

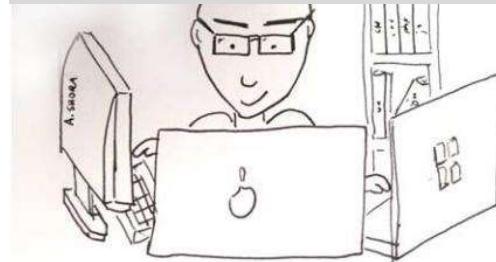
Wprowadzenie do Aplikacji Internetowych (Webowych)

dr inż. Grzegorz Rogus
rogus@agh.edu.pl

1

Tematyka wykładów

Od zera do webDevelopera



“THE FULL STACK DEVELOPER”

2

Plan wykładów

1. Wprowadzenie do aplikacji webowych – omówienie stosu technologicznego
- 2-4 Prezentacja standardów
2. HTML5 + wstęp do CSS
3. CSS + RWD – Zasady projektowania aplikacji webowych
4. Nowoczesny JS
ECMS7 (2016), TypeScript

Plan wykładów cd

5-6. Frontend:



7 . Backend:



express
web application
framework for
node



Autentykacja

Wprowadzenie do Aplikacje Webowych

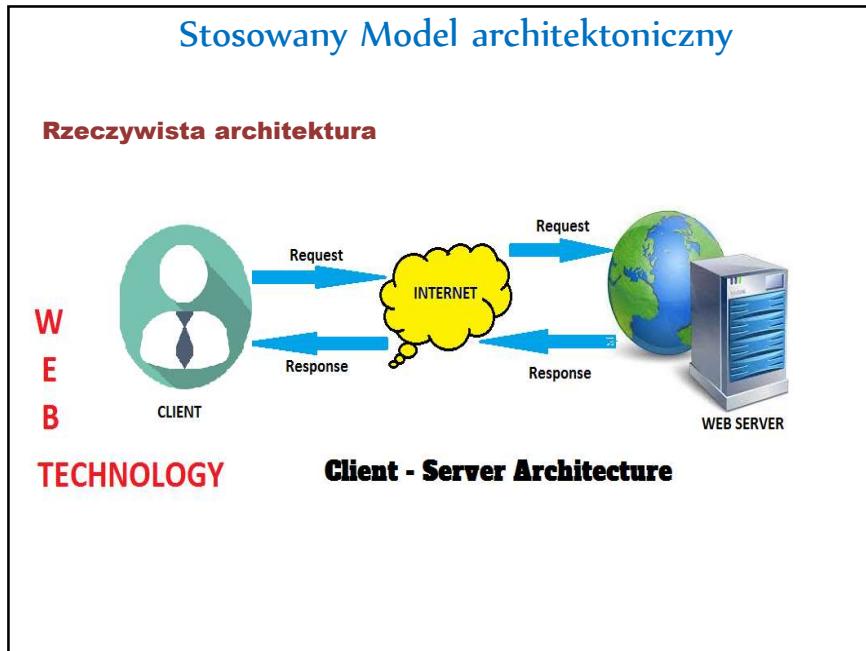
Koncepcja
Architektura
Technologie

Czym są Aplikacje Webowe

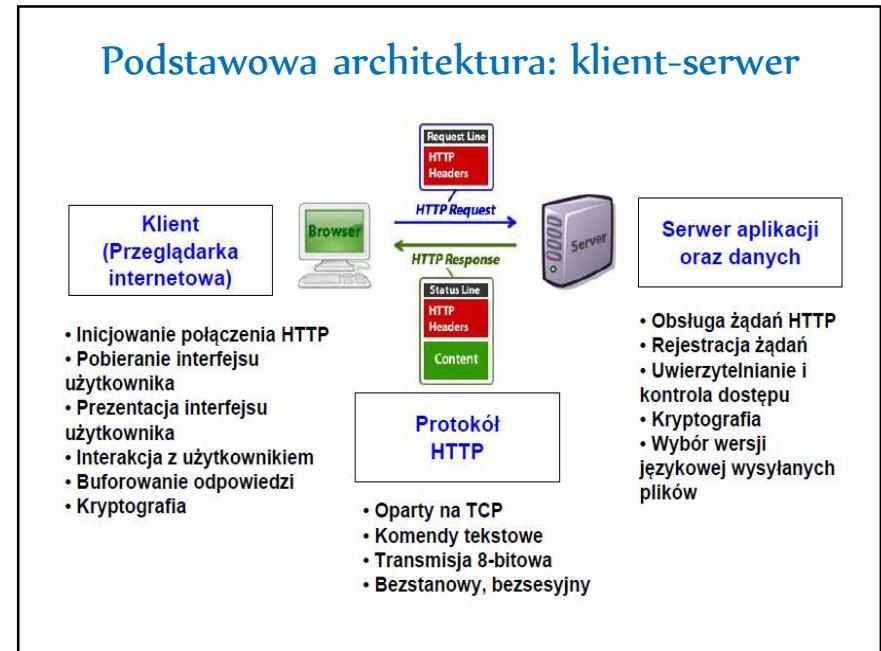
Perspektywa użytkownika końcowego



[Aplikacja webowa](#) (web application) to aplikacja uruchamiana w przeglądarce, która przez dostarczony interfejs ma dostarczać użytkownikowi jakąś konkretną usługę. Takie aplikacje w znakomitej większości komunikują się z głównym serwerem, by móc serwować użytkownikowi treści i reagować na jego akcje.



7

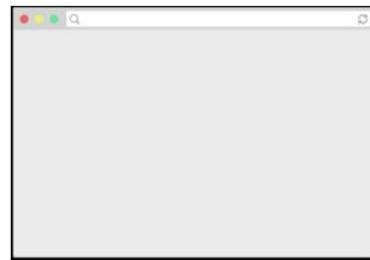


8

Klient HTTP

Przeglądarka – http client

Przeglądarka to aplikacja której zadaniem jest wyświetlenie zawartości stron internetowych lub w przypadku aplikacji webowej warstwy prezentacyjnej

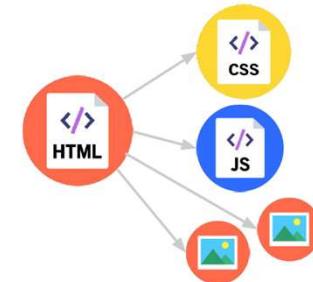
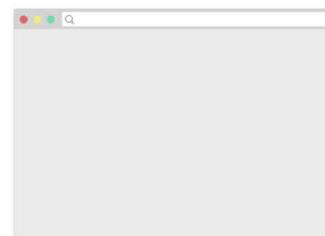


Odpowiada za:

- Wysyłanie żądań pobrania dokumentów
- Wizualizacje pobranych dokumentów
- Obsługę interakcji z użytkownikiem końcowym

Strony internetowe

Strony/widok są napisane w języku znaczników zwanym HTML, więc przeglądarki wyświetlają stronę internetową, czytając i interpretując jej kod HTML.

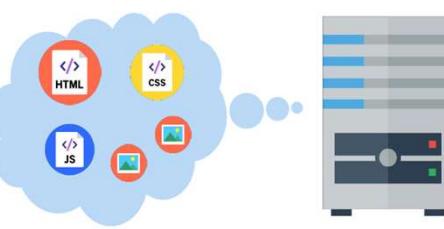


Plik HTML może zawierać linki do innych zasobów, takich jak obrazy, filmy, a także pliki **JavaScript** i **CSS** (arkusz stylów), które przeglądarka również ładuje

Serwer HTTP

Web Server

Serwer WWW to program działający na komputerze, który dostarcza strony internetowe w odpowiedzi na żądania. Przechowuje lub generuje zwróconą stronę internetową.



Web Server Type	Percentage
Apache	36.1%
Nginx	32.6%
Cloudflare Server	16.1%
Microsoft-IIS	7.7%
LiteSpeed	7.3%

Legend: Overall (Grey), top 1,000,000 (Green), top 100,000 (Red), top 10,000 (Blue), top 1,000 (Yellow)

Percentages of websites using the selected web servers broken down by ranking
Note: a website may use more than one web server

11

Serwer HTTP

Serwer HTTP - serwer WWW - program nieprzerwanie pracujący, obsługujący repozytorium dokumentów (np. HTML), które udostępnia sieciowym klientom HTTP.

Do zadań serwera HTTP należy:

- obsługa żądań HTTP i ich rejestracja w plikach dziennika (log files),
- uwierzytelnianie i kontrola dostępu użytkowników końcowych za pomocą nazwy i hasła,
- kryptograficzne szyfrowanie komunikacji sieciowej z klientem http,
- automatyczny wybór odpowiedniej wersji językowej dokumentu.

12

Protokół Http

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

- Oparty na TCP,
- Komendy tekstowe,
- Transmisja 8-bitowa,
- Bezstanowy i bezsesyjny.

RFC 2616 (czerwiec 1999)
RFC (Request for Comments)

Bezstanowość (ang. stateless), co oznacza że nigdzie nie istnieje zapis stanu poprzednio wykonanych operacji, a kolejne transakcje są wykonywane niezależnie.

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.**

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.

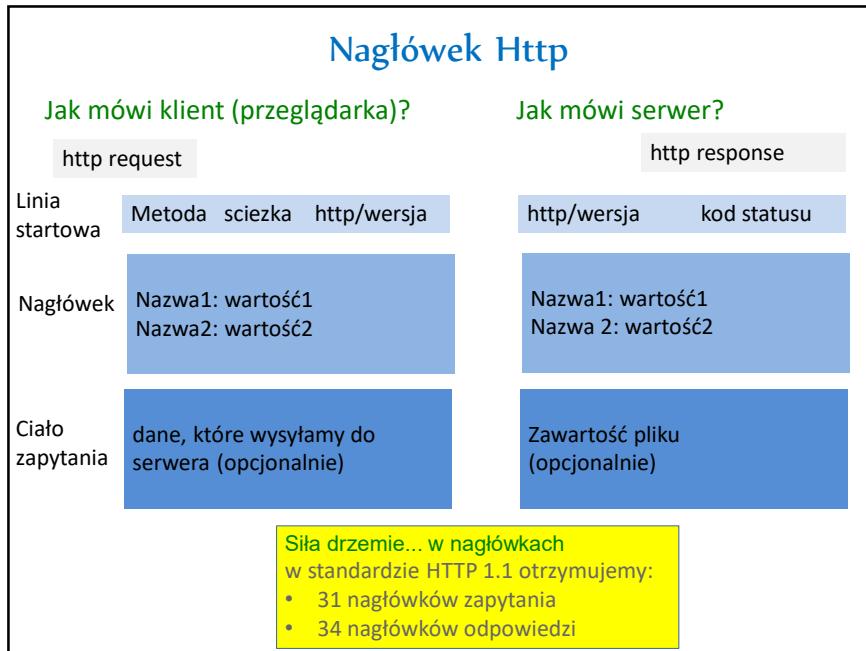
Metody w protokole Http

HTTP Methods (Verbs)

