



Fundusze  
Europejskie  
Wiedza Edukacja Rozwój



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz Społeczny



# Przeglądarki internetowe jako interfejs webowy, komunikator i narzędzie programistyczne

Uniwersytet Morski w Gdyni  
Narzędzia Informatyczne

*Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój*

# Przeglądarka internetowa

Przeglądarki internetowe należą do grupy jednego z bardziej złożonych rodzajów oprogramowania użytkowego, z jakim styka się na co dzień przeciętny użytkownik komputera.

**Przeglądarka internetowa jest przede wszystkim programem do wyświetlania stron WWW, ale dodatkowo dzisiejsza przeglądarka to kompozycja wielu elementów, zajmujących się generowaniem zarówno statycznej, jak i dynamicznej oraz multimedialnej treści.**

Bez problemu jest w stanie wyświetlić dokumenty w wielu popularnych formatach (np. pdf) czy posłużyć za klienta poczty elektronicznej, komunikator bądź interfejs do edycji dokumentów przechowywanych w chmurze.

# Przeglądarka internetowa

Uniwersalność zastosowań doprowadziła do powstania całych systemów operacyjnych zbudowanych wokół przeglądarki (ChromeOS, FirefoxOS).

Z drugiej strony, przeglądarka, będąc swoistym oknem na świat i punktem „wejścia” do sieci, stanowi jednocześnie najczęstsze **źródło wszelkich „infekcji” naszego komputera**. To przez nią do naszego systemu przedostają się trojany, malware i inne złośliwe oprogramowanie.

Znajomość budowy i działania przeglądarki może nam pomóc bronić się przed nimi (choćby poprzez świadomość, w jaki sposób interpretowany jest element <iframe> lub jak przeglądarka obsługuje ciasteczka).

# Udział przeglądarek internetowych w polskim rynku

Najpopularniejsze przeglądarki internetowe w Polsce (wg [gs.statcounter.com](http://gs.statcounter.com) – październik 2018, z uwzględnieniem wszystkich platform):

Chrome – 77,04%

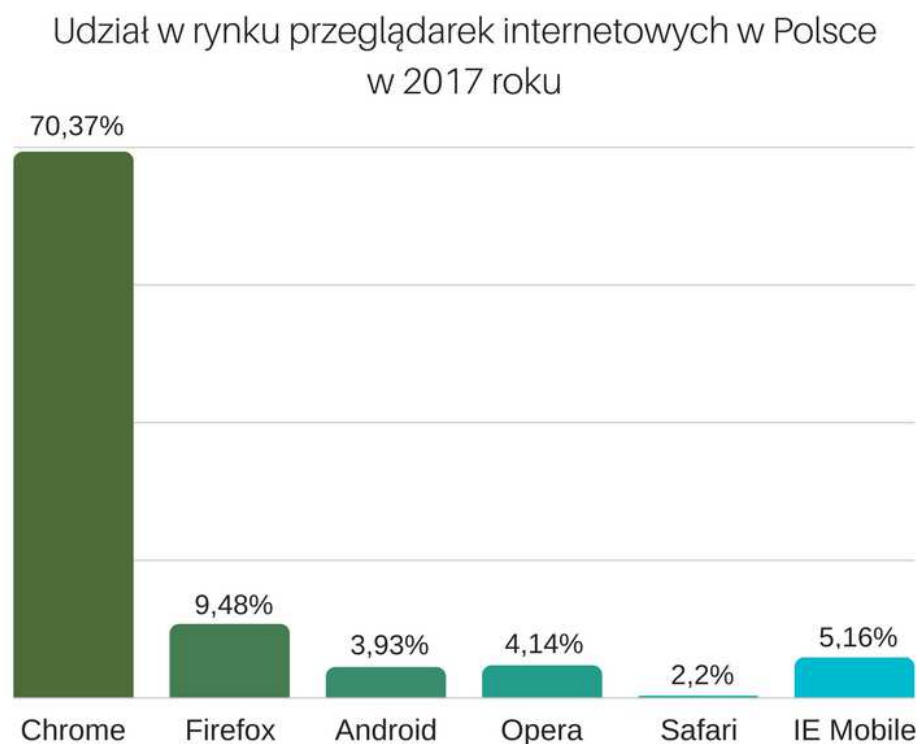
Firefox – 7,56%

Samsung Internet – 4,42%

Opera – 3,83%

Safari – 2,52%

Android – 1,47%.



# Popularność przeglądarek internetowych na świecie

Tak prezentują się najlepsze przeglądarki internetowe na świecie ze względu na ich popularność (wg gs.statcounter.com – październik 2018):

Chrome – 61,55%

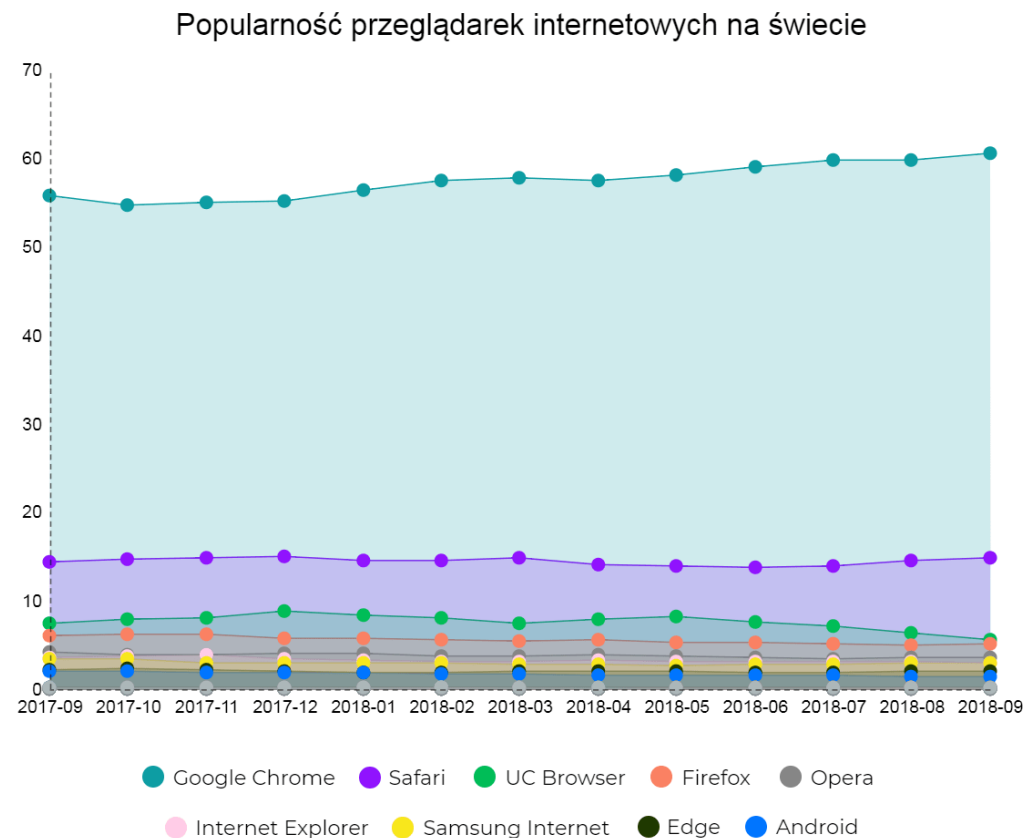
Safari – 15,13%

Firefox – 5,02%

UC Browser – 4,41%

Opera – 3,16%

Internet Explorer – 2,88%.



