

## Koncepcja RWD

RWD (ang. Responsive Web Design — projektowanie elastycznych stron internetowych)

<https://alistapart.com/article/responsive-web-design/>

Elastyczna strona internetowa zawiera dwa ważne składniki. Pierwszy z nich to metaznacznik

### Viewport

[https://www.w3schools.com/css/css\\_rwd\\_viewport.asp](https://www.w3schools.com/css/css_rwd_viewport.asp)

Metaznacznik viewport z ustawieniem parametru initial-scale na 1 spowoduje, że strona internetowa przy pierwszym otwarciu będzie miała szerokość równą stu procentom szerokości ekranu urządzenia.

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

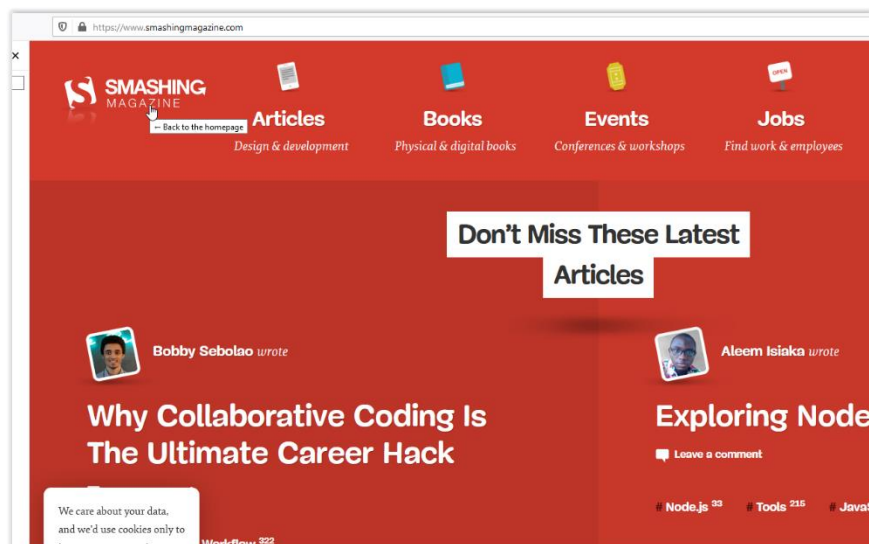
### zapytania medialne CSS3

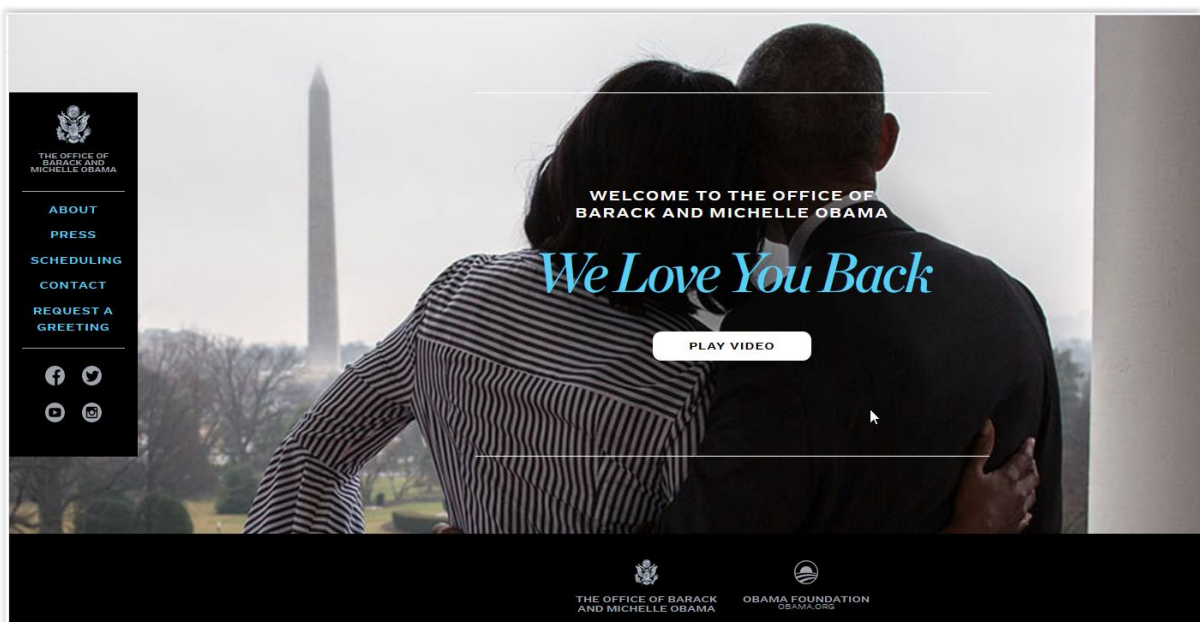
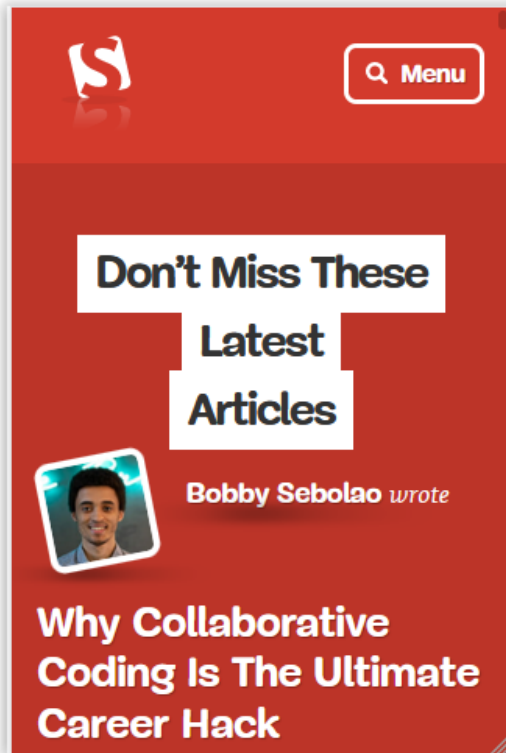
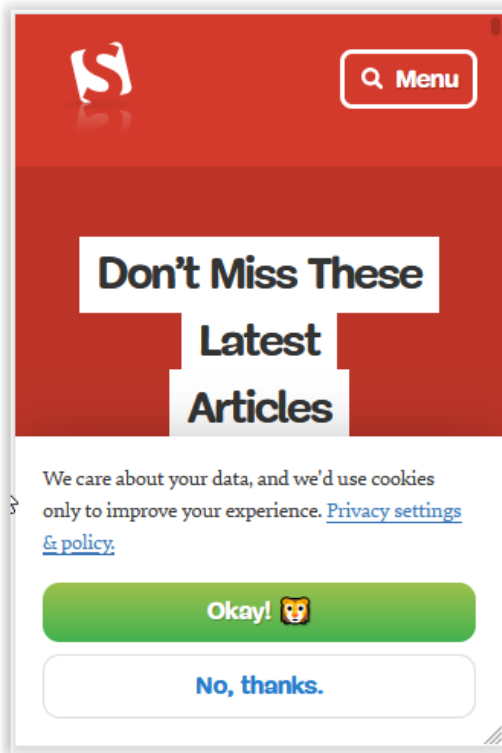
umożliwiają definiowanie arkuszy stylów przeznaczonych dla ekranów o różnych szerokościach.

Na przykład poniższa reguła CSS ukrywa obrazy, jeśli ekran ma szerokość w przedziale od 321 do 480 pikseli:

```
<style>
@media screen and (min-width: 321px) and (max-width: 480px) {
  img { display: none; }
}
</style>
```

Praktyczne przykłady RWD:





Narzędzia dla programistów — The Office of Barack and Michelle Obama — <https://barackobama.com/>