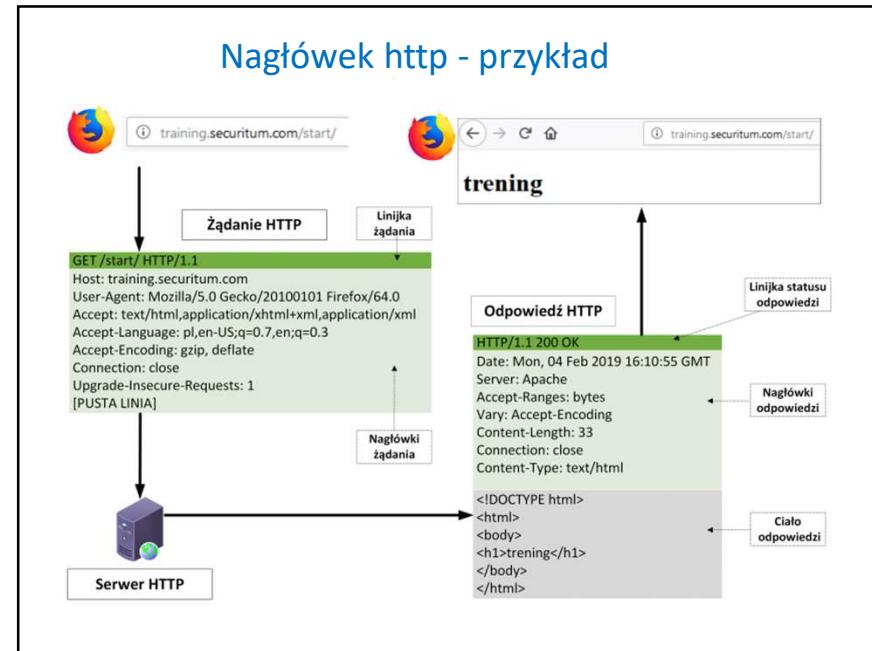
- GET – odczyt danych (w formularzach metoda wysyłki danych do URL – dane zapisane w adresie)
- HEAD – odczyt danych na temat adresu URL
- PUT – zapis do zasobu pod URL
- POST – wysłanie danych pod adres URL oraz otrzymanie informacji zwrotnej
- DELETE - Delete a URL

GET oraz POST (formularze) są najczęściej używane.

REST APIs używa GET, PUT, POST, and DELETE



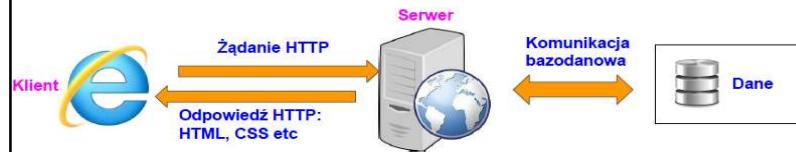
15



16

Podstawowa architektura: klient-serwer

Klasyczny model komunikacji synchronicznej HTTP:



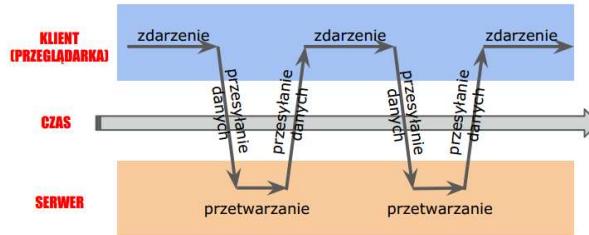
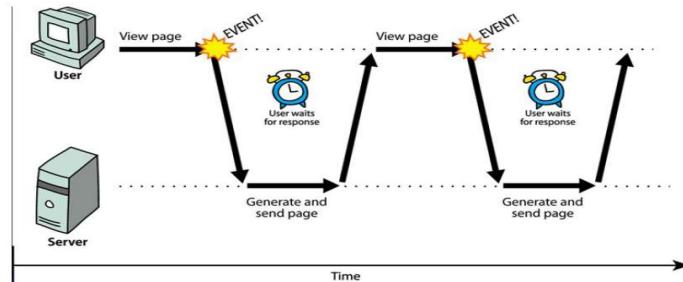
Jest to model synchronicznej komunikacji HTTP, gdzie klient wysyła żądanie, serwer je odbiera następnie przetwarza (np. komunikuje się z bazą danych) i na końcu generuje odpowiedź.

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 (pobrać) po każdym żądaniu klienta, jeżeli strony są złożone to proces ten jest długi.

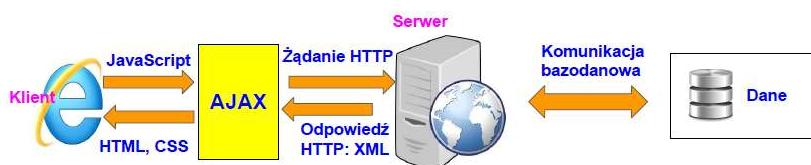
Podstawowa architektura: klient-serwer

Klasyczny model komunikacji synchronicznej HTTP:



Podstawowa architektura: klient-serwer

Model komunikacji asynchronicznej HTTP:

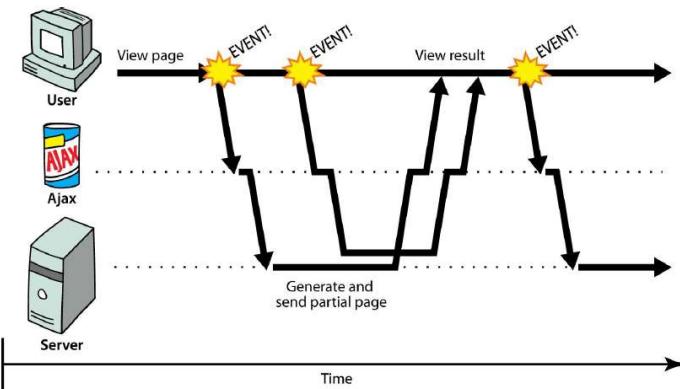


Jest to model asynchronicznej komunikacji HTTP, gdzie klient (przeglądarka) nie czeka na przyjście odpowiedzi na żądanie serwera, a wykonuje dalsze żądania. Pośredniczy w tym wbudowany w przeglądarkę mechanizm silnika AJAX.

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

Podstawowa architektura: klient-serwer

Model komunikacji asynchronicznej HTTP:



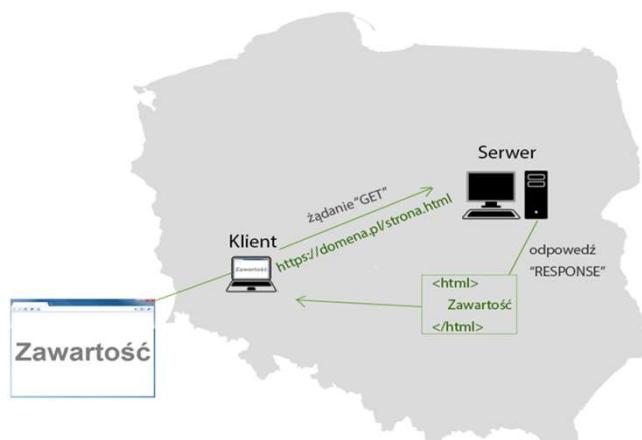
Typy aplikacji webowych

4 główne typy aplikacji :

1. O statycznej treści (to jest strona internetowa a nie web application!!!!)
2. Server-side rendering:
Serwer wysyła nową stronę HTML dla każdej unikalnej ścieżki
(może to strony internetowe, ale często jest to aplikacja internetowa)
3. Single-page application
4. Aplikacje progresywne

Od strony WWW do aplikacji internetowej

Jak działają strony internetowe?



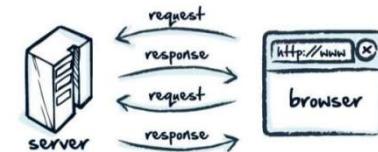
23

Na początku była strona statyczna

Tradycyjna strona internetowa



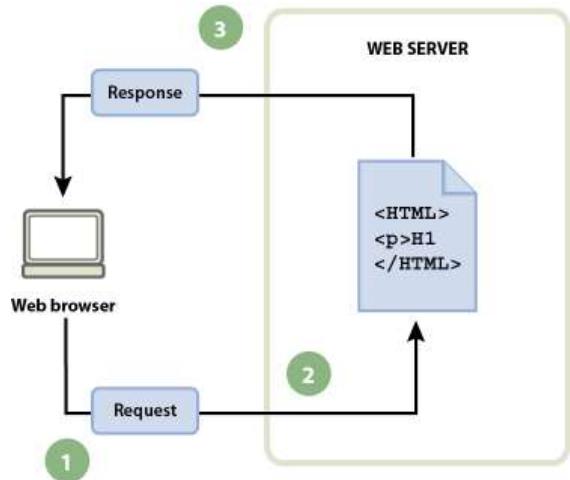
- Korzysta z HTML5, CSS3, JavaScript
- Bazuje na klasycznej architekturze klient-serwer
- Zawiera wiele stron



Strona WWW to dokument HTML udostępniany w Internecie lub lokalnej sieci komputerowej przez tzw. serwer WWW.