# Popularność przeglądarek internetowych na świecie

Ranking przeglądarek internetowych na świecie na komputerach stacjonarnych (wg [gs.statcounter.com](https://gs.statcounter.com) – październik 2018):

Chrome – 69,64%

Firefox – 10,14%

Internet Explorer – 6,01%

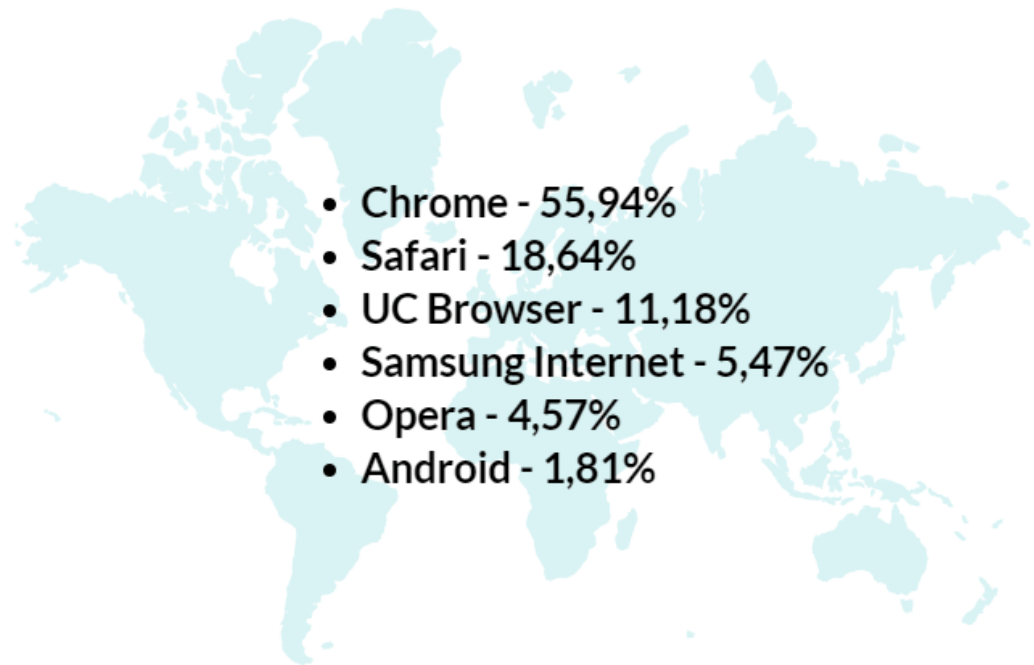
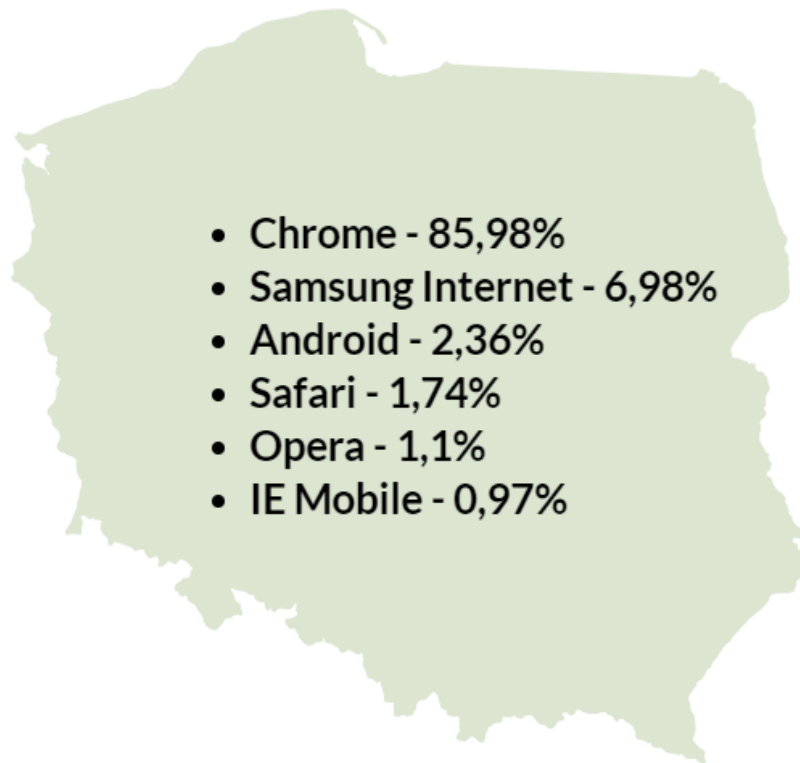
Safari – 5,61%

Edge – 4,21%

Opera – 2,29%

# Popularność przeglądarek

Ranking przeglądarek internetowych 2018 -  
urządzenia mobilne



Źródło: [gs.statcounter.com](https://gs.statcounter.com) - sierpień 2018

# Współczesne aplikacje internetowe

Aplikacja internetowa (aplikacja webowa, webaplikacja) to oprogramowanie stworzone przy użyciu kilku technologii, 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) – celowo wymieniam je w tym miejscu, jako osobny rodzaj aplikacji, gdyż obecnie portale te posiadają także interfejsy klienckie w postaci np. natywnych aplikacji na urządzenia mobilne działające pod kontrolą systemów Android czy iOS),
- klient poczty (GMail, Outlook Web Access, interfejsy poczty WP, Onet, O2 itp.),
- sklepy internetowe, transakcyjne systemy bankowe,
- webowe interfejsy komunikatorów, IRC, gry MMORPG.

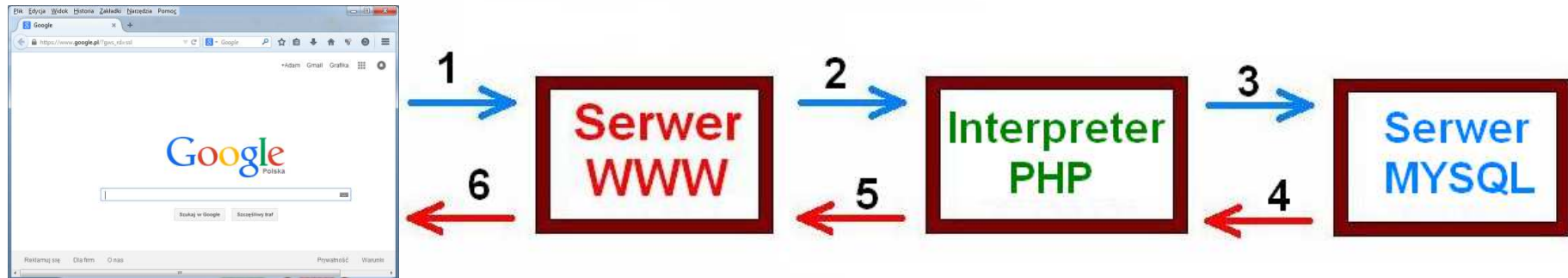


# Relacja klient-serwer między przeglądarką a serwerem WWW

- Najprostszy system internetowy obejmuje przeglądarkę internetową, serwer WWW i możliwość nawiązania połączenia między nimi.
- Przeglądarka wysyła żądanie do serwera WWW a serwer odpowiada na nie.
- Taka architektura sprawdza się w przypadku obsługi przez serwer WWW statycznych stron WWW.

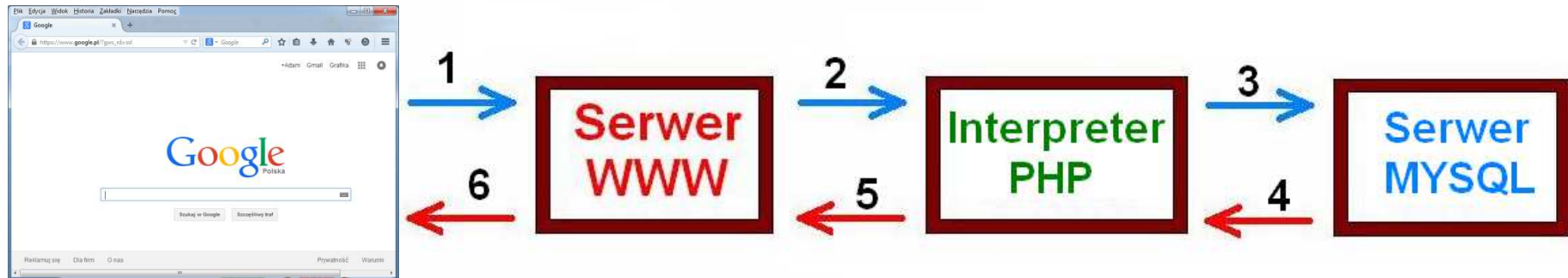


# Klasyczna architektura strony dynamicznej



1. Przeglądarka internetowa użytkownika wysyła żądanie udostępniania określonej strony WWW. Za pośrednictwem formularza HTML i skryptu PHP można żądać wyświetlenia pewnych danych przechowywanych w bazie danych.
2. Serwer WWW przyjmuje żądanie wyświetlenia strony zawierającej skrypt PHP. Serwer odnajduje właściwy plik i przekazuje go do interpretera PHP.
3. Interpreter PHP przetwarza skrypt PHP. Skrypt zawiera funkcje łączenia z bazą danych i wykonania zapytań. Następnie dochodzi do połączenia z serwerem MySQL i przesłania zapytania.

