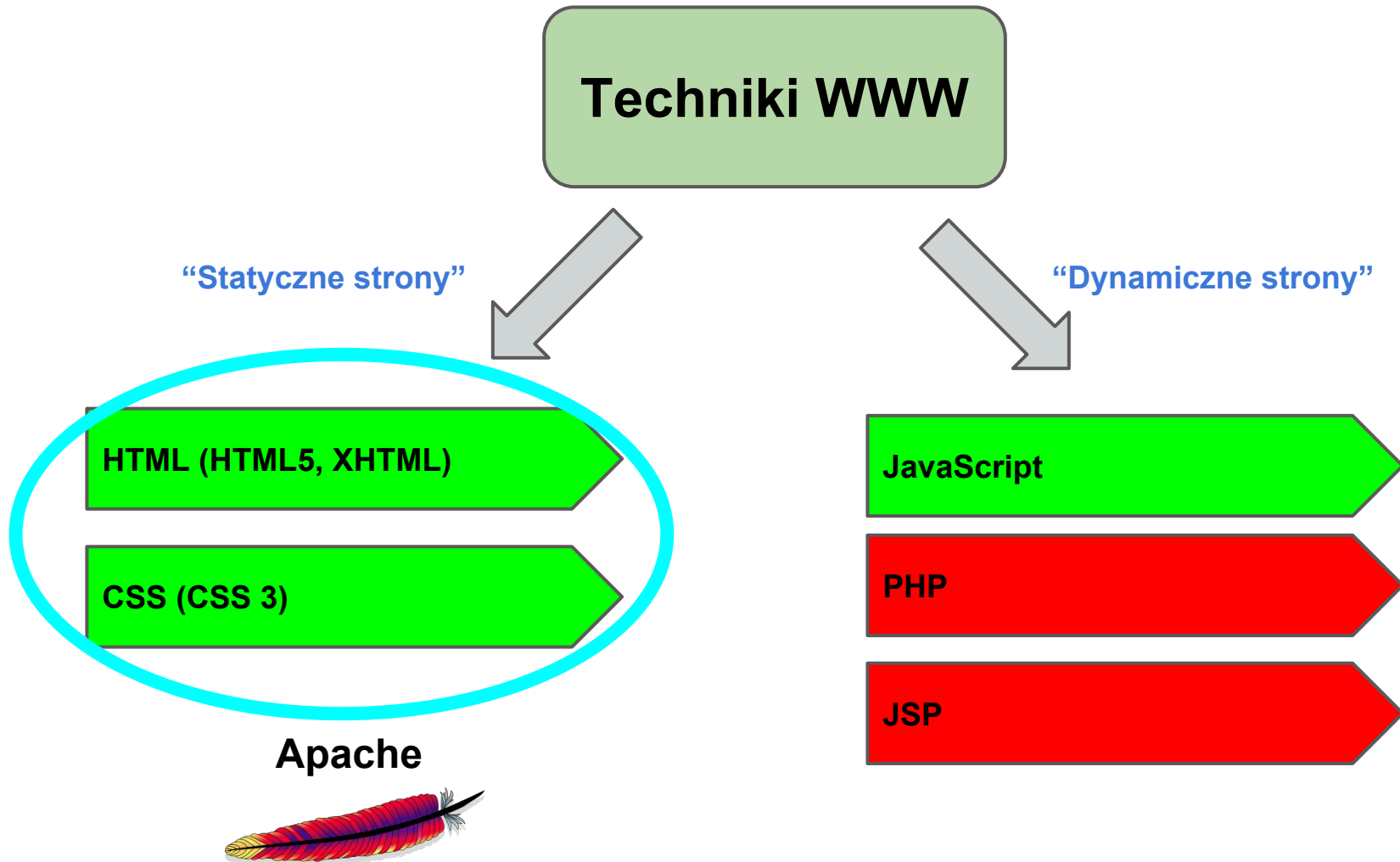
Rok później firma Netscape zapoczątkowała pracę nad językiem skryptowym JavaScript, który miał umożliwiać podniesienie atrakcyjności strony oraz wprowadzenie do niej interaktywności.