24

Proces dostarczenia statycznej strony - podsumowanie

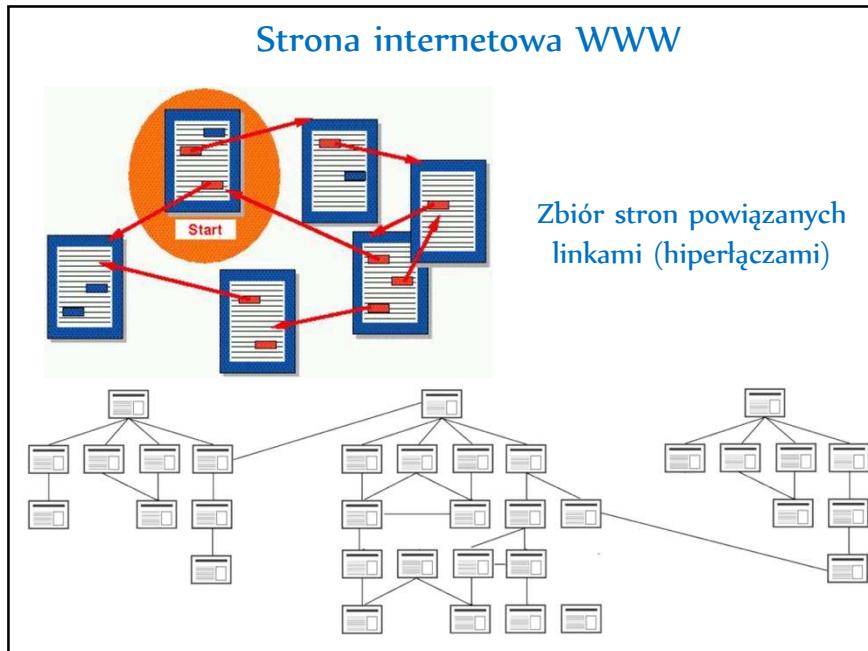


25

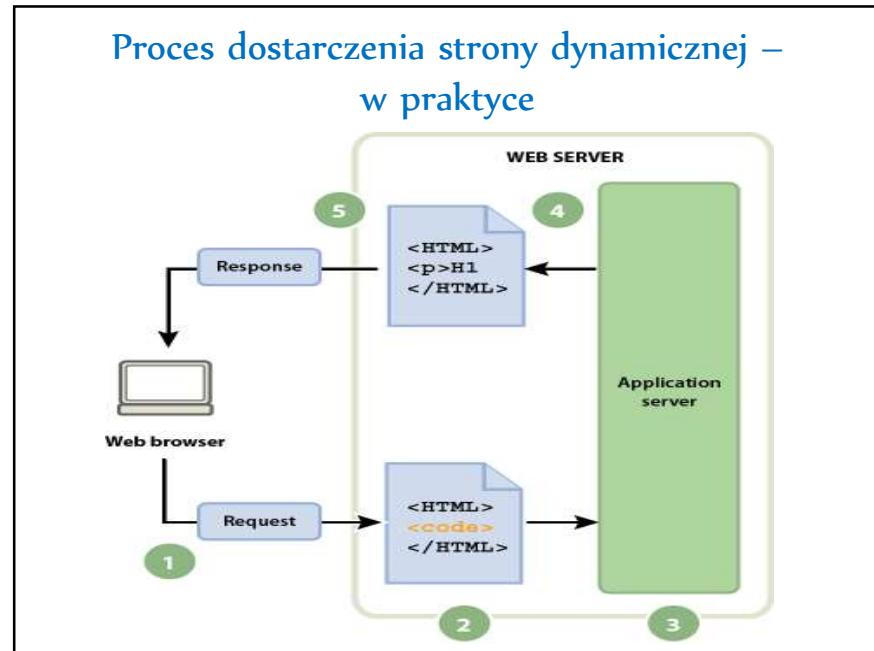
Strony statyczne



26

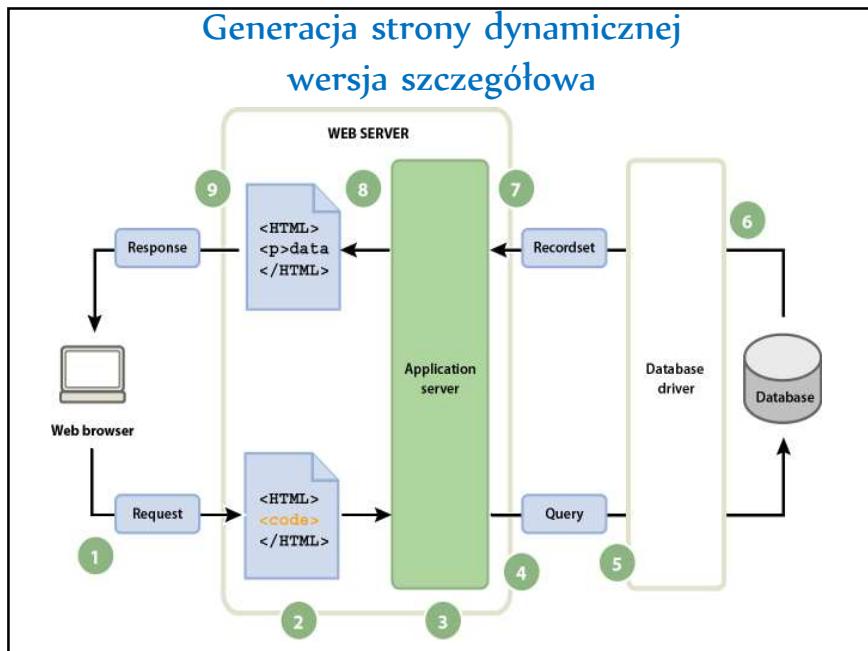


27



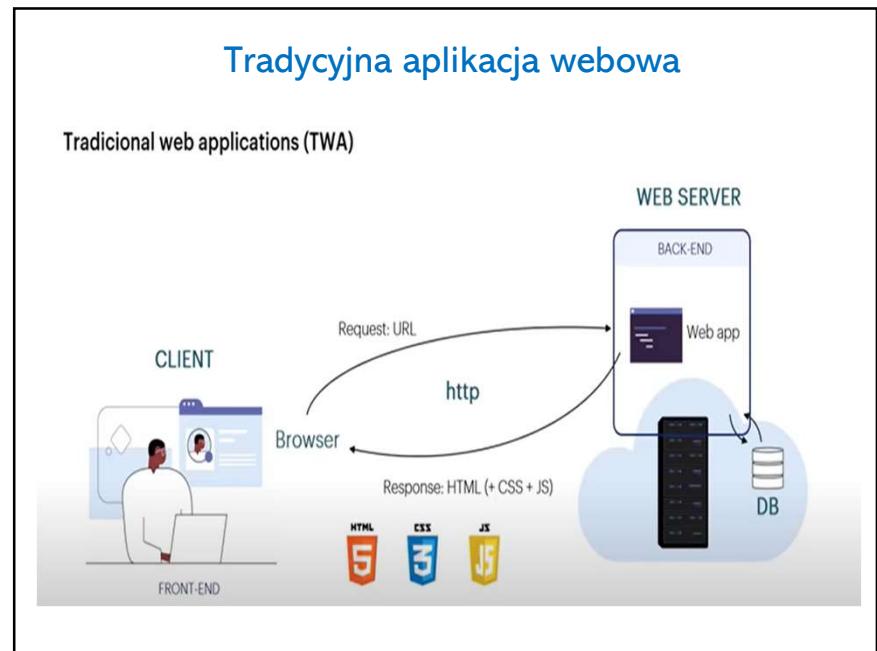
28

Generacja strony dynamicznej wersja szczegółowa

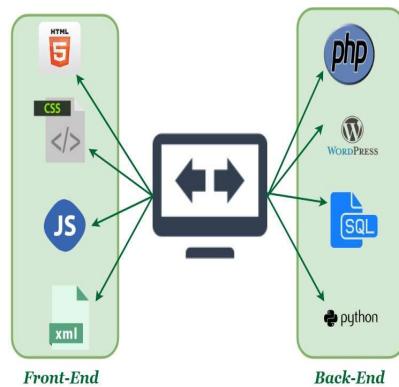


Tradycyjna aplikacja webowa

Tradicional web applications (TWA)



Technologie Web

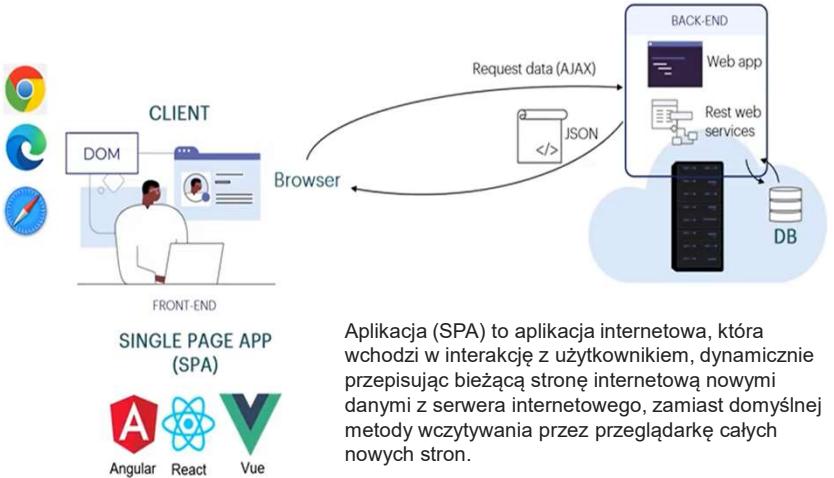


FRONT-END - pojęciowo odnosi się w technologiiach internetowych do kodu wykonywanego po stronie użytkownika. W ogólności do tej kategorii można zaliczyć HTML, CSS oraz JavaScript.

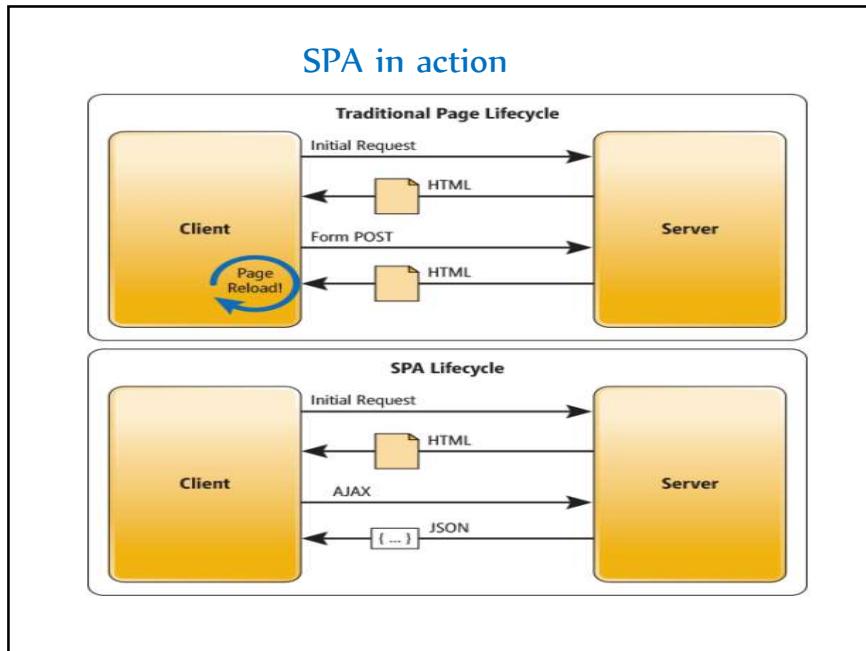
BACK-END - pojęciowo odnosi się w technologiiach internetowych do kodu wykonywanego po stronie serwera. W ogólności do tej kategorii można zaliczyć PHP, Perl, CGI, Ruby, Java, C#, itp.

