

RWD, SEO i inne skrótowce

Projektowanie i programowanie systemów internetowych I

mgr inż. Krzysztof Rewak

28 maja 2018

Wydział Nauk Technicznych i Ekonomicznych

Państwowa Wyższa Szkoła Zawodowa im. Witelona w Legnicy

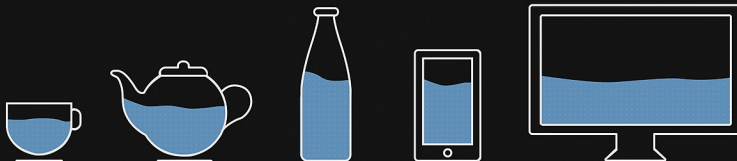
Plan prezentacji

1. RWD
2. SPA
3. SEO
4. Podsumowanie

RWD

RWD (od ang. *responsive web design*) to podejście w projektowaniu aplikacji internetowych polegające na tworzeniu frontendu, który dopasowuje się do urządzenia na którym jest wyświetlany.

CONTENT IS LIKE WATER



“ You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot. ”

Josh Clark (originally Bruce Lee) - Seven deadly mobile myths

Illustration by Stéphanie Walter

Jak nie robić?

Zanim RWD zagościło na dobre w internecie, jednym ze sposobów na radzenie sobie z urządzeniami o mniejszej rozdzielczości było tworzenie specjalnych stron w "wersji mobilnej".

Przykładem może być archaiczny `mobile.facebook.com`.

Jak robić?

Obecny trend to tworzenie jednej aplikacji frontendowej, która dynamicznie dopasowuje się do rozdzielczości urządzenia.

Przykład: 1920px na szerokości

Krzysztof REWAK

Aktualności

Kursy

Oceny

FAQ

Kontakt

mgr inż. Krzysztof Rewak

Wydział Nauk Technicznych i Ekonomicznych

Państwowa Wyższa Szkoła Zawodowa im. Witłona w Legnicy

Kilka słów o mnie

Naukowo

Ukończyłem studia inżynierskie i magisterskie na kierunku Automatyka i Robotyka na Wydziale Elektroniki Politechniki Wrocławskiej. Specjalizowałem się, kolejno na studiach pierwszego i drugiego stopnia, w systemach informatycznych w automatyce oraz technologiach informacyjnych w systemach automatyki.

Od pierwszego semestru 2016 roku prowadzę zawodowe praktyki dla studentów i uczę praktycznie programowania w ramach projektu Legnica.IT. W 2017 miałem być uszytych mój wykład w ramach Chwasty Seminarium Naukowych. Od 2017 pracuję jako nauczyciel akademicki na Wydziale Nauk Technicznych i Ekonomicznych Państwowej Wyższej Szkoły Zawodowej im. Witłona w Legnicy.

Zawodowo

Od 2014 roku pracuję jako programista. Zaczynałem jako młodszy programista PHP w małym wrocławskim software house, gdzie pod baczym okiem specjalistów nauczyłem się realnego i praktycznego podejścia do zawodu. Od 2015 roku pracuję w grupie IT-Motion - najpierw jako programista Pythona, następnie jako lider zespołu programistycznego.

Zawsze gdy mogę wybrać, programuję w PHP - najchętniej w najnowszej wersji) oraz przede wszystkim w Laraweli i Phalconie. Mam zawodowe doświadczenie w pracy z popularnymi frameworkami JavaScripta, Pythonem oraz C#.

Prywatnie

Urodziłem się i mieszkam od urodzenia w Legnicy (choć mieskam w tym pięcioletnią przerwę na mieszkanie i studiowanie we Wrocławiu). W wolnym czasie lubię siedzieć nad książkami wszelkiego gatunku, śledzić przeróżne serie oraz jeździć po legnickich i okolicznych drogach rowerowymi.

Ponadto jestem plwowarem domowym, więc warzę w domu własnego piwa. Jestem zapalonym fanem Gwiazdnych wojen, a także - a jakżeby inaczej! - programuję najbardziejże rzeczy w najbardziejzych technologiach w najbliższym celu.