Strony statyczne vs. dynamiczne

Statyczna strona WWW - strona taka zawiera w kodzie dane, które są wyświetlane w przeglądarce internetowej. Zawartość strony nie zmienia się pod wpływem interakcji z użytkownikiem, zawsze wyświetlane są te same treści. Każda zmiana danych (treści strony) wymaga ingerencji programisty w kod strony. Użytkownik nie posiadający wiedzy na temat struktury strony oraz języka HTML nie będzie w stanie zmienić zawartości strony.

Dynamiczna strona WWW - strona dynamiczna jest generowana przez serwer HTTP pod wpływem żądań przychodzących od klienta na podstawie przesłanych parametrów i zmiennych. Strony takie dostosowują swoją zawartość w zależności od działań użytkownika w przeglądarce. W takim wypadku zmiany stanu strony mogą być wykonywane po stronie użytkownika dzięki wykorzystaniu języków skryptowych np. JavaScript lub po stronie serwera wykorzystując do tego celu języki programowania takie jak PHP, Perl, Python. W nowoczesnych serwisach internetowych wykorzystywane są obie metody. Dodatkowo najczęściej dane (czyli zawartość strony) jest przechowywana w bazach danych (np. SQL), z których są pobierane w zależności od wywoływanych przez użytkownika żądań.

Technologie wykorzystane na kursie



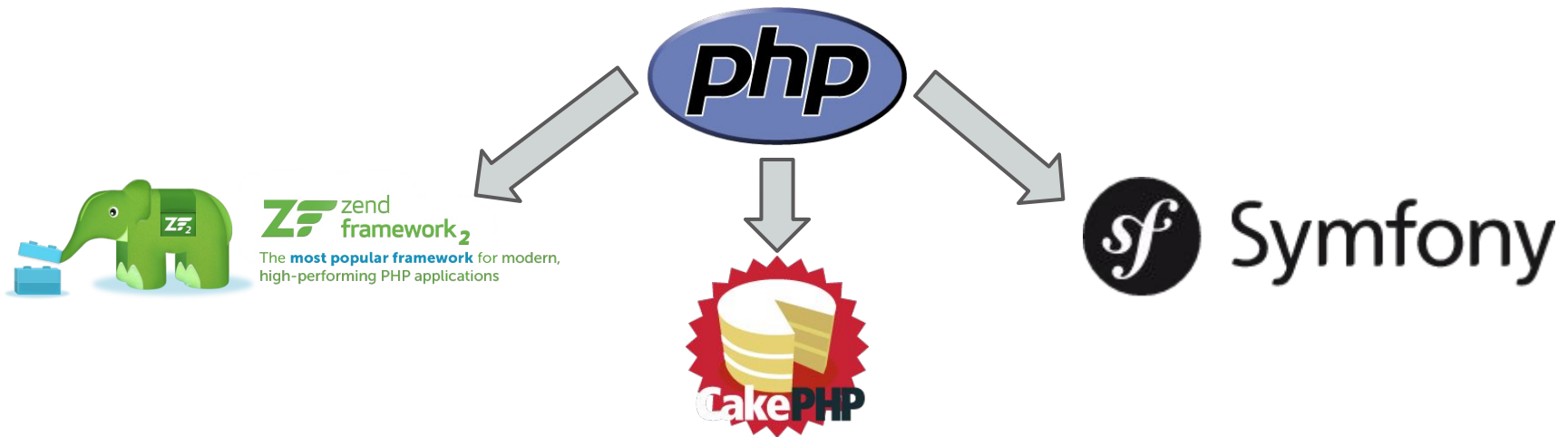
Przegląd dostępnych rozwiązań

Najpopularniejsze środowiska programistyczne:



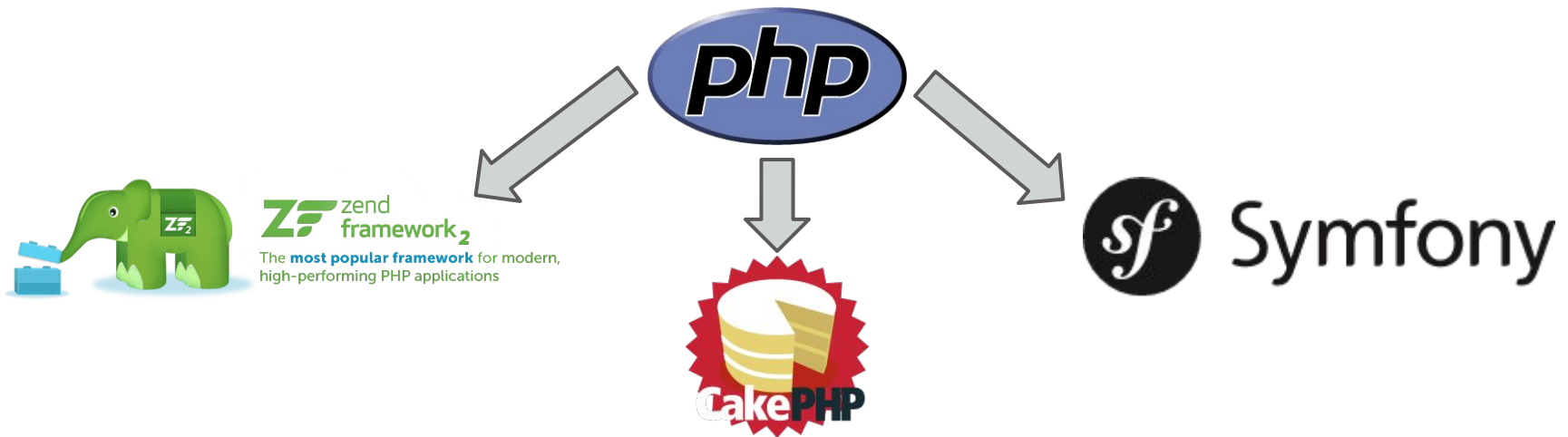
Przegląd dostępnych rozwiązań

Najpopularniejsze środowiska programistyczne:



Przegląd dostępnych rozwiązań

Najpopularniejsze środowiska programistyczne:



oraz systemy CMS (Content Manager System):



Dlaczego NODE.JS a nie PHP

Dlaczego NODE.JS a nie PHP



vs.



Dlaczego NODE.JS a nie PHP



vs.



Obecnie PHP jest wykorzystywane na około 75% wszystkich stron internetowych.

Dlaczego NODE.JS a nie PHP



vs.



Obecnie PHP jest wykorzystywane na około 75% wszystkich stron internetowych.

Ale większość nowych projektów odchodzi od PHP na rzecz NODE.JS jako rozwiązania bardziej wydajnego, elastycznego i niezależnego.

Node.JS jest środowiskiem programistycznym bazującym na języku JavaScript, instalowanym na serwerze. Jego serce stanowi interpreter (silnik) V8 stworzony przez firmę Google.

Daje możliwości tworzenia łatwo skalowalnych aplikacji internetowych, których działanie jest sterowane zdarzeniowo wykorzystując żądania asynchroniczne czyli bez konieczności przeładowania danego zasobu.