Nowoczesne aplikacje webowe

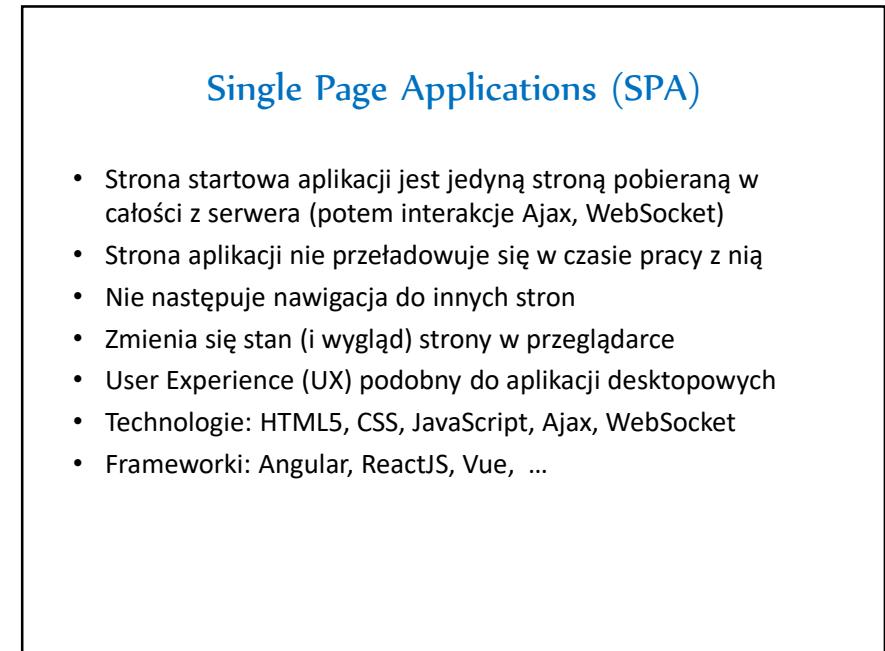
Single page applications (SPAs)



Aplikacja (SPA) to aplikacja internetowa, która wchodzi w interakcję z użytkownikiem, dynamicznie przepisując bieżącą stronę internetową nowymi danymi z serwera internetowego, zamiast domyślnej metody wczytywania przez przeglądarkę całych nowych stron.

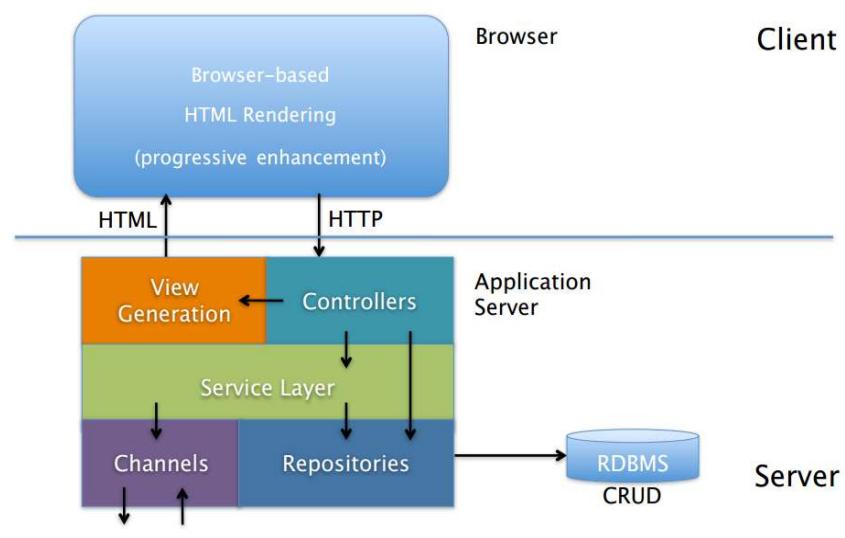


33

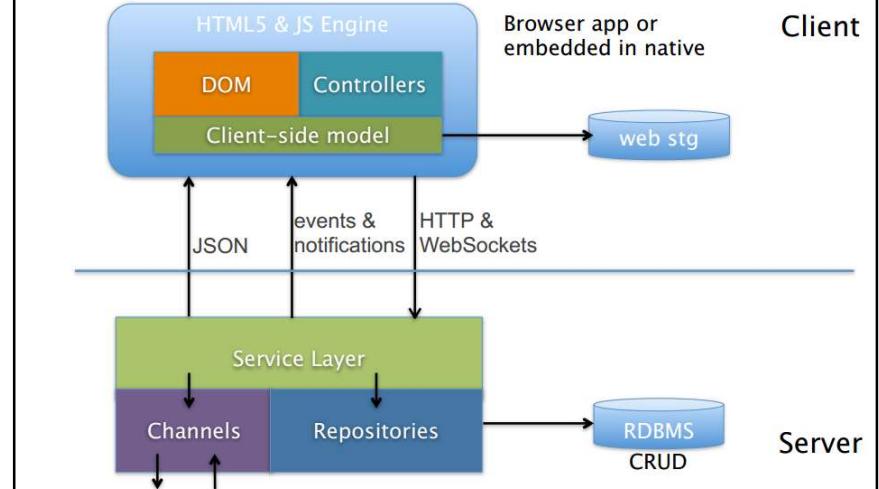


34

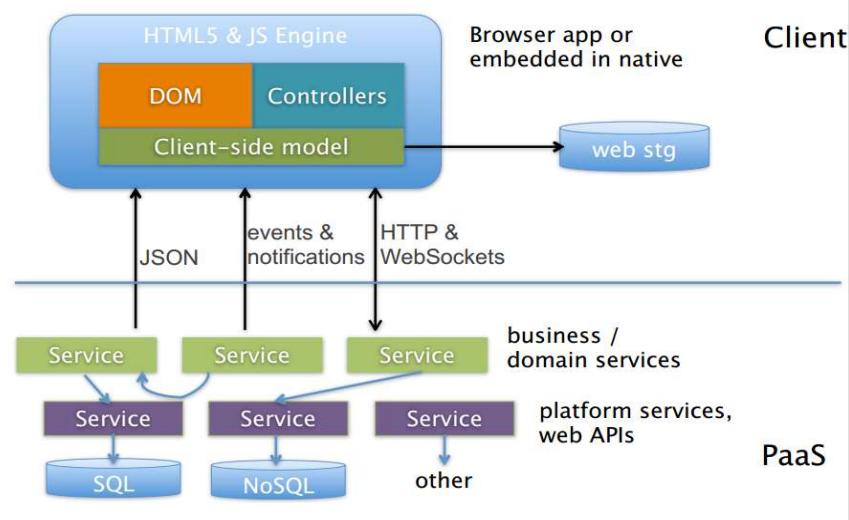
Od server-side app w kierunku inteligentnych klientów i serwisów



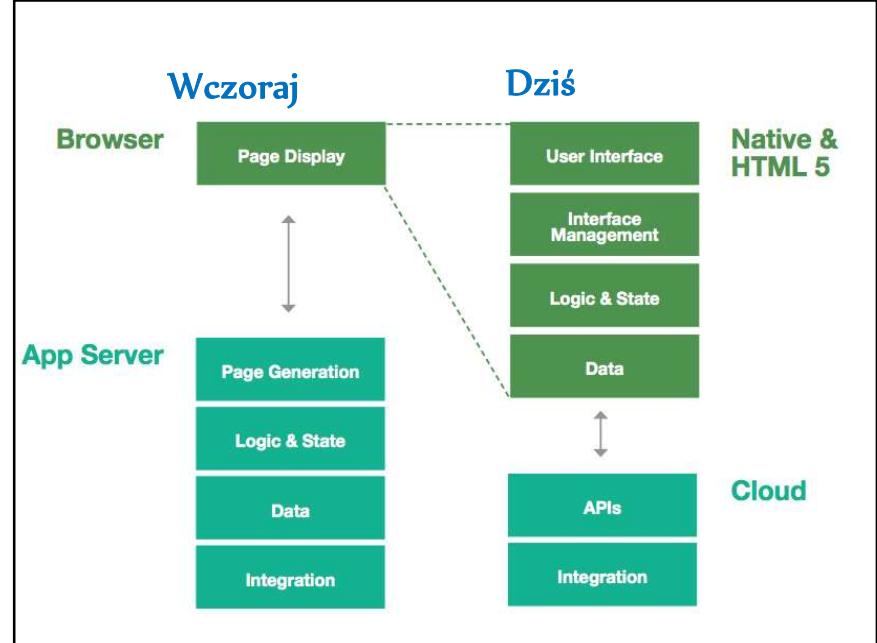
Od server-side app w kierunku inteligentnych klientów i serwisów



Od server-side app w kierunku inteligentnych klientów i serwisów



37



38

Zmiany w tworzeniu aplikacji webowych

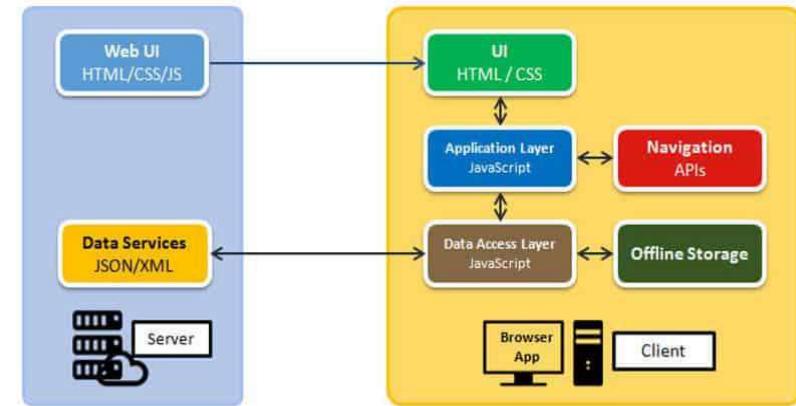
Warunki:

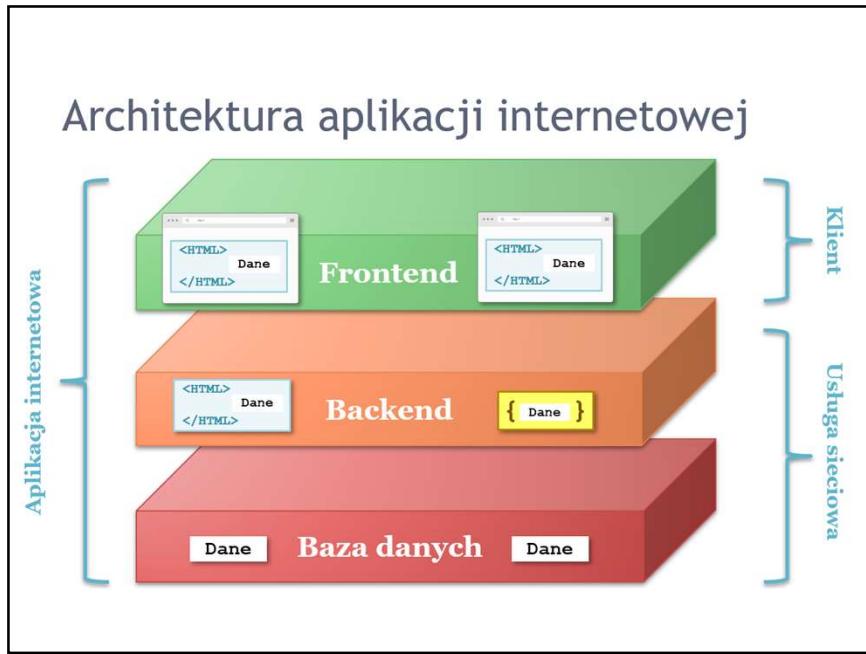
- nowoczesne przeglądarki,
- rozwój języka JavaScript,
- większy nacisk położony na wygodę użycia.