Informacje

Krzysztof Rewak © 2018

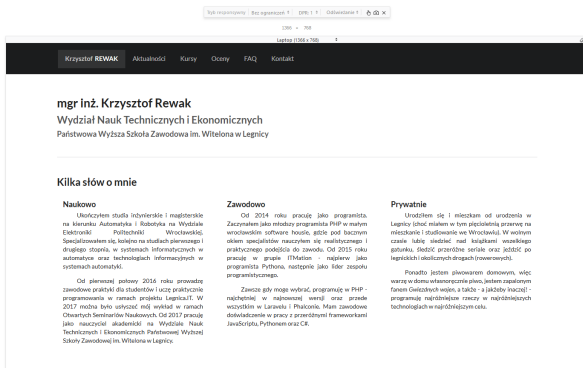
Linki na zewnątrz

[Facebook](#) [LinkedIn](#) [rewak.pl](#)

Technologie

Phalcon, Vue.js, Semantic UI

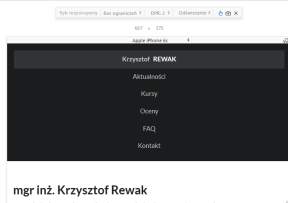
Przykład: 1366px na szerokości



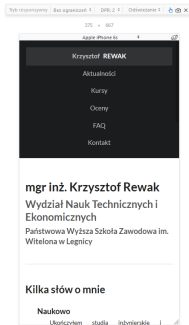
Przykład: 1280px na szerokości



Przykład: 667px na szerokości



Przykład: 375px na szerokości



Od strony kodu

```
<div class="row">  
  <div class="column">Content A</div>  
  <div class="column">Content B</div>  
  <div class="column">Content C</div>  
</div>
```

Jak robić?

Jak domyślnie będzie wyglądał przykład z poprzedniego slajdu?

Będą to ułożone, jeden pod drugim, kontenery szerokości i wysokości dopasowanej do wstawionego w nich tekstu.

Od strony kodu

Co się stanie po dodaniu poniższych reguł CSS?

```
.row {  
  
    width: 100%;  
  
    .column {  
        width: 33%;  
        background: Red;  
        display: inline-block;  
    }  
  
}
```

Od strony kodu

Nic się nie stanie po dodaniu powyższych reguł. Należy je wpierw przetworzyć z Sass na CSS:

```
.row {  
    width: 100%;  
}  
  
.row .column {  
    width: 33%;  
    background: Red;  
    display: inline-block;  
}
```


Co dalej?

Wówczas każdy z kontenerów z tekstem zajmie $1/3$ szerokości okna.
Utworzy to system kolumn, w którym można wygodnie prezentować dane.
Czasami jednak może źle się to wyświetlić na węższych ekranach.

Opcja #1: stworzenie dwóch arkuszy dla różnych rozdzielczości ekranu:

```
<link
  rel="stylesheet"
  media="screen and (min-width: 900px)"
  href="widescreen.css">
```

```
<link
  rel="stylesheet"
  media="screen and (max-width: 600px)"
  href="smallscreen.css">
```

Media queries

Opcja #2: wstawianie różnych reguł wewnątrz arkusza:

```
.row {  
  
    width: 100%;  
  
    .column {  
        width: 33%;  
        background: Red;  
        display: inline-block;  
    }  
  
    @media screen and (max-width: 768px) {  
        .column {  
            width: 100%;  
            display: block;  
        }  
    }  
}
```

Jak punkty brzegowe przyjąć przy tworzeniu responsywnego frontendu?
Bootstrap wykorzystuje sensowny podział *breakpointów*:

- do 576px dla urządzeń bardzo małych (np. telefonów w pozycji pionowej)
- do 768px dla urządzeń małych (np. telefonów w pozycji poziomej)
- do 992px dla urządzeń średnich (np. tabletów)
- do 1200px dla urządzeń dużych (głównie laptopów)
- powyżej 1200px dla urządzeń bardzo dużych

Jakich rozdzielczości się używa?