# Klasyczna architektura strony dynamicznej



4. Serwer MySQL przyjmuje zapytanie i je przetwarza, po czym rezultat odsyłany jest do interpretera PHP.
5. Interpreter kończy wykonywanie skryptu, który zazwyczaj formatuje otrzymane wyniki zgodnie ze standardami HTML, po czym przesyła wynikowy kod HTML do serwera WWW.
6. Serwer WWW przesyła kod HTML do przeglądarki, która wyświetla pobrane z bazy danych informacje.

# Najważniejsze technologie wytworzenia stron www

HTML – jest to język znaczników, za pomocą którego opisuje się strukturę strony, będącą elementem aplikacji widocznym w oknie przeglądarki.

To HTML odpowiada za podział strony na paragrafy, nagłówki, listy, tabelki.

Pozwala też na wyświetlanie elementów formularzy oraz hiperłączy (linków) do innych dokumentów bądź elementów aplikacji webowej.

HTML5 (Hypertext Markup Language 5), który pojawił się kilka lat temu, a obecnie staje się standardem nie tylko, jeśli chodzi o aplikacje typowo webowe, czyli działające jedynie w przeglądarkach. Poza dobrze znanym językiem hipertekstowym HTML, HTML5 to tak naprawdę zbiór wielu, często nowych, technologii służących do budowy tzw. RIA (Rich Internet Application).

# Najważniejsze technologie wytworzenia stron www

RIA (Rich Internet Application) - skrót ten oznacza aplikacje, które mogą komunikować się z użytkownikiem, wyświetlać elementy multimedialne, odtwarzać filmy czy umożliwiać ich nagrywanie. Służą do tego m.in. nowe elementy, takie jak <video> i <audio>; obsługa kamer i mikrofonów (UserMedia API), zdalna komunikacja pomiędzy przeglądarkami (WebRTC), geolokalizacja (Geolocation API), obsługa systemu plików (File API), obsługa gniazd sieciowych (Web Sockets) i wiele, wiele innych. Część z nich wymaga użycia języka JavaScript.

# Najważniejsze technologie wytwarzania stron www

CSS – technologia służy do opisu, jak poszczególne elementy strony WWW mają wyglądać (m.in. jaki mają mieć kolor czy też jak mają być położone na stronie względem innych elementów).

Najnowsza wersja tzw. kaskadowych arkuszy stylów (Cascading Style Sheet) pełna jest nowych rozwiązań, które przenoszą współczesne aplikacje internetowe w zupełnie inny wymiar. Animacje, transformacje i dowolne przemieszczanie elementów drzewa DOM, cienie, gradienty, zaokrąglenia, możliwość dołączania własnych fontów to tylko wycinek możliwości, jakie dają nowe rozszerzenia CSS.

# Najważniejsze technologie wytwarzania stron www

JavaScript – interpretowany przez jeden z mechanizmów przeglądarki język programowania służący do definiowania zachowania elementów strony w reakcji na akcje podejmowane przez użytkownika (np. kliknięcie elementu bądź wysłanie formularza).

Drugim popularnym zastosowaniem JavaScriptu jest implementacja części logiki biznesowej aplikacji po stronie klienta (od niedawna także w całości po stronie serwerowej w przypadku wykorzystania serwera Node.js).

# Najważniejsze technologie wytwarzania stron www

Język JavaScript, a właściwie ECMAScript wersja 5 (a wkrótce wersja 6, nad którą prace właśnie dobiegają końca), szturmem wdarł się na podium współczesnych języków programowania. Jego niesamowita popularność nie kończy się jedynie na stronach i aplikacjach internetowych. W ostatnich latach bardzo dynamicznie rozwijają się projekty, które albo wykorzystują JavaScript bezpośrednio, albo stanowią dla niego platformę do tworzenia aplikacji natywnych: Node.js, FirefoxOS, PhoneGAP, interfejs Metro w Windows 8, Gnome Shell w systemach linuksowych, Cinnamon w systemie Mint – to tylko część zastosowań tego niezwykle elastycznego, niesamowicie szybkiego i pozornie łatwego do opanowania języka.



# Najważniejsze technologie wytwarzania stron www

Wcześniej język ten wykorzystywano do prostego dodawania do przycisków niezbyt skomplikowanych efektów, tworzenia zegarków czy walidacji pól formularza po stronie klienta, **obecnie jest pełnoprawnym, zorientowanym obiektowo i opartym na dziedziczeniu** prototypowym, skryptowym językiem o ogromnych możliwościach, na które składają się chociażby: frameworki wykorzystujące wzorce MVC czy MVVM, dzięki którym bez większych problemów można zaimplementować praktycznie dowolną aplikację (Backbone.js, AngularJS), biblioteki do testów jednostkowych czy end-to-end (QUnit, Mocha), systemy buildujące (Grunt), biblioteki do obsługi baz NoSQL (MongoDB) czy też relacyjnych, jak MySQL czy grafiki 3D (WebGL). Doczekał się nawet własnego serwera WWW wraz z całym systemem pakietów (Node.js oraz npm – Node Packaged Modules). Przykładem dużej aplikacji zbudowanej z wykorzystaniem jedynie wymienionych technologii (Node.js, Express.js) jest nowa wersja serwisu MySpace.

# Najważniejsze technologie wytwarzania stron www

Często przy stronach www wykorzystuje się XML oraz standardy implementowane w XML (np. grafika wektorowa w SVG).

JSON (JavaScript Object Notation) jest sposobem zapisu danych w postaci obiektów języka JavaScript i jest wykorzystywany do wymiany danych pomiędzy aplikacją a serwerem.

Pamiętajmy, że cały czas mówimy o technologiach obsługiwanych przez przeglądarkę, czyli część kliencką (nie zagłębialiśmy się w szczegóły strony serwerowej).

# Standaryzacja WWW

Organizacja W3C (World Wide Web Consortium), stojąca obecnie na straży standardów przygotowuje propozycje, tzw. drafty, nowych rozwiązań i technologii, a następnie koordynuje pracę nad nimi przy współpracy wymienionych firm.

W3C jest zorganizowana w tak zwane grupy robocze (Working groups), pracujące nad standardami, które następnie są rekomendowane i implementowane w przeglądarkach.

# Standaryzacja WWW

Proces wdrażania nowej technologii, zanim stanie się właśnie takim standardem i znajdzie się oficjalnie w dokumentacji HTML5, CSS czy JavaScript, trwa często nawet kilka lat.

Oprócz W3C do czołowych kreatorów współczesnych technologii webowych należą Mozilla, Google oraz Apple. To te cztery wymienione firmy (dwie z nich to właściwie organizacje, a nie typowe korporacje: Mozilla i W3C) odpowiadają za to, co widzą miliardy internautów w swoich przeglądarkach.

# Silnik przeglądarki

Za wyświetlenie strony internetowej nie jest odpowiedzialna sama przeglądarka, a zespół współpracujących ze sobą komponentów znanych pod nazwą „**silnika renderującego**” (rendering engine).

**Silnik renderujący odpowiada za:**

- przetworzenie kodu HTML/XML w drzewo DOM,
- nadanie elementom odpowiedniego wyglądu przy pomocy arkuszy CSS,
- dołączenie zdjęć i grafiki wektorowej (SVG) i wreszcie przekazania do oprogramowania graficznego systemu, który ostatecznie wyświetla stronę na ekranie komputera (nie zajmuje się tym bezpośrednio sama przeglądarka).