Zmiany w aplikacjach internetowych:

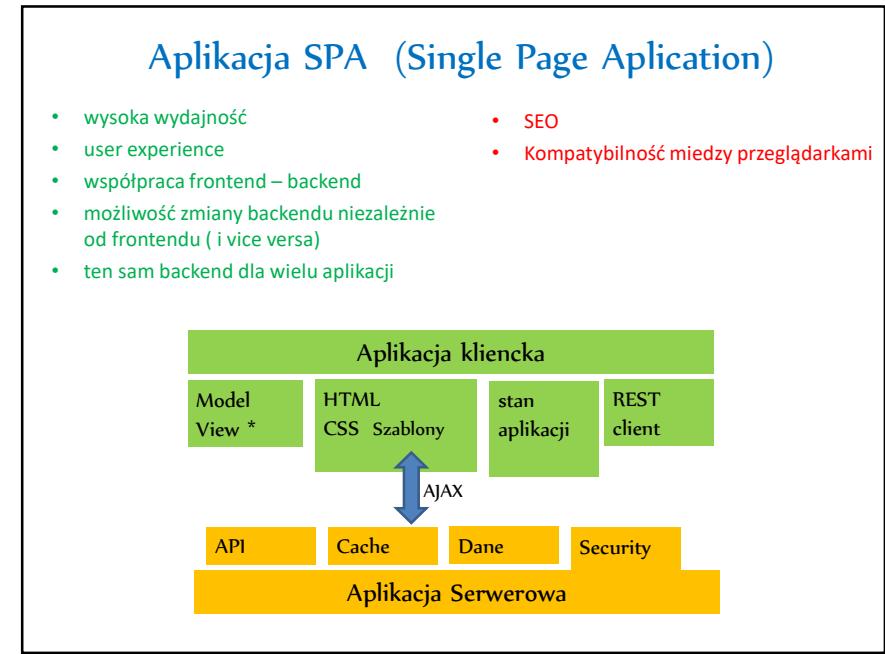
- z punktu widzenia użytkownika działają jak aplikacje desktopowe,
- szybki i dużo bardziej interaktywny interfejs,
- potrafią działać nawet offline,
- działają na wielu platformach (RWD).

Architektura SPA





41



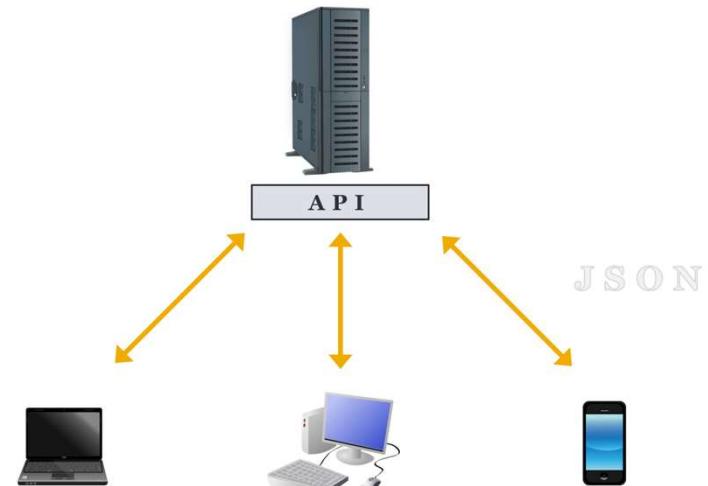
42

Nowoczesna architektura aplikacji Webowych - podsumowanie

- jeden backend wspólny dla wielu frontendów/klientów
 - Aplikacja HTML-owa
 - Natwna aplikacja mobilna
 - Inna aplikacja korzystająca z API
- backend nie generuje żadnego HTML-a (tylko przy inicjalizacji)
- odpowiada jedynie na żądania HTTP
- na wyjściu generuje XML lub JSON
- frontend HTML-owy napisany w JavaScript
- komunikacja z backendem z wykorzystaniem AJAX
- komunikacja fronten – backend z wykorzystaniem RestAPI (coraz częściej GraphQL)

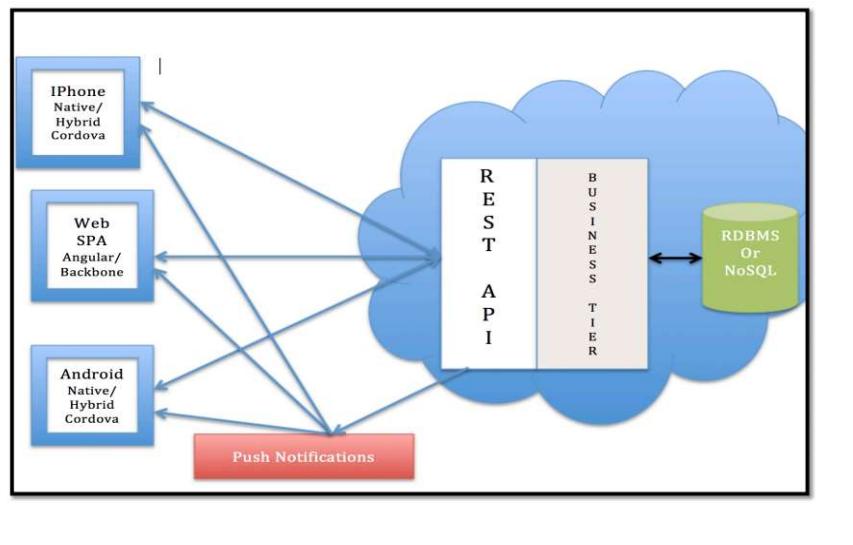
43

Nowoczesna architektura Web



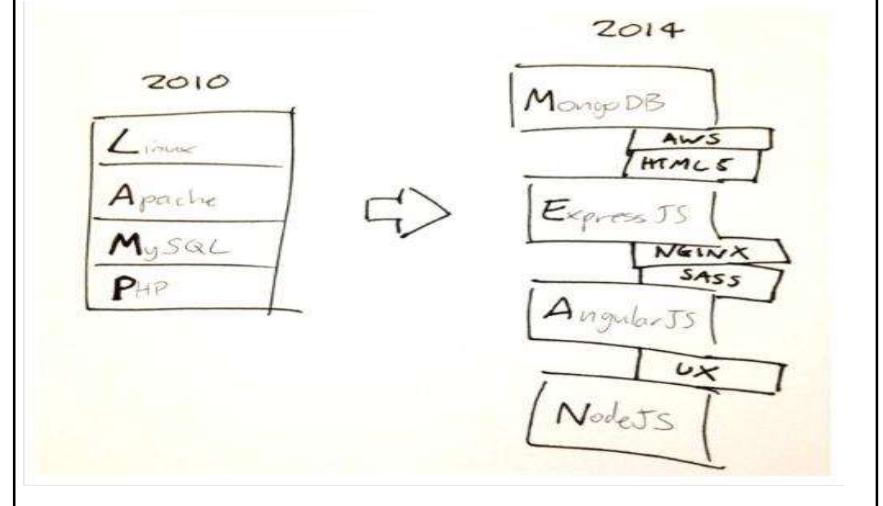
44

Nowoczesna architektura Web



45

Stos technologiczny – kiedyś i dziś

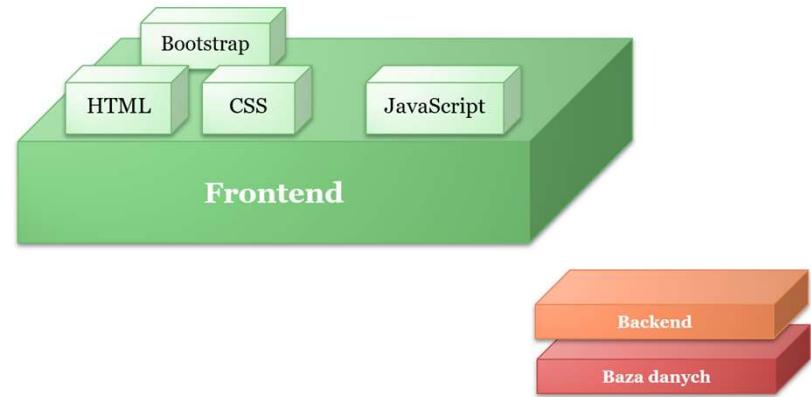


46

TECHNOLOGIE FRONT-END

47

Technologie frontendowe

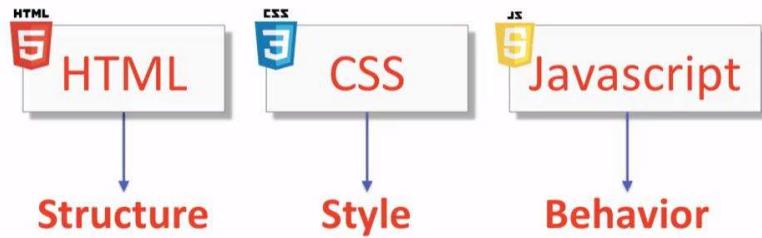


48

Podstawy FRONT-END –czyli MUST HAVE

Podstawowe umiejętności:

- HTML 5;
- Kaskadowe Arkusze Styli CSS;
- Język JavaScript;