Dlaczego NODE.JS a nie PHP

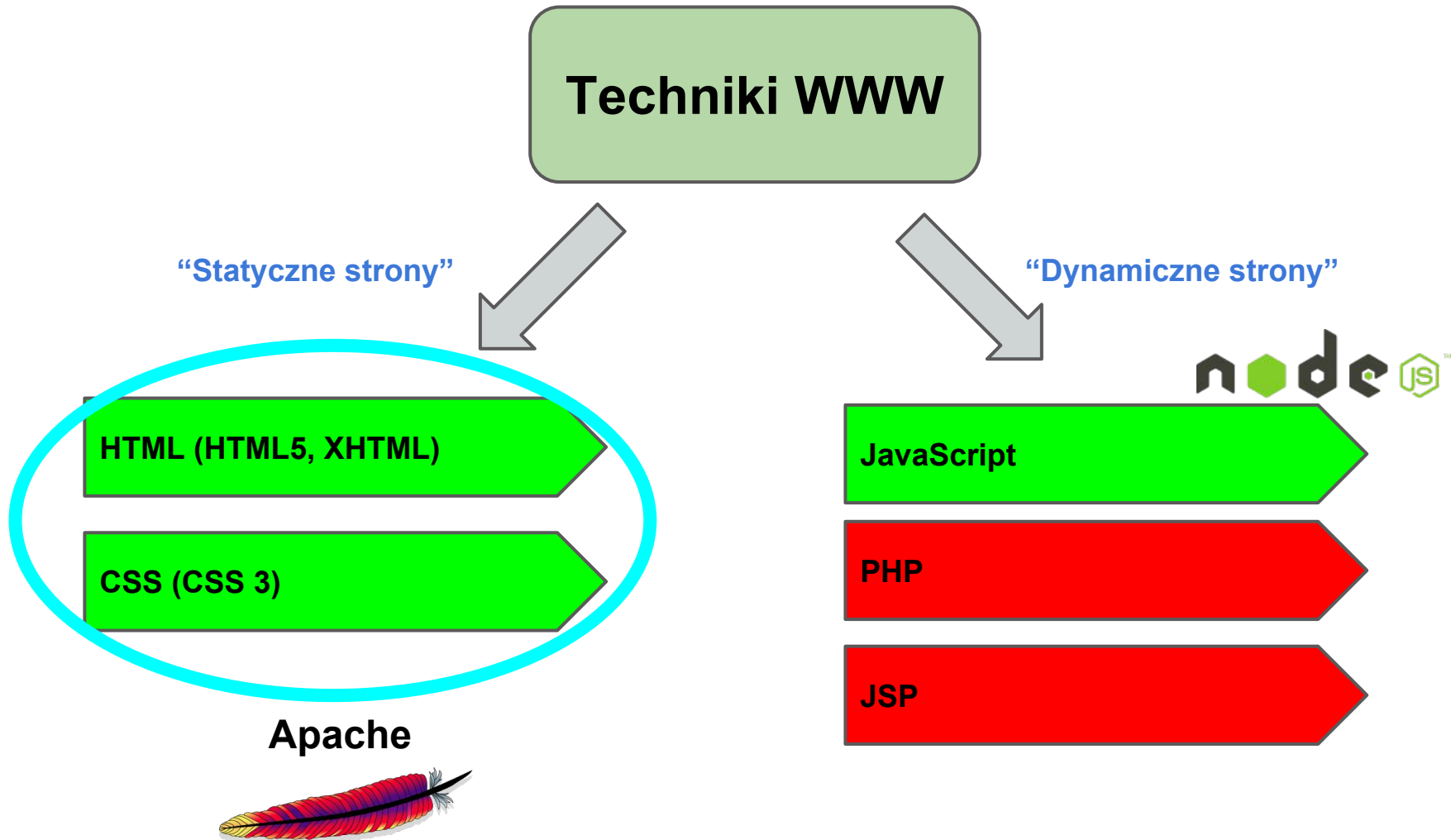


vs.

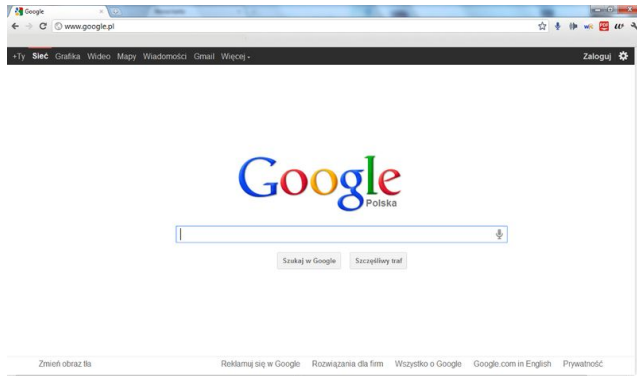


Więcej szczegółów o NODE.JS na kolejnych wykładach

Technologie wykorzystane na kursie



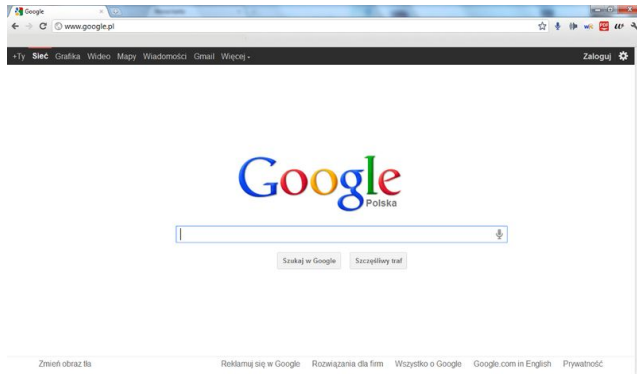
Front-end a Back-end



FRONT-END - pojęciowo odnosi się w technologiach internetowych do kodu **wykonywanego po stronie użytkownika**. W ogólności do tej kategorii można zaliczyć HTML, CSS oraz JavaScript.