Rozdzielczość ekranu	Pozyskiwanie			Zachowanie			Konwersje		
	Użytkownicy	Nowi użytkownicy	Sejse	Współczynnik odrzuceń	Strony / sesja	Śr. czas trwania sesji	Współczynnik konwersji celu	Realizacja celu	Wartość celu
	776 % całkow.: 100,00% (776)	778 % całkow.: 100,00% (778)	3 462 % całkow.: 100,00% (3 462)	18,26% Śr. dla widoku danych: 18,26% (0,00%)	4,81 Śr. dla widoku danych: 4,81 (0,00%)	00:03:58 Śr. dla widoku danych: 00:03:58 (0,00%)	0,00% Śr. dla widoku danych: 0,00% (0,00%)	0 % całkow.: 0,00% (0)	0,00 USD % całkow.: 0,00% (0,00 USD)
1. 1920x1080	375 (46,36%)	370 (47,56%)	1 686 (40,70%)	12,69%	6,02	00:05:05	0,00%	0 (0,00%)	0,00 USD (0,00%)
2. 1536x864	112 (13,84%)	110 (14,14%)	235 (6,79%)	11,91%	4,58	00:03:49	0,00%	0 (0,00%)	0,00 USD (0,00%)
3. 360x640	112 (13,84%)	112 (14,40%)	669 (19,32%)	34,23%	2,81	00:02:30	0,00%	0 (0,00%)	0,00 USD (0,00%)
4. 1366x768	87 (10,75%)	86 (11,09%)	344 (9,94%)	12,21%	3,36	00:02:42	0,00%	0 (0,00%)	0,00 USD (0,00%)
5. 1440x900	18 (2,22%)	15 (1,93%)	106 (3,06%)	17,92%	3,51	00:03:07	0,00%	0 (0,00%)	0,00 USD (0,00%)
6. 2048x1152	11 (1,36%)	8 (1,03%)	49 (1,42%)	24,49%	15,51	00:07:28	0,00%	0 (0,00%)	0,00 USD (0,00%)
7. 1280x1024	10 (1,24%)	9 (1,16%)	36 (1,04%)	2,78%	4,36	00:01:46	0,00%	0 (0,00%)	0,00 USD (0,00%)
8. 1680x1050	10 (1,24%)	7 (0,90%)	47 (1,36%)	14,89%	3,32	00:05:56	0,00%	0 (0,00%)	0,00 USD (0,00%)
9. 1600x900	9 (1,11%)	9 (1,16%)	34 (0,98%)	23,53%	3,68	00:01:47	0,00%	0 (0,00%)	0,00 USD (0,00%)
10. 1280x800	6 (0,74%)	4 (0,51%)	87 (2,51%)	22,99%	3,72	00:01:58	0,00%	0 (0,00%)	0,00 USD (0,00%)
11. 1024x768	5 (0,62%)	4 (0,51%)	9 (0,26%)	66,67%	1,33	00:00:01	0,00%	0 (0,00%)	0,00 USD (0,00%)
12. 400x712	5 (0,62%)	5 (0,64%)	19 (0,55%)	21,05%	2,74	00:01:53	0,00%	0 (0,00%)	0,00 USD (0,00%)
13. 1360x768	4 (0,49%)	2 (0,26%)	7 (0,20%)	14,29%	3,29	00:03:11	0,00%	0 (0,00%)	0,00 USD (0,00%)
14. 1152x864	3 (0,37%)	1 (0,13%)	5 (0,14%)	40,00%	3,80	00:01:02	0,00%	0 (0,00%)	0,00 USD (0,00%)
15. 1280x720	3 (0,37%)	2 (0,26%)	6 (0,17%)	83,33%	1,50	00:00:57	0,00%	0 (0,00%)	0,00 USD (0,00%)
16. 1920x1200	3 (0,37%)	3 (0,39%)	3 (0,09%)	0,00%	4,67	00:10:53	0,00%	0 (0,00%)	0,00 USD (0,00%)
17. 2560x1440	3 (0,37%)	3 (0,39%)	3 (0,09%)	33,33%	8,33	00:00:25	0,00%	0 (0,00%)	0,00 USD (0,00%)
18. 640x360	3 (0,37%)	1 (0,13%)	3 (0,09%)	33,33%	1,67	00:00:46	0,00%	0 (0,00%)	0,00 USD (0,00%)
19. 2048x1153	2 (0,25%)	2 (0,26%)	4 (0,12%)	75,00%	1,50	00:00:12	0,00%	0 (0,00%)	0,00 USD (0,00%)
20. 480x800	2 (0,25%)	2 (0,26%)	2 (0,06%)	50,00%	1,50	00:00:03	0,00%	0 (0,00%)	0,00 USD (0,00%)

Bootstrap to obecnie standard

Poniższy kod utworzy wiersz z trzema kolumnami równej szerokości, ale na ekranach o rozdzielczości mniejszej niż 576px kolumny ustawią się jedna nad drugą.