# Silnik przeglądarki

Zupełnie osobny komponent parsuje i wykonuje kod JavaScript.

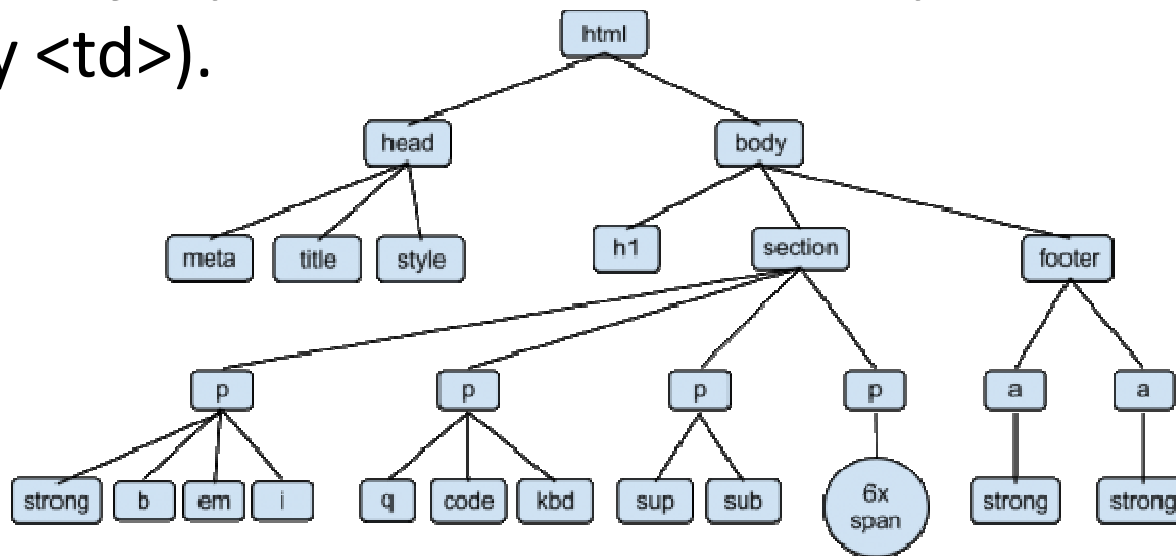
Komponent ten to silnik-interpreter JavaScript.

To on, wraz z silnikiem renderującym, generuje całość – kompletną, interaktywną aplikację internetową, którą użytkownik widzi i może obsługiwać w oknie przeglądarki.

nazwa	wersja (lipiec 2013)	producent	silnik renderujący	silnik JavaScript
Firefox	22	Mozilla	Gecko	SpiderMonkey; od FF 22 OdinMonkey ze wsparciem dla asm.js
Chrome	27	Google	WebKit (do wersji 27), Blink(od wersji 28)	V8
Internet Explorer	10	Microsoft	Trident	Chakra
Opera	15.0	Opera	Presto (do wersji 12.15), Blink (od wersji Opera Next/Opera 15.0)	Carakan

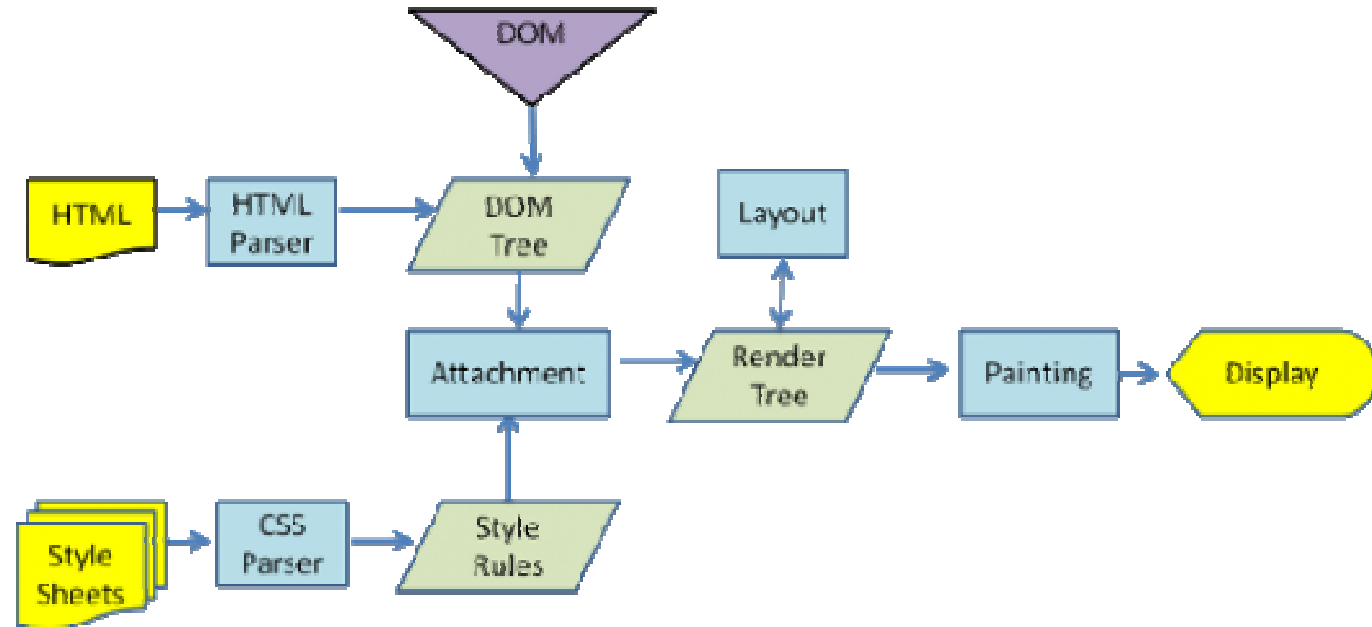
# Drzewo DOM

Drzewo DOM (Document Object Model) to struktura drzewiasta symbolizująca układ elementów strony. Elementem głównym (root) jest `<html>`, wszystkie kolejne elementy tworzą zależności dzieci-rodzic (np. element `<table>` jest rodzicem dla kilku wierszy tabeli, czyli elementów `<tr>`. Z kolei w każdym wierszu istnieje wiele dzieci nadrzędnych elementów `<tr>`, czyli komórki tabeli – elementy `<td>`).



# Drzewo DOM

Po analizie i parsowaniu kodu CSS elementom drzewa DOM jest nadawany wygląd, zgodnie z regułami zawartymi w arkuszach stylów, oraz są one rozmieszczane względem siebie. Całość tworzy tzw. „render tree” (drzewo renderingu), które ostatecznie jest wyświetlane w oknie przeglądarki (tzw. faza „painting”, czyli „odmalowywania”, bądź po prostu rysowania strony w oknie przeglądarki).



*Proces generowania dokumentu w przeglądarce opartej na silniku WebKit (Google Chrome, Chromium, Safari).*



# Drzewo DOM

Cały opisany proces w optymalnych warunkach, przy założeniu prawidłowo skonstruowanego dokumentu, **powinien wykonać się w maksymalnie 1000 ms (1 sekunda)**. Jest to czas, po którym statystyczny internauta zauważa już tzw. opóźnienie w ładowaniu strony i zaczyna wykazywać pierwsze oznaki zniecierpliwienia.

# Struktura strony HTML

Każda strona internetowa bazująca na HTML musi spełniać zasady struktury wytyczone przez swoje DTD. Oznacza to, że język HTML jest zbiorem znaczników, których może użyć web developer według ściśle określonych warunków.

Większość znaczników posiada właściwości oraz atrybuty. Właściwości określają w jaki sposób zachowa się treść znacznika oraz w jaki sposób ta treść zostanie sformatowana i wyświetlona. Właściwości edytować można stosując kaskadowe arkusze stylów.

HTML wraz z CSSem nadają się do stworzenia szablonu strony i umieszczeniu na niej danych, lecz nie umożliwiają zbyt na interakcję z użytkownikiem.

# Struktura strony HTML

Na każdej stronie internetowej umieszczona musi być informacja o typie dokumentu oraz trzy znaczniki: `<html>`, `<head>` oraz `<body>`.