Front-end a Back-end



FRONT-END - pojęciowo odnosi się w technologiach internetowych do kodu **wykonywanego po stronie użytkownika**. W ogólności do tej kategorii można zaliczyć HTML, CSS oraz JavaScript.



BACK-END - pojęciowo odnosi się w technologiach internetowych do kodu **wykonywanego po stronie serwera**. W ogólności do tej kategorii można zaliczyć PHP, NODE.JS, Perl, CGI, Ruby



HTML - początek

HTML (Hyper Text Markup Language) - powstał na bazie zasad języka SGML (Standard Generalized Markup Language), który jest pewnego rodzaju “meta-językiem” służącym do definiowania znaczników i zasad ich używania.

HTML - początek

HTML (Hyper Text Markup Language) - powstał na bazie zasad języka SGML (Standard Generalized Markup Language), który jest pewnego rodzaju “meta-językiem” służącym do definiowania znaczników i zasad ich używania.

Pierwsza wersja HTML została opracowana przez Tima Bernersa-Lee fizyka pracującego w CERN i została opublikowana w 1991 roku. Pierwsza propozycja zawierała 22 znaczniki.



Przewodniczący konsorcjum
W3C
(www.w3c.org)



HTML - początek

HTML (Hyper Text Markup Language) - powstał na bazie zasad języka SGML (Standard Generalized Markup Language), który jest pewnego rodzaju “meta-językiem” służącym do definiowania znaczników i zasad ich użytkowania.

Pierwsza wersja HTML została opracowana przez Tima Bernersa-Lee fizyka pracującego w CERN i została opublikowana w 1991 roku. Pierwsza propozycja zawierała 22 znaczniki.

Podstawowe cechy standardu HTML:

- **hipertekstowość** - za pomocą znaczników można tworzyć hiperpołączenia (hiperlinki), które wskazują na inne zasoby w obrębie tej samej witryny lub dowolnej innej zamieszczone w sieci. Dzięki hiperpołączeniom można w łatwy sposób poruszać się po stronach.
- **uniwersalność** - dokument zapisany w standardzie HTML jest zwykłym plikiem tekstowym (ASCII), więc może być stworzony na dowolnej platformie systemowej bez żadnego specjalistycznego i dedykowanego oprogramowania.



Przewodniczący konsorcjum
W3C
(www.w3c.org)



HTML - początek

HTML (Hyper Text Markup Language) - powstał na bazie zasad języka SGML (Standard Generalized Markup Language), który jest pewnego rodzaju “meta-językiem” służącym do definiowania znaczników i zasad ich używania.

Ostatni obowiązujący standard języka HTML opublikowany przez konsorcjum W3, oznaczony jest numerem 4.01 i powstał w 1999 roku. W wersji tej zostały uporządkowane i ujednolicone dotychczas niekompatybilne ze sobą wersje HTML w oparciu o standard zapisu SGML.

HTML - początek

HTML (Hyper Text Markup Language) - powstał na bazie zasad języka SGML (Standard Generalized Markup Language), który jest pewnego rodzaju “meta-językiem” służącym do definiowania znaczników i zasad ich używania.

Ostatni obowiązujący standard języka HTML opublikowany przez konsorcjum W3, oznaczony jest numerem 4.01 i powstał w 1999 roku. W wersji tej zostały uporządkowane i ujednolicone dotychczas niekompatybilne ze sobą wersje HTML w oparciu o standard zapisu SGML.

Drugim obowiązującym standardem skonstruowanym na bazie HTML 4.01 oraz w oparciu o zasady języka XML (Extensible Markup Language) jest standard XHTML 1.1. Standard ten łączy zalety HTML oraz XML dając ostatecznie specyfikę o funkcjonalnościach XML. Jedną z zalet takiego rozwiązania jest możliwość łączenia XHTML z innymi językami opartymi o XML oraz można do ich przetwarzania używać transformat XSLT (*Extensible Stylesheet Language Transformations*).