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

Do czego można wykorzystać RWD?

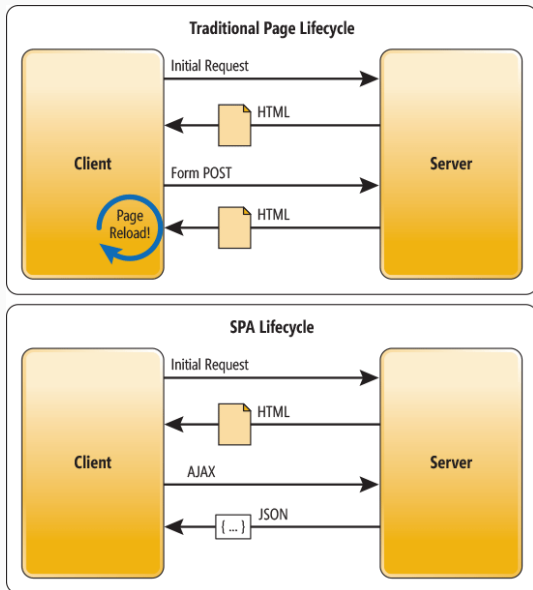
- ukrycie paska nawigacyjnego pod tzw. *hamburger menu*
- przemieszczenie bocznych pasków (najczęściej lewy nad zawartość, prawy pod)
- zmiana umieszczenia przycisków/tagów (normalnie jeden obok drugiego, na małych ekranach - pod sobą, ale wypełniające całą szerokość ekranu)
- marginesy wewnętrzne i zewnętrzne różnie wyglądają na różnej wielkości ekranach
- itd. itp.

SPA

SPA (od ang. *single-page application*) to podejście w projektowaniu aplikacji internetowych polegające na tworzeniu frontendu, który nie wymaga przeładowania strony.

Wykorzystywane są w tym celu asynchroniczne odpytywania serwera, czyli AJAX.

SPA

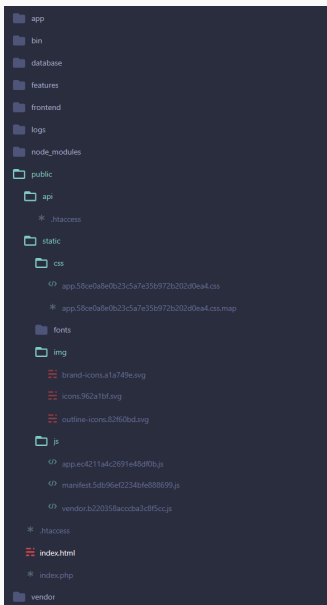


Jak wygląda taka aplikacja?

Dobrze skonstruowana SPA powinna składać się z jednego pliku `index.html` oraz plików pomocniczych (przede wszystkim JavaScript, CSS, grafiki, fonty).

Pliki pomocnicze można umieścić na CDN-ie.

Jak wygląda taka aplikacja?



Jak wygląda taka aplikacja?

Wszelkie *dynamiczne* informacje powinny zostać pobrane z API wystawionego na backendzie. Backend ten może funkcjonować zarówno na tej samej instancji serwera co frontend, jak i na całkiem innej maszynie z innym adresem.

Domyślnie SPA mogą mieć problem z przekierowaniem użytkownika na wybraną stronę. Rozwiązaniem może być wykorzystanie routera będącego częścią współczesnych frameworków frontendowych.

W kodzie JavaScriptowym umieszcza się listę ścieżek i akcji jakie mają zostać zrealizowane po wejściu na ścieżkę. Odpowiednie reguły w ustawieniach serwera HTTP przekierują cały ruch na plik `index.html` z załączonym routerem, który zajmie się wywołaniem odpowiedniej akcji.