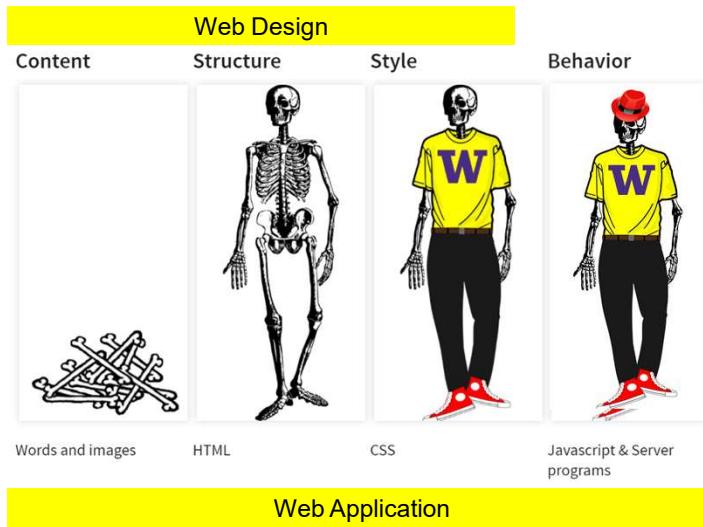
49

Brygada budowlana



50

Stos technologiczny front-enu frontend triad



51

JĘZYK ZNACZNIKÓW HTML

52

JĘZYK HTML

(*HyperText Markup Language, hipertekstowy język znaczników*)

Język używany do tworzenia stron internetowych

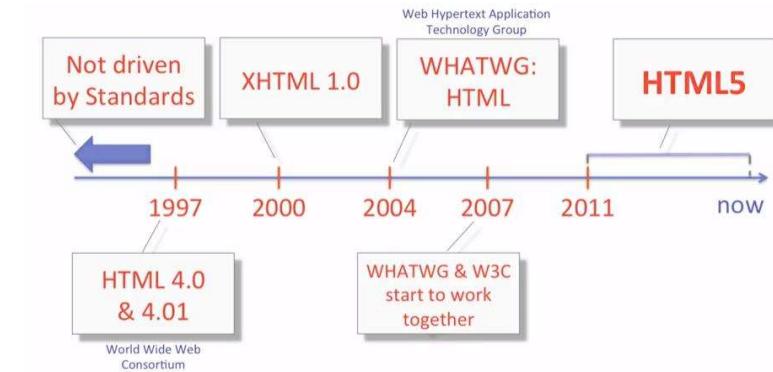
Język znaczników, które umożliwiają nam określenie ról naszych elementów na stronie. Możemy za jego pomocą przekazać, że dany element będzie nagłówkiem, akapitem bądź obrazkiem.

Jest niezależny od systemu operacyjnego i parametrów sprzętowych komputera na którym strona jest wyświetlana

Właściwie możemy myśleć o naszych elementach strony jak o *klockach lego*, z których będziemy budować. Każdy taki "klocek" musi być zapisany w specjalny sposób - za pomocą **tagów HTML**.

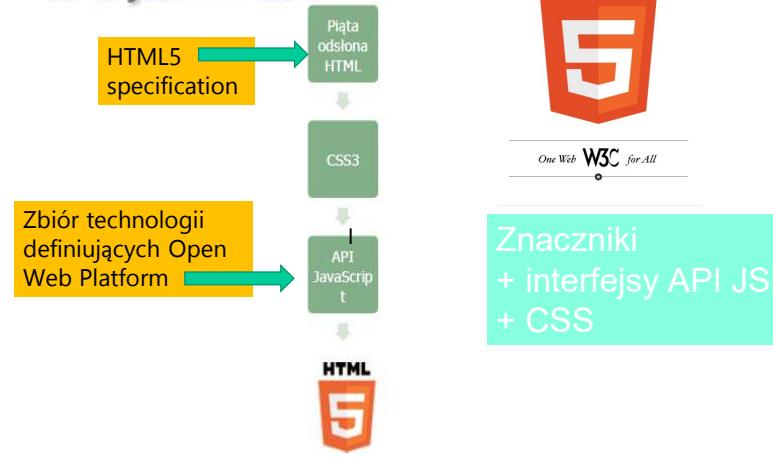
53

Ewolucja



54

Co to jest HTML5 ?



tworzenie serwisów i aplikacji które do tej pory były zarezerwowane tylko dla aplikacji desktopowych

55

HTML 5 — czego dotyczy

- ⌚ Przechowywanie danych
- ⌘ Semantyka HTML 5
- ⌚ Dostęp do systemu plików urządzeń
- ↗ Łączność
- ✿ Wydajność i integracja
- ⌚ 3d, grafika, efekty
- ⌚ Multimedia
- ⌚ CSS 3

56

Teraz kilka słów o składni i semantyce HTML

Na początek powrót do postaci klasycznej

KOMPONENTY JĘZYKA HTML

Język HTML składa się z kilku kluczowych komponentów:

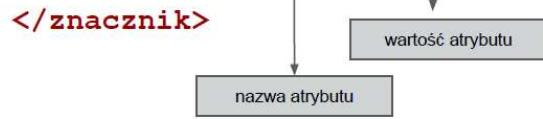
- deklaracji typu dokumentu,
- znaczników,
- atrybutów,
- typów danych,
- referencji znakowych i odwołań w postaci encji.

Podstawy składni ZNACZNIKI (TAGI)

<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
```



Uwaga:

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

ATRYBUTY SPECJALNE

Atrybutu **title** używa się w celu dodania do elementu informacji tekstowej.

```
<span title="Hypertext Markup Language">HTML</span>
```

Atrybut **id** nadaje elementowi unikalny identyfikator.

```
<div id="site-header">...</div>
```

Atrybut **class** służy do sklasyfikowania elementów dla celów prezentacyjnych.

```
<span class="title">Tytuł w tekście</span>
```

TYPY DANYCH

Typy danych wprowadzane jako wartości atrybutów:

- skrypty (script data) <script type="text/javascript">
- dane arkuszy stylów
- identyfikatory i nazwy <div id="siteId"><form name="kontakt">
- adresy URI
- liczby i jednostki <div width="200" > <div width="200px">
- języki i kodowanie znaków <html lang="en"><meta charset="utf-8">
- deskryptory mediów <object data="helloworld.swf">
- kolory

ZNACZNIKI (TAGI)

<p/>

<p></p>



Niektóre znaczniki nie posiadają atrybutów:
 <hr>

a także znacznika domykającego:

<meta charset="UTF-8">
<link rel="stylesheet" href="style.css">
<input type="text" name="imie" value="Grzegorz">.

TYPY ZNACZNIKÓW HTML

- **Znaczniki strukturalne** – opisują logiczną strukturę dokumentu,
- **Znaczniki prezentacyjne** – opisują wygląd poszczególnych elementów (stopniowo usuwane),
- **Znaczniki hipertekstowe** – zawierają linki do innych dokumentów lub innych części tego samego dokumentu.

63

Znaczniki prezentacyjne to już historia

Semantyka: Redukcja znaczników

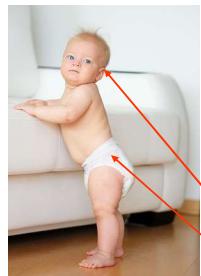
- <applet>
- <big>
- <center>
-
- <frame>
- <frameset>
- <strike>
- ...

64

Zaczynamy projekt strony



Strona internetowa jest jak dziecko: trzeba ją zaplanować i zrobić.



U dziecka są raczki, nóżki, głowa, tułów itp.

Strona potrzebuje tylko głowy (head) oraz ciała (body)

DOKUMENT HTML

Prawidłowo skonstruowany dokument HTML składa się z dwóch części:

- **sekcja head** - zawiera podstawowe informacje o dokumencie, w tym znaczniki META, służące do identyfikacji witryny internetowej przez przeglądarki oraz wyszukiwarki; tu znajduje się informacja o kodowaniu, o zewnętrznych plikach i bibliotekach, które chcemy na stronie wykorzystać.
- **sekcja body** – zawartość dokumentu HTML wyświetlana w oknie przeglądarki internetowej. To, co widzi odwiedzający witrynę internetową.

Podstawowa struktura dokumentu

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

67

Anatomia dokumentu HTML

```

<html>
<head>
    ...
</head>
<body>
    <header>
        <h1> Liderzy Europy </h1>
        <h2> dobrobytu </h2>
    </header>
    <section>
        <p> Trojka wspaniałych: </p>
        <ul>
            <li> Albania </li>
            <li> Moldawia </li>
            <li> Ukraina </li>
        </ul>
    </section>
</body>
</html>

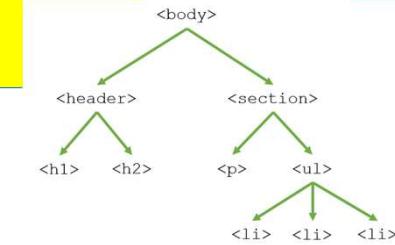
```

68

Liderzy Europy dobrobytu

Trojka wspaniałych:

- Albania
- Moldawia
- Ukraina



DEKLARACJA TYPU DOKUMENTU (DOCTYPE)

Służy do celów poprawnej walidacji utworzonej strony. DTD, do którego odwołuje się sekcja DOCTYPE, zawiera gramatykę i zasady wykorzystywania znaczników w dokumencie. Większość graficznych przeglądarek korzysta z sekcji DOCTYPE, a także innych dodatkowych informacji, warunkując tryb renderowania danej strony.

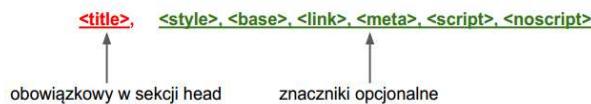
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html>
```

Deklaracja DOCTYPE mówi nam o wersji i typie dokumentu HTML.
Nie ma znaczenia z punktu widzenia treści, jednak jest bardzo ważna dla przeglądarki interpretującej nasz kod

Nagłówek <head> ... </head>

Znacznik head otacza nagłówek dokumentu HTML. Zawiera informacje o tytule dokumentu, meta-informacje, a w większości przypadków także dołączone skrypty. W ramach tego znacznika należy zagnieździć znacznik <title>, który zawiera tekst, wyświetlany przez przeglądarkę jako tytuł strony. Następujące znaczniki mogą znaleźć się w sekcji nagłówka:



Przykład:

```
<head>  
  <link rel="stylesheet" type="text/css" href="styles.css" />  
  <title>Tytuł dokumentu</title>  
  <meta name="description" content="Przykładowa strona" />  
  <meta name="keywords" content="przykład, nagłówek, strona" />  
  <script type="text/javascript" src="skrypt.js">  
  </script>  
</head>
```

Dodawanie arkusza stylu i skryptów

Style CSS:

```
<head>
  <link href="styl.css" rel="stylesheet">
</head>
```

W porównaniu z
HTML4 i XHTML
nie ma atrybutu
type="text/css"

Skrypty:

```
<head>
  <script src="skrypt.js"></script>
</head>
```

Uwaga:
Ten znacznik musi
być zapisany
dokładnie w ten
sposób, a nie
<script/>

71

Kodowanie znaków i język strony

W HTML5 kodowanie znaków jest bardzo ułatwione i sprowadza się do jednej dyrektywy meta w części head naszej strony internetowej:

Kodowanie:

```
<head>
  <meta charset="utf-8">
  <title> Gżegżolka </title>
</head>
```