HTML - początek

HTML (Hyper Text Markup Language) - powstał na bazie zasad języka SGML (Standard Generalized Markup Language), który jest pewnego rodzaju “meta-językiem” służącym do definiowania znaczników i zasad ich używania.

Ostatni obowiązujący standard języka HTML opublikowany przez konsorcjum W3, oznaczony jest numerem 4.01 i powstał w 1999 roku. W wersji tej zostały uporządkowane i ujednolicone dotychczas niekompatybilne ze sobą wersje HTML w oparciu o standard zapisu SGML.

Drugim obowiązującym standardem skonstruowanym na bazie HTML 4.01 oraz w oparciu o zasady języka XML (Extensible Markup Language) jest standard XHTML 1.1. Standard ten łączy zalety HTML oraz XML dając ostatecznie specyfikę o funkcjonalnościach XML. Jedną z zalet takiego rozwiązania jest możliwość łączenia XHTML z innymi językami opartymi o XML oraz można do ich przetwarzania używać transformat XSLT (*Extensible Stylesheet Language Transformations*).

Różnica pomiędzy HTML i XHTML jest z pozoru jest nieznaczna, natomiast istotnie standardy te różnią się z punktu widzenia systematyzacji informacji, np. strona niepoprawnie przygotowana w języku XHTML nie powinna być wyświetlona podobnie jak to jest z XML-em. Natomiast HTML zostanie wyświetlony mimo błędów.

Znacznik

Znacznik (tag)- pojęcie w informatyce odnoszące się do klasyfikowania dowolnych informacji, ale najczęściej tekstu. Znaczniki nie są elementami żadnego języka programowania, określają działania które mają zostać wykonane w stosunku do zasobu jaki kwalifikują. Najczęściej są to działania które informują jak dana informacja ma zostać zaprezentowana.

W dziedzinie tworzenia stron internetowych w standardach HTML i XHTML znacznik to tekst umieszczony pomiędzy znakiem większości i znakiem mniejszości:

<znacznik>

Znacznik

Znacznik (tag)- pojęcie w informatyce odnoszące się do klasyfikowania dowolnych informacji, ale najczęściej tekstu. Znaczniki nie są elementami żadnego języka programowania, określają działania które mają zostać wykonane w stosunku do zasobu jaki kwalifikują. Najczęściej są to działania które informują jak dana informacja ma zostać zaprezentowana.

W dziedzinie tworzenia stron internetowych w standardach HTML i XHTML znacznik to tekst umieszczony pomiędzy znakiem większości i znakiem mniejszości:

<znacznik>

Znaczniki w innych dziedzinach np.: LaTeX:

```
\begin{center}  
\huge{Tytuł dokumentu}  
\end{center}
```

Znacznik

<znacznik>

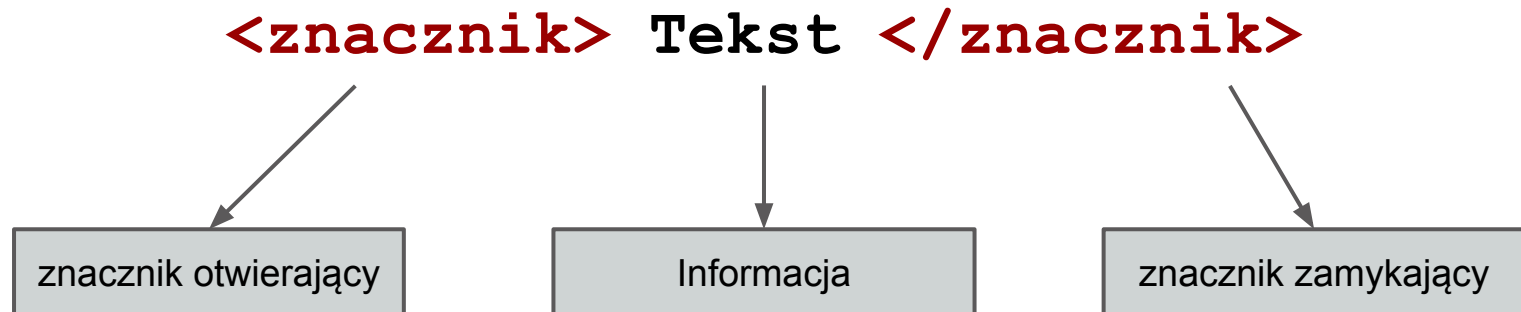
Najczęściej znacznik służy do kwalifikacji obszaru tekstu (w ogólności dowolnej informacji). Kwalifikacja polega na umieszczeniu tekstu pomiędzy znacznikiem otwierającym i znacznikiem zamykającym:

<znacznik> Tekst </znacznik>

Znacznik

<znacznik>

Najczęściej znacznik służy do kwalifikacji obszaru tekstu (w ogólności dowolnej informacji). Kwalifikacja polega na umieszczeniu tekstu pomiędzy znacznikiem otwierającym i znacznikiem zamykającym:



Znacznik

<znacznik>

Czasem znacznik spełnia tylko rolę określenia układu w dokumencie i wtedy nie obejmuje swoim zasięgiem żadnej informacji, jest to tzw. znacznik “pusty”

<znacznik/>

Znacznik

<znacznik>

Czasem znacznik spełnia tylko rolę określenia układu w dokumencie i wtedy nie obejmuje swoim zasięgiem żadnej informacji, jest to tzw. znacznik “pusty”

<znacznik/>



Znacznik pusty

Uwaga:

- według standardu HTML 4.01 nie trzeba pisać “ukośnika w znaczniku pustym.
- w XHTML znacznik musi być zamknięty, inaczej jest to błąd składniowy.
- ze względu na wsteczną kompatybilność zaleca się przed ukośnikiem w znaku pustym stawiać spację
 .

Znacznik

<znacznik>

Znacznik poza słowem kluczowym definiującym jego cechę, może zawierać różne atrybuty modyfikujące jego podstawowe własności. Atrybuty są definiowane w znaczniku otwierającym w następujący sposób:

<znacznik atrybut1="wartość">

Tekst

</znacznik>

Znacznik

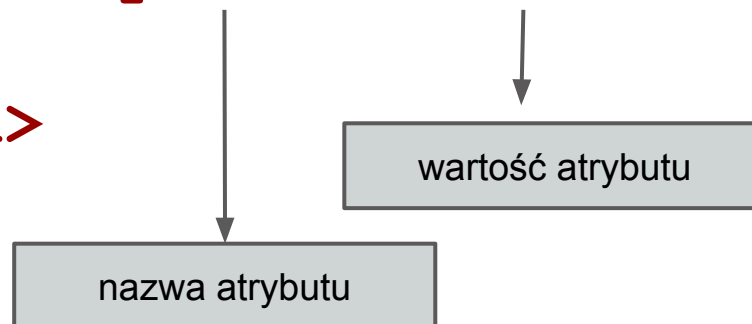
<znacznik>

Znacznik poza słowem kluczowym definiującym jego cechę, może zawierać różne atrybuty modyfikujące jego podstawowe własności. Atrybuty są definiowane w znaczniku otwierającym w następujący sposób:

<znacznik atrybut1="wartość">

Tekst

</znacznik>



Uwaga:

- nazwy elementów (znaczników) oraz atrybutów według standardu XHTML powinny być pisane małymi literami.

Nazwy plików

Jeśli tworzymy dokument hipertekstowy to dla poprawnej interpretacji jego zawartości przez przeglądarki oraz dla identyfikacji pliku przez serwer i system operacyjny zaleca się aby plikom nadawać rozszerzenia **“.html”** lub **“htm”**.

Dodatkowo jako, że pliki takie stają się częścią zasobów internetowych i są identyfikowane przez adresy URL, zaleca się aby konsekwentnie w nazwach plików używać **małych liter**.



Nazwy plików

Jeśli tworzymy dokument hipertekstowy to dla poprawnej interpretacji jego zawartości przez przeglądarki oraz dla identyfikacji pliku przez serwer i system operacyjny zaleca się aby plikom nadawać rozszerzenia **“.html”** lub **“htm”**.