Router

```
27 Vue.use(Router)
28
29 let routes = [
30   { path: "/", meta: { section: "home" }, name: "home", component: Home },
31
32   { path: "/aktualnosci", meta: { section: "news" }, name: "news", component: NewsReel },
33   { path: "/aktualnosci/:id", meta: { section: "news" }, name: "news.entry", component: NewsEntry },
34
35   { path: "/kursy", meta: { section: "courses" }, name: "courses", component: CoursesTable },
36   { path: "/kursy/:id", meta: { section: "courses" }, name: "course.page", component: CoursePage },
37
38   { path: "/oceny", meta: { section: "grades" }, name: "grades", component: GradesForm },
39
40   { path: "/faq", meta: { section: "faq" }, name: "faq", component: FAQ },
41   { path: "/plan-zajec", meta: { section: "schedule" }, name: "schedule", component: Schedule },
42
43   { path: "/kontakt", meta: { section: "contact" }, name: "contact", component: ContactPage },
44   { path: "/kontakt/formularz", meta: { section: "contact" }, name: "contact.form", component: ContactPage },
45   { path: "/kontakt/konsultacje", meta: { section: "contact" }, name: "contact.consultations", component: ContactPage },
46
47   { path: "/logowanie", meta: { section: "dashboard", requiresGuest: true }, name: "login", component: LoginForm },
48   { path: "/zarzadzaj", meta: { section: "dashboard", requiresAuth: true }, name: "dashboard", component: Dashboard },
49   { path: "/zarzadzaj/konto", meta: { section: "dashboard", requiresAuth: true }, name: "dashboard.account", component: AccountPage },
50   { path: "/zarzadzaj/oceny", meta: { section: "dashboard", requiresAuth: true }, name: "dashboard.grades", component: ManagementGradesForm },
51   { path: "/zarzadzaj/logi", meta: { section: "dashboard", requiresAuth: true }, name: "dashboard.logger", component: Logger },
52
53   { path: "/brak-dostepu", meta: { section: "error" }, name: "not-allowed", component: NotAllowed },
54   { path: "**", meta: { section: "error" }, name: "not-found", component: NotFound },
55 ]
56
57 const router = new Router({
58   mode: "history",
59   routes: routes
60 })
```

SPA mają różnorodne zastosowania:

- małe witryny firm,
- panele administracyjne systemów zarządzania treścią,
- panele administracyjne systemów typu CRM lub ERP,
- portale społecznościowe...

Czyli praktycznie wszystko, co można obejrzeć w internecie.

Plusy i minusy

Plusy:

- wygodne i *efektywne* dla użytkownika,
- pełne odseparowanie back- i frontendu,
- zmniejszenie czasu wczytywania strony.

Plusy i minusy

Minusy:

- trzeba poznać nową (i często niestabilną) technologię,
- trzeba konfigurować drugie (frontendowe) środowisko,
- nie wszystkie przeglądarki będą działały tak samo.

SEO

SEO (od ang. *search engine optimization*) to podejście w projektowaniu aplikacji internetowych polegające na zwiększaniu przewidywanej pozycji witryny w wynikach wyszukiwarek internetowych.

Polska często spotykana nazwa to **pozycjonowanie**.

Jakie są narzędzia SEO?

Trudno odpowiedzieć w prosty i jednoznaczny sposób. Rynek wyszukiwarek zmienia się dynamicznie i ciężko sprecyzować sensowny sposób na SEO.

Podstawowym narzędziem wstępnej optymalizacji pod wyszukiwarki jest uzupełnienie tagów <meta> w nagłówku strony internetowej.

<meta>

```
< > router.js x index.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="author" content="Krzysztof Rewak">
7     <meta name="description" content="Strona uczelniana Krzysztofa Rewaka; dostęp do aktualności, prowadzonych kursów, ocen oraz kontaktu.">
8     <meta name="keywords" content="Państwowa Wyższa Szkoła Zawodowa im. Witelona w Legnicy, PWSZ, Legnica, Krzysztof Rewak, Informatyka, programowanie, PHP, C++">
9     <meta name="viewport" content="width=device-width, initial-scale=1">
10    <title>Krzysztof Rewak :: Wydział Nauk Technicznych i Ekonomicznych :: Państwowa Wyższa Szkoła Zawodowa im. Witelona w Legnicy</title>
11  </head>
12
13  <body>
14    <div id="app"></div>
15  </body>
16 </html>
17
```

Stety-niestety nadużycia tejże formy spowodowały, że boty Google i innych wyszukiwarek skanują obecnie całe strony w poszukiwaniu *organicznych* słów kluczowych.

Jak nie ogarniać SEO?

Jak popsuć swój wynik w Google?

- zamieszczać nieprawdziwe dane, słowa kluczowe,
- zamieszczać zbyt wiele słów kluczowych,
- ukrywać elementy przed użytkownikiem,
- być linkowanym ze stron ze słabym wskaźnikiem SEO,
- budować zły kod.