Dodatkowo dobrą praktyką i oznaką profesjonalizmu jest wskazanie języka strony, co pozwala np. wyszukiwarek na odfiltrowywanie treści pasujących tylko do języka wyszukującego. Atrybut wskazujący na język można dodać do dowolnego znacznika jednak najczęściej umieszcza się go w znaczniku html na początku strony:

```
/>
<html lang="pl">
```

72

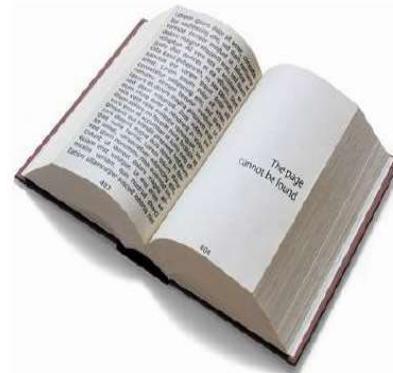
Minimalistyczna wersja szablonu strony w HTML5

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

73

Budowanie struktury strony

Na początek warto wyobrazić sobie stronę internetową jak książkę z rozdziałami, podrozdziałami, pod-pod-rozdziałami, akapitami, listami, ilustracjami



74

Lista znaczników

<!---->	Komentarz
<!DOCTYPE>	Definicja typu dokumentu
<a>	Odkładnik (kotwica)
<blockquote>	Komentarz blokowy
<body>	Ciało dokumentu HTML
 	Przejście do nowej linii
<button>	Przycisk
<caption>	Tytuł tabeli
<center>	Centrowanie tekstu (niezalecany)
<cite>	Cytat
<code>	Fragment kodu programu
<dd>	Opis definicji
	Tekst skasowany
<dfn>	Definiowane słowo lub wyrażenie
<div>	Sekcja w dokumencie
<fieldset>	Grupa pól
<form>	Formularz
<frame>	Definicja zawartości ramki
<frameset>	Grupa ramek na stronie z ramkami
<h1>	Nagłówek poziomu 1
<h6>	Nagłówek poziomu 6
<head>	Definicja informacji o dokumencie
<hr>	Pozioma linia
<iframe>	Ramka漂wająca (iframe)
	Obrazek
<input>	Pole formularza (opis ogólny)
<ins>	Tekst wstawiony
<label>	Etykieta dla kontrolki formularza
<legend>	Tytuł dla grupy pól
	Element listy
<link>	Odkładnik do zasobu
<map>	Mapa odsyłaczy graficznych
<meta>	Informacje meta
<noscript>	Sekcja noscript
<object>	Osadzony obiekt
	Lista numerowana
<option>	Opcja w liście rozwijalnej
<p>	Paragraf
<param>	Parametr dla obiektu
<pre>	Tekst preformatowany
<samp>	Przykład kodu komputerowego
<script>	Skrypt

Lista znaczników

<select>	Lista wybieralna
<small>	Pomniejszony tekst
	Sekcja w dokumencie
	Tekst silnie wyróżniony
<style>	Definicja arkuszy stylów CSS
<sub>	Tekst w dolnym indeksie
<sup>	Tekst w górnym indeksie
<table>	Tabela
<tbody>	Ciało tabeli
<td>	Komórka tabeli
<textarea>	Pole do wprowadzania wielu linii
<tfoot>	Stopka tabeli
<th>	Komórka nagłówka tabeli
<thead>	Nagłówek tabeli
<title>	Tytuł dokumentu HTML
<tr>	Wiersz tabeli
<tt>	Tekst "maszynowy"
	Lista wypunktowana

i inne ...

Więcej -> www.w3schools.com/tags

ZNACZNIKI HIPERTEKSTOWE - uwagi



Zawierają linki do innych dokumentów lub innych części tego samego dokumentu.

```
<h2 id="sekcja2">First topic</h2>
.....
<a href="#sekcja2">Idz do sekcji 2 </a>
```

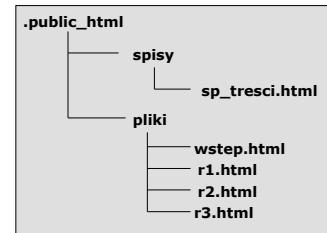
```
<a href="http://szkola.pl" title="Strona szkoły">Szkoła</a>
<a href="index.html#ofirmie" title="Opis firmy">O nas</a>
```

Zamieszanie wokół target="_blank".

```
<a href="http://szkola.pl" title="Strona szkoły" target="_blank"> Szkoła </a>
```

Odsyłacze do plików lokalnych

```
<h2>Spis treści:</h2>
<ol>
<li><a href="../pliki/wstep.html">Wstęp</a>
<li><a href="../pliki/r1.html">Rozdział 1</a>
</ol>
<li><a href="../pliki/r1.html#zal">Założenia wstępne</a>
<li><a href="../pliki/r1.html#teza">Teza pracy</a>
</ol>
<li><a href="../pliki/r2.html">Rozdział 2</a>
<li><a href="../pliki/r3.html">Podsumowanie</a>
</ol>
```



Spis treści:

1. [Wstęp](#)
2. [Rozdział 1](#)
 1. [Założenia wstępne](#)
 2. [Teza pracy](#)
3. [Rozdział 2](#)
4. [Podsumowanie](#)

Element Liniowy, a Element Blokowy



Elementy liniowe zajmą tylko tyle miejsca ile same potrzebują, pozostawiając miejsce na tej samej linii dla kolejnych elementów liniowych.

Natomiast elementy blokowe zawsze wyświetlane są na nowej linii i zajmują jej całą powierzchnię, nakazując kolejnym elementom ukazanie się pod spodem, na kolejnej linii.

Typy elementów HTML cd.

Dowolny element HTML należy do jednej z kategorii:

1. block: duże bloki zawartości

- Zajmują zawsze całą szerokość viewport (przepływ wertykalny - top to bottom)
- Mają wysokość i szerokość height width
- Mogą zawierać inne elementy (block lub inline) jako elementy zagnieżdżone (children)



Przykłady:

<p>, <h1>, , , <table>, <div>

2. inline: małe ilość treści

Przykłady:

<a>, , ,

- Szerokość ustalana według potrzeb (przepływ horyzontalny (from left to right))

- nie ma szerokości ani długości height width

- nie może mieć elementów zagnieżdżonych (children)

- nie może być pozycjonowany (np. przez float lub position)



ELEMENTY BLOKOWE

- Tekst preformatowany `<pre>`
- Oddzielanie elementów za pomocą linii `<hr>`
- Listy uporządkowane i nie ``, ``
- Tabele `<table>`
- Akapity `<p>`
- Nagłówki sekcji `<h1>``<h2>``<h3>``<h4>``<h5>``<h6>`
- Sekcje `<div>`
- Sekcje `<section>`
- Nagłówki `<header>`
- Stopki `<footer>`
- Nawigacje `<nav>`

81

Czym jest formularz

Służy do umieszczania na stronie www elementów interaktywnych:

- Pola tekstowe
- Przyciski
- Listy i listy rozwijalne
- Checkboxy i Radiobuttony
- Okno dialogowe „plik”

82

Typy pól formularza

INPUT

Text – zwykłe pole tekstowe

Hidden – pole ukryte dla użytkownika
(widoczne w kodzie)

Password – pole ukrywające wprowadzane znaki

Checkbox – pole wyboru

Radio – przycisk radiowy (jeden z wielu)

Submit - przycisk wysyłający dane z formularza

Reset – przycisk przywracający wartości formularza

Image – przycisk w formie obrazka
działający jak Submit

Button – dowolny przycisk bez szczególnego znaczenia dla formularza

SELECT – lista rozwijalna

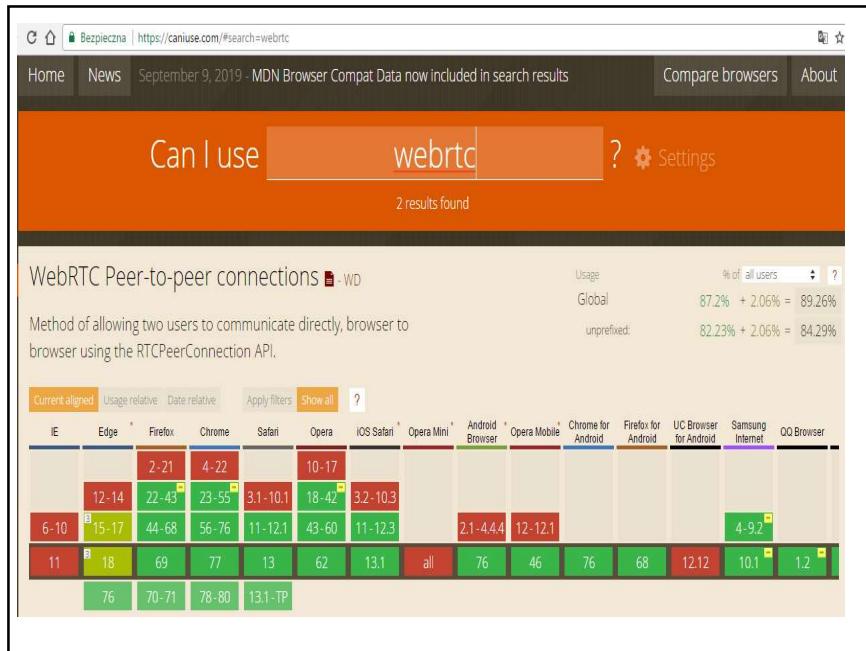
TEXTAREA – pole tekstowe

Dane użytkownika

Imię i Nazwisko	Jan Malinowski
Hasło	*****
Wykształcenie	<input checked="" type="radio"/> wyższe (mgr) <input type="radio"/> wyższe (dr) <input type="radio"/> średnie <input type="radio"/> podstawowe
Miejscowość	Sopot
Opis	To jest miejsce na dłuższy opis...
Języki	<input checked="" type="checkbox"/> Angielski <input checked="" type="checkbox"/> Polski <input type="checkbox"/> Rosyjski <input checked="" type="checkbox"/> Inny

Zapisz Kasuj Lista użytkowników





85

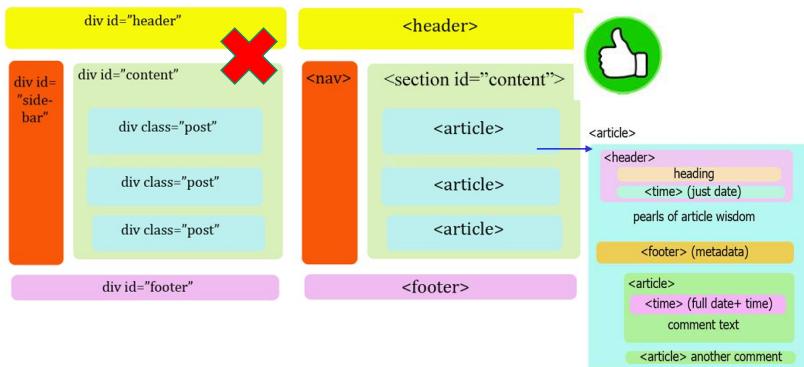
HTML5 a JavaScript

```
function setDuration() {
    var duration = document.getElementById("player").duration;
    document.getElementById("durInfo").innerHTML = "Całość: " +
        + duration.toFixed(2);
    document.getElementById("bar").max = duration;
}
function showInfo() {
    var player = document.getElementById("player");
    var info = document.getElementById("currInfo");
    var bar = document.getElementById("bar");
    info.innerHTML = "Aktualnie: " + player.currentTime.toFixed(2);
    bar.value = player.currentTime;
}
<video id="player" src="wildlife.ogg" controls
    ontimeupdate="showInfo()"
    ondurationchange="setDuration()">
    Ta przeglądarka nie odtwarza wideo.
</video>
```

The screenshot shows a video player interface. The video frame displays a group of horses running on a beach. Below the video is a control bar with a progress bar showing the current time as 00:01 and the total duration as 30.12. There are also buttons for play/pause and volume control. To the left of the video, there is some JavaScript code demonstrating how to interact with the video element to show current and total duration.