Dodatkowo jako, że pliki takie stają się częścią zasobów internetowych i są identyfikowane przez adresy URL, zaleca się aby konsekwentnie w nazwach plików używać **małych liter**.

index.html



index.htm

mojastrona.html

twojastrona.htm



GLOWNA.HTML

PODSTRONA.html

podpodstrona.HTML



Uwaga:

- gdy przygotowujemy dokument tekstowy, dobrą praktyką jest robienie “wcięć” w kodzie,
 - przeglądarki ignorują zbędne spacje oraz przejścia do nowej linii robione za pomocą klawisza ENTER.
-

Znaczniki podstawowe

Nazwy elementów zapisanych w znacznikach w języku HTML i XHTML wywodzą się od słów z języka angielskiego.

Wszystkie dokumenty hipertekstowe niezależnie od używanego standardu, posiadają trzy podstawowe znaczniki:

- | | | |
|---------------------------|----------------------------|---|
| <code><html></code> | <code></html></code> | - znacznik wskazujący na dokument hipertekstowy (otacza wszystkie pozostałe znaczniki) |
| <code><head></code> | <code></head></code> | - znacznik wskazuje na nagłówek dokumentu hipertekstowego (zasadniczo znajduje się na jego początku). |
| <code><body></code> | <code></body></code> | - znacznik wskazuje na część główną dokumentu hipertekstowego, w niej znajdują się wszystkie inne znaczniki związane z prezentowaniem danych. |



Znaczniki podstawowe

Nazwy elementów zapisanych w znacznikach w języku HTML i XHTML wywodzą się od słów z języka angielskiego.

Wszystkie dokumenty hipertekstowe niezależnie od używanego standardu, posiadają trzy podstawowe znaczniki:



- `<html>` `</html>` - znacznik wskazujący na dokument hipertekstowy (otacza wszystkie pozostałe znaczniki)
- `<head>` `</head>` - znacznik wskazuje na nagłówek dokumentu hipertekstowego (zasadniczo znajduje się na jego początku).
- `<body>` `</body>` - znacznik wskazuje na część główną dokumentu hipertekstowego, w niej znajdują się wszystkie inne znaczniki związane z prezentowaniem danych.

Ważną częścią każdego języka jest możliwość dodawania komentarzy. W języku HTML, komentarz jest zaznaczany w następujący sposób:

`<!-- Komentarz -->`

W komentarzu możemy umieścić zwykły tekst lub inne znaczniki. Zakomentowane znaczniki nie będą wyświetlane przez przeglądarkę w trakcie wczytywania strony.

Znaczniki podstawowe

Nazwy elementów zapisanych w znacznikach w języku HTML i XHTML wywodzą się od słów z języka angielskiego.

Wszystkie dokumenty hipertekstowe niezależnie od używanego standardu, posiadają trzy podstawowe znaczniki:



- `<html>` `</html>` - znacznik wskazujący na dokument hipertekstowy (otacza wszystkie pozostałe znaczniki)
- `<head>` `</head>` - znacznik wskazuje na nagłówek dokumentu hipertekstowego (zasadniczo znajduje się na jego początku).
- `<body>` `</body>` - znacznik wskazuje na część główną dokumentu hipertekstowego, w niej znajdują się wszystkie inne znaczniki związane z prezentowaniem danych.

Ważną częścią każdego języka jest możliwość dodawania komentarzy. W języku HTML, komentarz jest zaznaczany w następujący sposób:

`<!-- Komentarz -->`

UWAGA: Ale komentarz cały czas jest widoczny w kodzie strony i można go odczytać mimo nie wyświetlania go przez przeglądarkę.

Podstawowa struktura dokumentu

<code><!DOCTYPE ... ></code>	deklaracja typu dokumentu
<code><html></code>	początek dokumentu hipertekstowego
<code><head></code>	sekcja nagłówkowa
<code><title></code> Tytuł strony <code></title></code>	tytuł strony
<code></head></code>	
<code><body></code> Treść strony <code></body></code>	sekcja zawartości strony
<code></html></code>	koniec dokumentu hipertekstowego

Struktura hierarchiczna z zagnieżdżeniami