Typ dokumentu deklaruje się w pierwszej linii kodu HTML. W przypadku HTML5 ta deklaracja przyjmuje postać: `<!DOCTYPE html>`.

W kolejnej linii powinien pojawić się znacznik `<html>`. Jest to korzeń dokumentu HTML, który zawierać musi znaczniki `<head>` oraz `<body>`. Znacznik `<html>` może posiadać atrybuty, m.in. warto umieścić atrybut `lang`, by zdefiniować główny język strony (np. `<html lang="pl">`).

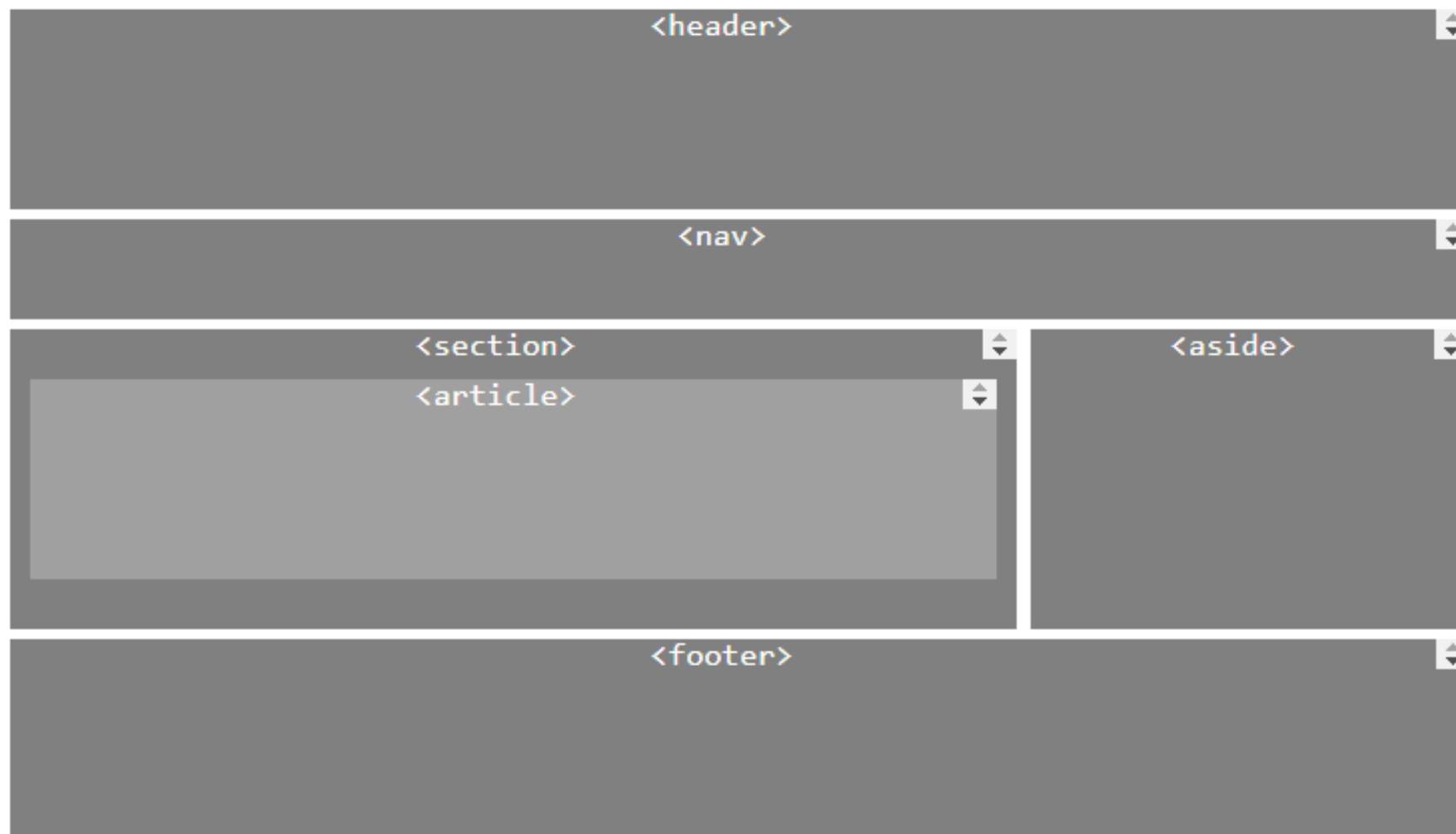
# Struktura strony HTML

Jako pierwszy element w znaczniku `<html>` powinien pojawić się znacznik `<head>`. Znacznik ten służy definicji meta danych strony. `<head>` nie przyjmuje argumentów, lecz umieszcza się w nim znaczniki służące do: nadania tytułu strony internetowej (znacznik `<title>`), określenia strony kodowej (znacznik `<meta>`), dołączenia zewnętrznego kaskadowego arkusza stylów (znacznik `<link>`) oraz do zdefiniowania wewnętrznego arkusza stylów (znacznik `<style>`). W `<head>` można także umieścić skrypt (znacznik `<script>`), lecz uznane za dobrą praktykę jest umieszczanie skryptów jako ostatnie znaczniki w `<body>` - w ten sposób skrypt nie opóźnia załadowania strony internetowej.

# Struktura strony HTML

Znacznikiem, który należy umieścić po `<head>` jest znacznik `<body>`. Jest to znacznik, w którym zostają umieszczone znaczniki umieszczające na stronie elementy (np. `<p>`, `<img>`, itp.). Znacznik przyjmuje atrybuty dotyczące stanu strony.

# Struktura strony HTML



# Style na stronie

## Zewnętrzny arkusz CSS (osobny plik)

Zewnętrzny arkusz CSS dołączamy do dokumentu HTML znacznikiem znajdującym się w sekcji `<head>` podstrony (wartość atrybutu `href` musi wskazywać rzeczywisty, istniejący plik):

```
<link rel="stylesheet" href="arkusz.css">
```

## Znaczniki `<style>` w sekcji head

Kod CSS można osadzić pomiędzy tagami `<style></style>`, koniecznie umieszczonymi wewnątrz sekcji `<head>`:

```
<style>
```

```
    #container {  
        color: red;  
    }
```

```
</style>
```

# Style na stronie

## Style inline (wewnątrz atrybutu znacznika)

Stylizowanie inline (ang. w linii), oznacza zapisanie CSS jako wartości atrybutu style wybranego tagu HTML, znajdującego się w sekcji <body>:

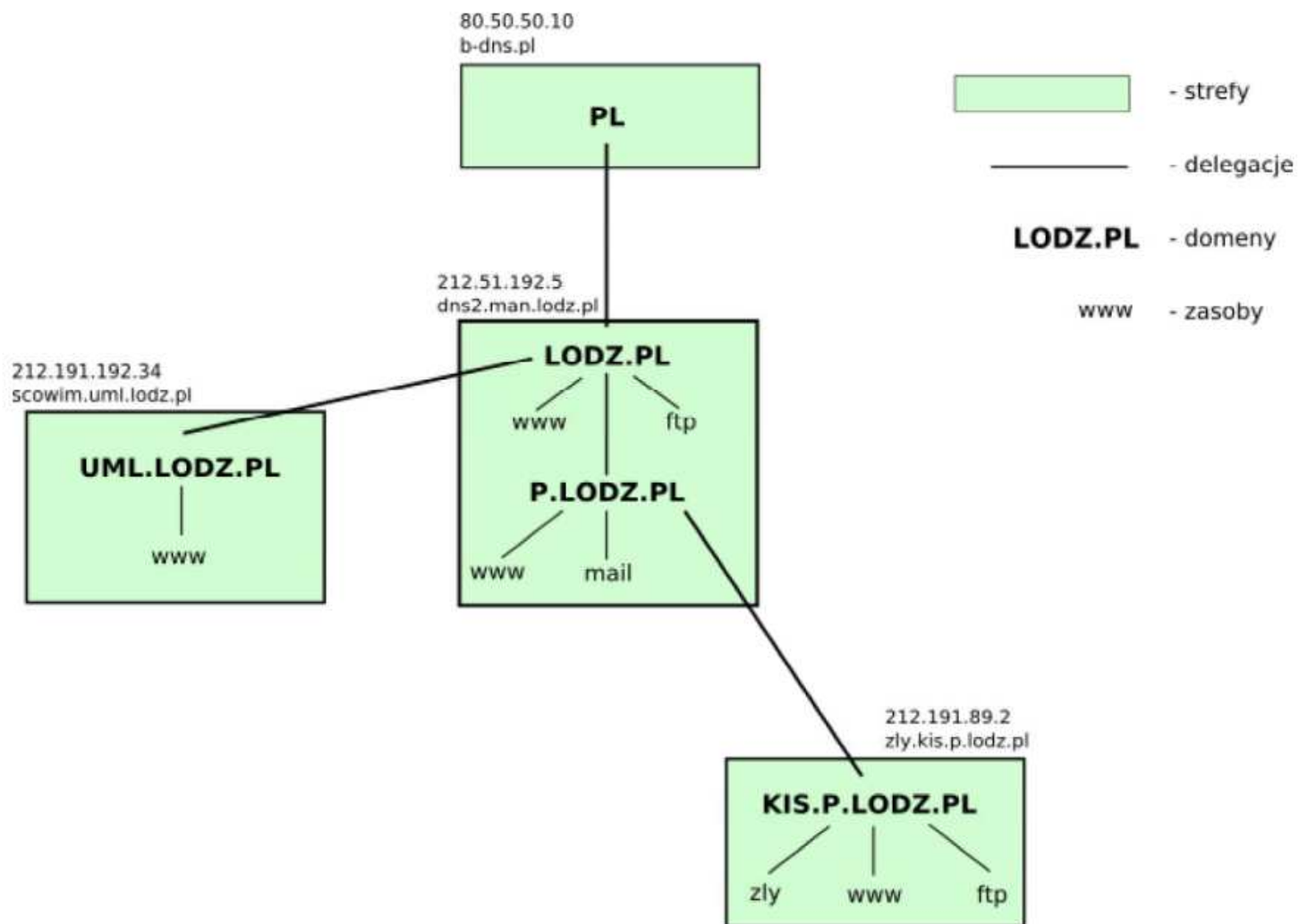
```
<div id="container" style="color:blue;">
```