86

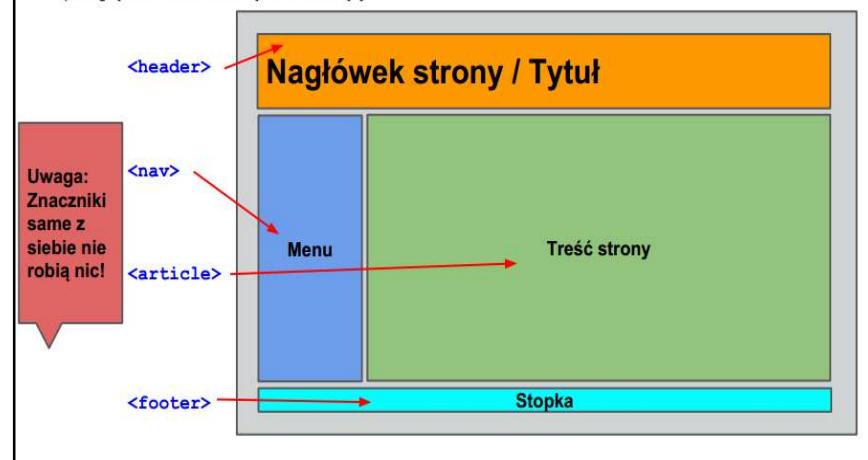
Struktura strony internetowej



87

Budowanie struktury strony

Dzięki HTML5 mamy nowe znaczniki które organizują treści na stronie w sposób logiczny oraz dzięki temu są bardziej zoptymalizowane pod kątem wyszukiwarek. Dodatkowo pozwalają na przyjazną organizację strony i czynią stronę bardziej dostępną np. dla czytników ekranów ułatwiających przeglądanie stron osobą niedowidzącym.



88

PRZYKŁADOWE NOWE ELEMENTY W HTML5

- Header <header></header>
- Footer <footer></footer>
- Nav <nav></nav>
- Main <main></main>
- Section <section></section>
- Article <article></article>
- <audio> i <video> - odtwarzacz muzyki i odtwarzacz filmów
- <canvas> - przestrzeń do rysowania w 2D i 3D



89

Co nam daje semantyka?

łatwiej nam znaleźć odpowiedni blok kodu, niż przeszukiwać ścianę divów i spanów

technologia asystująca lepiej przetworzy strukturę naszej strony (accessibility)

SEO - wyszukiwarki lepiej zindeksują zawartość, wiedząc co jest czym

ZASADA OGÓLNA:

- Znaczniki semantyczne nadają przeznaczenie wybranym treściom na stronie
- Nie należy w nich nadużywać stylów CSS
- Div dalej służą i pozwalają rozmieszczać, układać i stylizować treść witryny

90

Nowe znaczniki semantyczne — mark

```
...  
<section>  
  <article>  
    <header>  
      <h2>Tytuł nagłówka artykułu</h2>  
      <p>Jakaś dodatkowa treść i np. link: <a href="#">więcej...</a></p>  
    </header>  
    <p>Znacznik <mark>article</mark> reprezentuje pewną zamkniętą całość,  
      która może być traktowana niezależnie od całej strony, np. wpis  
      blogowy, artykuł, komentarz.</p>  
  </article>  
</section>  
...
```

Przy okazji — znacznik `<mark>` wyróżnienia fragmentu tekstu.

Tytuł nagłówka artykułu

Jakaś dodatkowa treść i np. link: [więcej...](#)

Znacznik `article` reprezentuje pewną zamkniętą całość, która może być traktowana
niezależnie od całej strony, np. wpis blogowy, artykuł, komentarz.



91

Nowe znaczniki semantyczne — details i summary

```
...  
<section>  
  <details>  
    <summary>Czytaj więcej...</summary>  
    <p>Znaczniki <em>details</em> i <em>summary</em>. </p>  
    <p>Aktualnie zaimplementowane jedynie w Chrome.</p>  
  </details>  
</section>  
...
```

▶ Czytaj więcej...

▼ Czytaj więcej...

↔ Znaczniki `details` i `summary`.

Aktualnie zaimplementowane jedynie w Chrome.

Na razie tylko Chrome

Znacznik `<details>` określa dodatkową informację, którą użytkownik może zobaczyć lub ukryć na życzenie.

Sterowanie pokazywaniem/ukrywaniem zawartości `<details>` steruje atrybut boolowski `open`.

Znacznik `<summary>` określa widoczny nagłówek sekcji `<details>`.

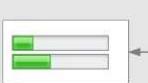
92

Nowe znaczniki semantyczne — meter i progress

```
<meter value="3" min="1" max="10">3 w skali od 1 do 10</meter><br />
<meter value="0.4">40%</meter>
```

Gdy brak implementacji

3 w skali od 1 do 10
40%



Opera, Chrom

Znacznik `<meter>` definiuje wskaźnik, np. zużycie przestrzeni dyskowej, page rank, wynikanki. Uwaga — nie używa się jako paska postępu, do tego jest `<progress>`.

```
<progress value="11" max="100">Ładowanie: 11%</progress>
```

Ładowanie: 11%



Gdy brak implementacji

Opera, Chrom, Firefox

Sterowanie elementem progress — prosty przykład

```
var timeoutObject;
var sek = 0;
function display()
{
    document.getElementById( "pb" ).value = sek++;
    if( sek > 100 ) zeruj();
}
function start()
{
    display();
    window.timeoutObject = setInterval( "display()", 100 )
}
function stop()
{
    window.clearInterval( timeoutObject );
}
function zeruj()
{
    document.getElementById( "pb" ).value = sek = 0;
}

<body>
<h1>Progress bar</h1>
<progress id="pb" value="0" max="100"></progress>
<input type="button" value="Start" onclick="start()" />
<input type="button" value="Stop" onclick="stop()" />
<input type="button" value="Zeruj" onclick="zeruj()" />
</body>
</html>
```

Progress bar



Nowe znaczniki semantyczne: figure i figurecaption

HTML5 wprowadza również pewną nowość ułatwiającą osadzanie ilustracji. Wcześniej aby osadzić na stronie obraz wraz z podpisem musiliśmy:

```
<div>
  
  <p>To jest podpis pod naszym obrazem</p>
</div>
```

Standard HTML oferuje nam rozwiązanie semantyczne `<figure>` dzięki któremu możemy w sposób zorganizowany w ramach jednego bloku osadzać obrazy:

```
<figure>
  
  <figcaption>To jest podpis pod naszym obrazem</figcaption>
</figure>
```



W obu przypadkach efekt jest ten sam.

95

<CANVAS>

```
<canvas id="myCanvas">
  Your browser does not support the HTML5 canvas tag.
</canvas>
<script>
  var c = document.getElementById("myCanvas");
  var ctx = c.getContext("2d");
  ctx.fillStyle = "#FF0000";
  ctx.fillRect(0, 0, 80, 100);
</script>
```



96

Nowa semantyka w Web page - podsumowanie



97

HTML5 Forms

Nowe input types (nowych pola formularza) w HTML5

„Stare” pola formularza w HTML

■ button ■ checkbox ■ file ■ hidden ■ image ■ password ■ radio ■ reset ■ submit ■ text

HTML5 dodaje 13 nowych typów pól:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Nowe atrybuty:

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

98

Forms

<input type> email

```
<input id="email" type="email"/>
```

<input type> url

```
<input id="website" type="url"/>
```

<input type> tel

```
<input id="phone" type="tel"/>
```

99

Forms

<input type> search

Pole na wpisanie frazy do wyszukiwarki. Search Google: `<input type="search" name="googlesearch" />`

Search type -> wsparcie w Chrome, Opera | Safari.

<input type> color

Okreś kolor tła.

ru. Po kliknięciu

Choose a color for background

Color

Basic colors:

Custom colors:

Hue: 160 Sat: 240 Luma: 180 ColorSrgb

OK Cancel Add to Custom Colors

100

Nowe znaczniki semantyczne — audio

```
<audio src="traktor.mp3" controls autoplay loop preload="none">  
Ta przeglądarka nie odtwarza dźwięku.  
</audio>
```

- ▶ **controls** – wyświetlanie panelu sterowania odtwarzaniem, → MSIE [MSIE] [Opera]
- ▶ **autoplay** – automatyczne odtwarzanie od razu, gdy to możliwe, → Chrome [Chrome]
- ▶ **loop** – powtarzanie odtwarzania.
- ▶ **preload** – wskazówka dla przeglądarki: jak postępować z dźwiękiem w trakcie ładowania strony: *auto* – ładowanie pliku w trakcie ładowania strony; *metadata* – ładowanie metadanych w trakcie ładowania strony; *none* – plik nie powinien być ładowany w trakcie ładowania strony. Dla *autoplay = true* atrybut *preload* jest ignorowany.

101

<VIDEO>

```
<video width="200" height="150" autoplay controls poster="vid_poster.jpg">  
  <source src="wildlife.mp4" type="video/mp4" />  
  <source src="wildlife.ogg" type="video/ogg" />  
  Ta przeglądarka nie odtwarza wideo.  
</video>
```

```
<video width="320" height="240" autoplay controls poster="vid_poster.jpg">  
  <source src="wildlife.mp4" type="video/mp4" />  
  <source src="wildlife.ogg" type="video/ogg" />  
  Ta przeglądarka nie odtwarza wideo.  
</video>
```



102