Jak ogarnąć SEO?

Jak poprawić swój wynik w Google?

- umieścić mapę strony;
- sensownie używać HTML;
- oczyścić znaczniki HTML;
- uzupełnić tagi HTML;
- *pretty URLs* i sensowne zawartości linków;
- stworzyć responsywny layout,
- i wiele, wiele innych.

PageSpeed Insights

http://pwsz.rewak.pl/

ANALIZUJ



Na komórki



Na komputery

Page Speed

Unavailable

Optimalizacja

Good

83 / 100

System PSI obecnie pokazuje raport jednostronnicowy. Raport na temat użytkowania Chrome [zawiera za mało danych o rzeczywistej szybkości](#) tej strony. Pomimo tego system PSI przeanalizował ją, by znaleźć możliwości optymalizacji, dzięki którym jej szybkość można poprawić. Zapoznaj się z poniższymi zaleceniami. [Więcej informacji](#)

Statystyki strony

Według oszacowania PSI ta strona wymaga 7 przebiegów w obie strony blokujących renderowanie i około 17 zasobów (0,4 MB), by została załadowana. Strona, która ma wartość mediany, wymaga 4 przebiegów w obie strony blokujących renderowanie i około 89 zasobów (1,3 MB), by została załadowana. Strony wymagające mniejszej liczby bajtów i przebiegów w obie strony są szybsze.

Wskazówki optymalizacyjne

Wyeliminuj blokujący renderowanie kod JavaScript i CSS z części strony widocznej na ekranie

» [Pokaż, jak to naprawić](#)

Wykorzystaj pamięć podręczną przeglądarki

» [Pokaż, jak to naprawić](#)

Nadaj priorytet widocznej treści

» [Pokaż, jak to naprawić](#)



Wykorzystaj pamięć podręczną przeglądarki

► [Pokaż, jak to naprawić](#)

Nadaj priorytet widocznej treści

► [Pokaż, jak to naprawić](#)

Już zastosowane rozwiązania optymalizujące

▼ [Ukryj szczegóły](#)

Skróć czas odpowiedzi serwera

Serwer odpowiedział szybko. Dowiedz się więcej o [optymalizacji czasu odpowiedzi serwera](#).

Unikaj przekierowań stron docelowych

Strona nie zawiera przekierowań. Dowiedz się więcej o [unikaniu przekierowań strony docelowej](#).

Włącz kompresję

Kompresja jest stosowana. Dowiedz się więcej o [stosowaniu kompresji](#).

Zmniejsz CSS

Kod CSS jest zmniejszony. Dowiedz się więcej o [zmniejszaniu kodu CSS](#).

Zmniejsz HTML

Kod HTML jest zmniejszony. Dowiedz się więcej o [zmniejszaniu kodu HTML](#).

Zmniejsz JavaScript

Zawartość pliku JavaScript jest zmniejszona. Dowiedz się więcej o [zmniejszaniu pliku JavaScript](#).

Zoptymalizuj obrazy

Grafiki są zoptymalizowane. Dowiedz się więcej o [optymalizacji grafik](#).

Dane dotyczące prędkości są aktualizowane codziennie, a wyniki optymalizacji są buforowane przez 30 sekund. Aby zobaczyć zaktualizowaną analizę optymalizacji, zaczekaj 30 sekund, zanim uruchomisz ponownie test.



Wydajność sieci

Więcej informacji o [narzędziach Google do testowania](#)

Prześlij opinię

Masz pytanie lub komentarz na temat narzędzia PageSpeed

O narzędziu PageSpeed Insights

PageSpeed Insights analizuje zawartość strony internetowej, a

Podsumowanie

Bibliografia i ciekawe źródła



https:

`//www.w3schools.com/cssref/css3_pr_mediaquery.asp`



`https://getbootstrap.com/docs/4.1/layout/grid/`



https:

`//msdn.microsoft.com/en-us/magazine/dn463786.aspx`



https:

`//developers.google.com/speed/pagespeed/insights/`

Pytania?

Kod prezentacji dostępny jest w repozytorium git pod adresem
<https://bitbucket.org/krewak/pwsz-ppsi>



Wszystkie informacje dot. kursu dostępne są pod adresem
<http://pwsz.rewak.pl/kursy/4>