Jakis tekst.

```
</div>
```

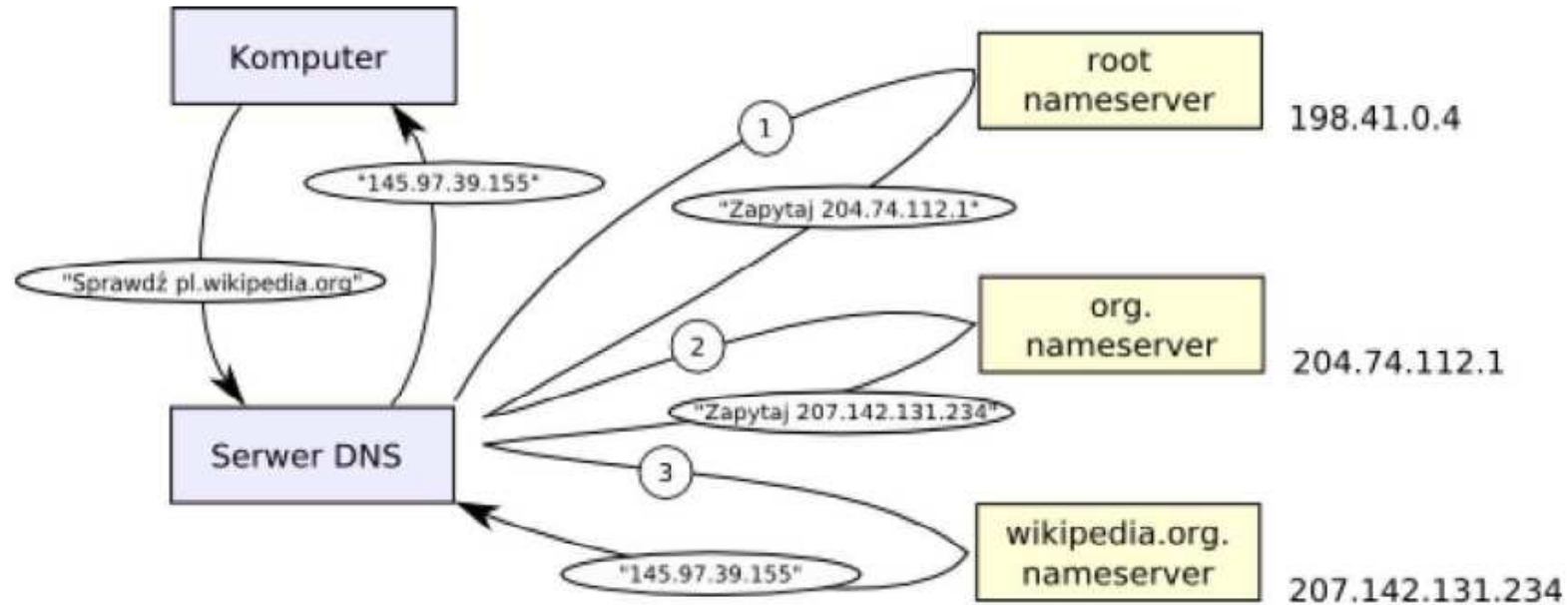


# Strefy i domeny



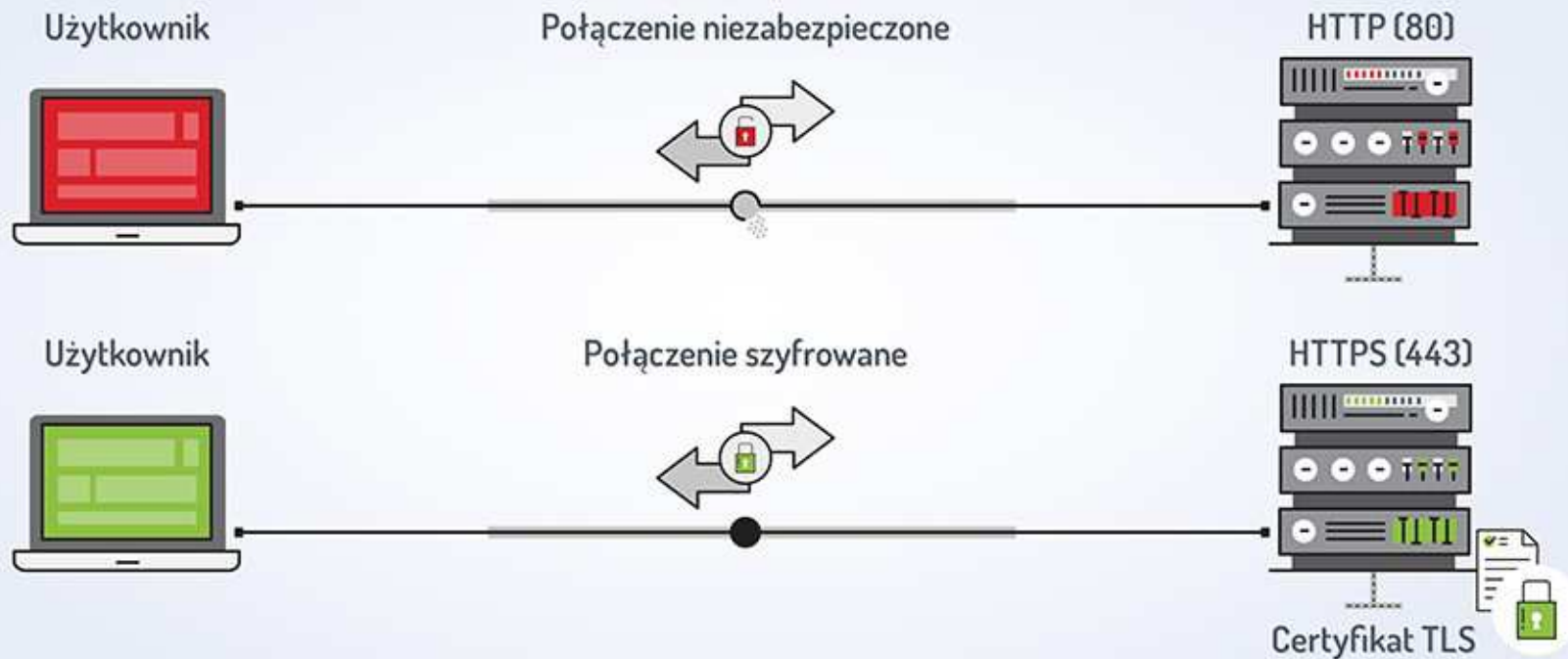
Rys. 4. Strefy i domeny – lodz.pl

# DNS w przypadku strony www



# Protokoły stron www

## HTTP CZY HTTPS?



# Protokoły stron www

- **Protokół http** (z *ang. Hypertext Transfer Protocol*) to protokół przesyłania danych dla sieci internetowej. Komunikacja w ramach protokołu http oparta jest na **wysyłaniu danych pomiędzy klientem i serwerem (odpowiadającym)**. Klient wysyła zapytanie – serwer wysyła odpowiednie dane.
- Protokół http zaliczany jest do **protokołów bezstanowych**, czyli takich, które nie przechowują danych. Z jednej strony pozwala to na zmniejszenie obciążenia serwera danymi, z drugiej może okazać się kłopotliwe przy kilkurazowym korzystaniu ze strony.
- Dlatego do każdego serwisu opartego na protokole http wprowadzany jest mechanizm ciasteczek, które zapamiętują dane sesji na stronie odwiedzanego serwera (dane zapisane dzięki ciasteczkom mogą być później wykorzystywane przez właściciela serwera do różnych celów – np. kampania remarketingowa).

# Protokoły stron www

- **Protokół https** to szyfrowana wersja protokołu http, która **szyfruje połączenie z wykorzystaniem protokołu SSL. Obecnie wykorzystuje się również protokół TLS będący rozwinięciem SSL.**
- Https poprzedza adresy portali związanych z bankowością internetową – banków, kantorów, portali oferujących pożyczki, a także sklepów internetowych (zwłaszcza tych, gdzie jest możliwość płacenia kartą kredytową).
- Coraz częściej na protokół https decydują się też portale gromadzące dane użytkowników – fora, portale społecznościowe i portale z możliwością rejestracji użytkowników (którzy w ten sposób dostają dostęp do możliwości komentowania treści).

# Metody protokoły HTTP

Metoda	Zastosowanie / opis
<b>GET</b>	Pobieranie zasobu lub jego wyświetlenie, np. wyświetlenie formularza lub strony. Parametry można przekazywać jedynie poprzez adres (np. ?nazwa=wartosc&nazwa2=wartosc2)
<b>POST</b>	Przesłanie danych zapisanych jako pary klucz-wartość do serwera (np. wysłanie formularza, gdzie kluczem jest nazwa danego pola a wartością wpisana przez nas wartość). Metoda ta pozwala przesyłać także pliki (a także wiele pliki oraz pary klucz-wartość jednocześnie). Parametry są przekazywane w ciele zapytania, można także przekazywać parametry poprzez adres (tak jak w metodzie GET)
<b>PUT</b>	Przesyłanie 'paczki' danych, np. jednego pliku. Metoda ta ma pewne ograniczenia, np. nie ma możliwości łączenia par klucz-wartość z inną przesyłaną treścią (np. plikiem). Obecnie używana głównie w przypadku RESTowych serwisów, gdzie ciałem jest np. formularz zapisany w postaci JSONa.
<b>DELETE</b>	Usuwanie zasobu na serwerze, z racji bezpieczeństwa praktycznie zawsze jest wyłączona domyślnie. Obecnie używana głównie w przypadku RESTowych serwisów, wskazując, że dany zasób ma być usunięty (i obsługiwany przez aplikację, a nie sam serwer).
<b>HEAD</b>	Analogiczny do zapytania GET, z tym wyjątkiem, że nie zwraca ciała (zawartości). Służy do pobrania metadanych o zasobie w postaci nagłówków HTTP. Dla danego adresu zwraca same nagłówki.

# Statusy HTTP (kody odpowiedzi)

Kody są podzielone na grupy, pierwsza cyfra kodu mówi nam, z której jest on grupy:

1xx — informacyjne, nieczęsto można spotkać, dotyczą bardziej środowiska niż samej aplikacji (np. 111 — serwer odrzucił połączenie)

2xx — zapytanie się powiodło

3xx — przekierowanie, zapytanie należy kierować pod inny adres / serwer

4xx — błąd aplikacji spowodowany działaniem użytkownika (np. wspomniany 404 — nie znaleziono — czy 403 — brak dostępu lub 400 — niepoprawnie zapytanie)

5xx — błąd serwera (np. nieobsłużony wyjątek w Javie)

Listę kodów można znaleźć m.in. na Wikipedii. Są one także opisane i zdefiniowane w dokumencie RFC 2616 z późniejszymi zmianami.

# Bezstanowość HTTP a ciasteczka

Ciasteczka — pierwotnie mające postać plików tekstowych w formacie klucz=wartość, obecnie przechowywane są w wewnętrznej bazie danych przeglądarki.

Formalnie ciasteczka (ang. cookies) to zbiór par klucz-wartość przypisanych do danej domeny (w ogólnym przypadku; możliwe jest też utworzenie cookies dla ścieżki, np.

kobietydokodu.pl/jednaaplikacja) które są **wysyłane do serwera z każdym zapytaniem**. Oczywiście ze względów bezpieczeństwa nie przechowuje się w nich danych użytkownika, do tego służą tzw. sesje, czyli kolekcje danych przechowywane po stronie serwera.

W ciasteczkach najczęściej zapisuje się unikalny identyfikator, na podstawie którego serwer ma możliwość powiązania jednego ze zbiorów danych które przechowuje z wysyłającym zapytanie klientem.



# Inne zastosowanie przeglądarek

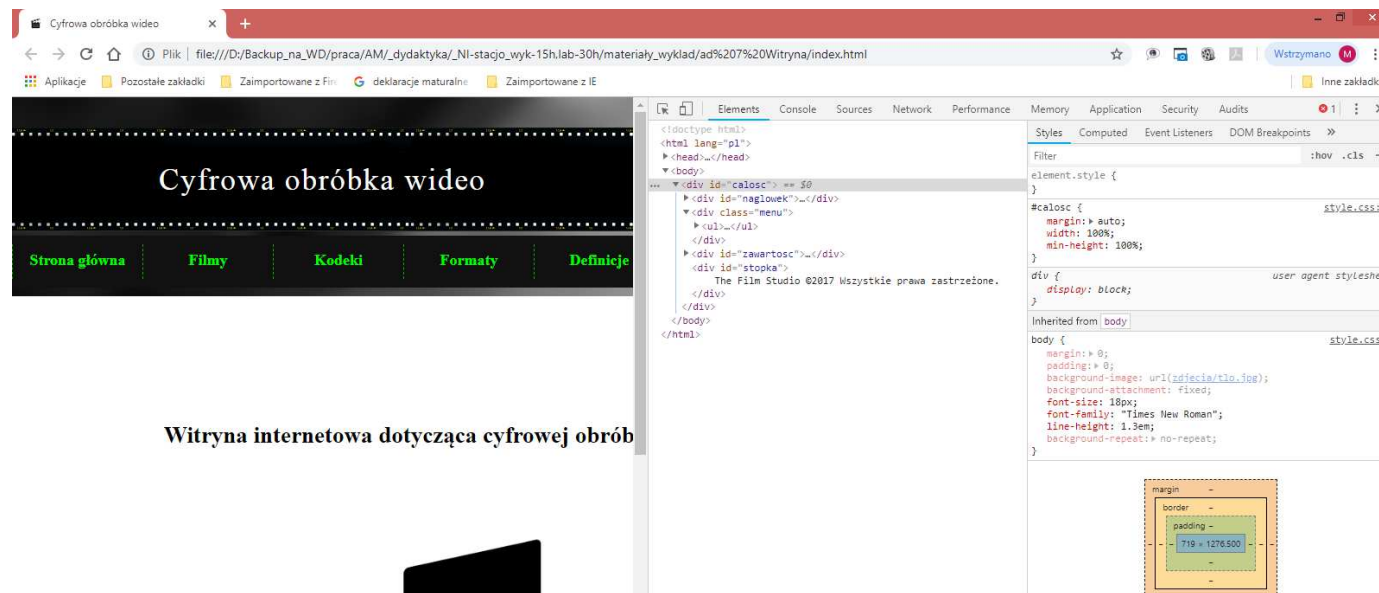
- Interfejs WebGUI dla urządzeń sieciowych, np. <https://www.tp-link.com/us/support/emulators>

[https://emulator.tp-link.com/TL-WR720N\\_V2/index.htm](https://emulator.tp-link.com/TL-WR720N_V2/index.htm)

- Klient ftp, np. <ftp://ftp.rs.internic.net/domain/named.root>

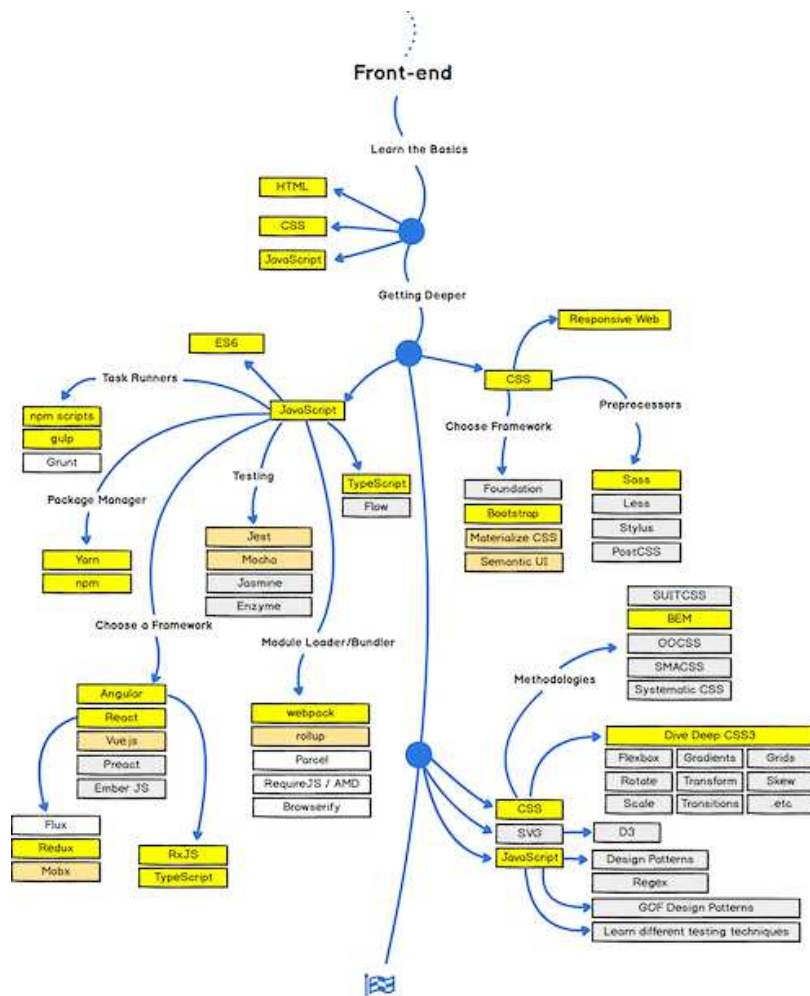
<ftp://ftp.eltrox.pl/FAQ/Kenik/>

- Narzędzie do testowania i walidacji stron www

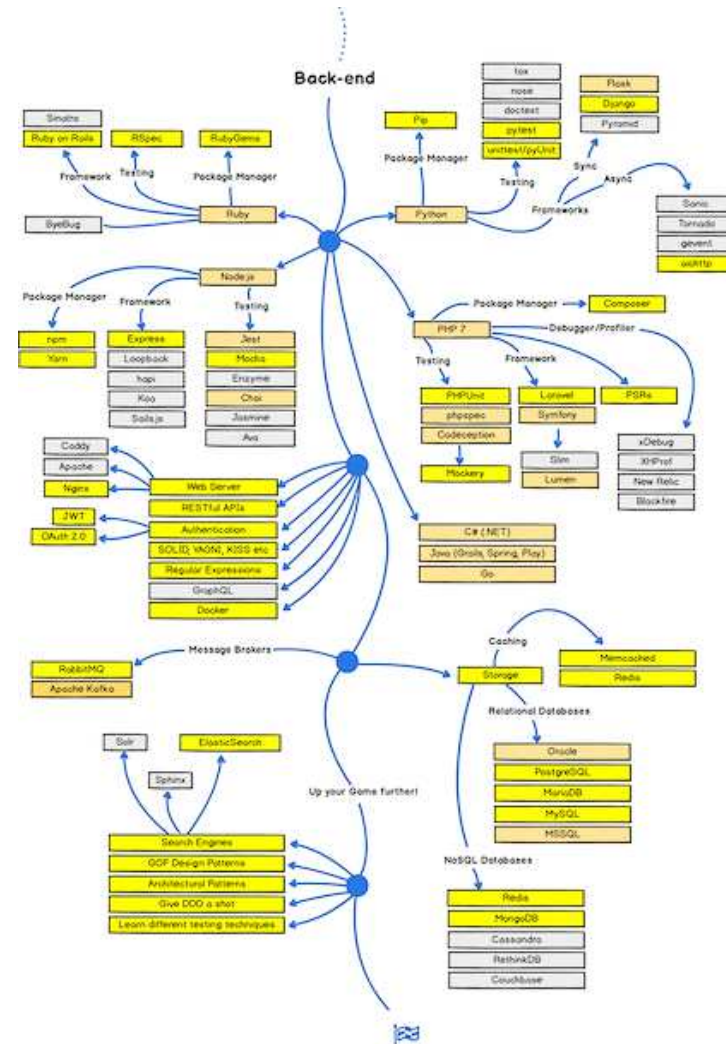


# Umiejętności programisty www

## Niezbędnik frontendowca



## Niezbędnik backendowca



# Umiejętności programisty www

## Niezbędnik frontendowca

Ścieżka frontendowca zakłada znajomość HTML, CSS i JavaScript, przy czym jest to ledwie wierzchołek góry lodowej. Na bazie tych technologii, zbudowano kolejne warstwy abstrakcji upraszczające tworzenie skomplikowanych i dużych projektów. I tak, podstawowa znajomość JavaScript pozwoli Wam na opanowanie jednego z frameworków takich jak Angular czy React (które będą na topie w 2018 i pewnie też w kolejnym roku). Podstawowa znajomość stylu CSS pozwoli Wam opanować jeden z preprocesorów, np. Sass, oraz jeden z frameworków, np. Bootstrap.

# Umiejętności programisty www

## Niezbędnik backendowca

W tym przypadku wybór technologii będzie miał dalej idące konsekwencje, niż w przypadku technologii frontendowych. Trudniej przeskoczyć tu z jednego środowiska do drugiego, jak np. z Reacta do Angulara. Możemy zacząć uczyć się PHP, C# (i generalnie .NET) lub Javy. W tych technologiach działa bardzo wiele istniejących projektów, szczególnie w Polsce. Warto jednak wiedzieć, że od pewnego czasu startupy i nowe zespoły stawiają na inne ścieżki, np. Go, Ruby (na tym działa nasze nowe forum), Python (ten zalicza wielki powrót, pojawi się niebawem jako środowisko skryptowe w Excelu) czy w końcu node.js. Coraz bardziej konieczna staje się znajomość technologii kontenerów, wśród których autor roadmapy wskazuje na Dockera. RESTful APIs stały się kluczowym narzędziem do łączenia różnych klocków w jeden organizm.