Inspektor   Konsola   Debugger   Sieć   Edytor stylów   Wydajność   Pamięć   Dane   Dostępność

Szukaj w kodzie HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body class="page-landing">
    <div style="display: none;">
    <a class="sr-only sr-only-focusable" href="#content">Skip to main content</a>
    <div class="page-wrapper">
    <div id="video" class="hidden video-modal js-video-modal" data-video="https://www.youtube.com/embed/uwRmBjFxttc?autoplay=1&modestbranding=1&rel=0&showinfo=0">
    <script async="" src="//www.google-analytics.com/analytics.js">
    <script>
    <script src="/js/main.min.js">
  </body>
</html>
```

Filtruj style

```
element {
}
.page-landing {
  background-image: none;
}
body {
  background: #282626;
  background-attachment: scroll;
  background-image: none;
  background-attachment: fixed;
  font: 15px/1.3 Ringside Regular A,Ringside Regular B,sans-serif;
}
body, html {
  height: 100%;
  margin: 0;
  overflow-x: hidden;
  padding: 0;
}
body {
  margin: 0;
}
*, ::after, ::before {
  box-sizing: inherit;
}
Odziedziczony po html
html {
  font-family: sans-serif;
  -webkit-text-size-adjust: 100%;
```

Układ   Wyliczone   Lokalne zmiany   Cz

Flexbox

Wybierz kontener Flex lub element, aby kontynuować.

Siatka

CSS Grid nie jest używany na tej stronie

Model pudełkowy

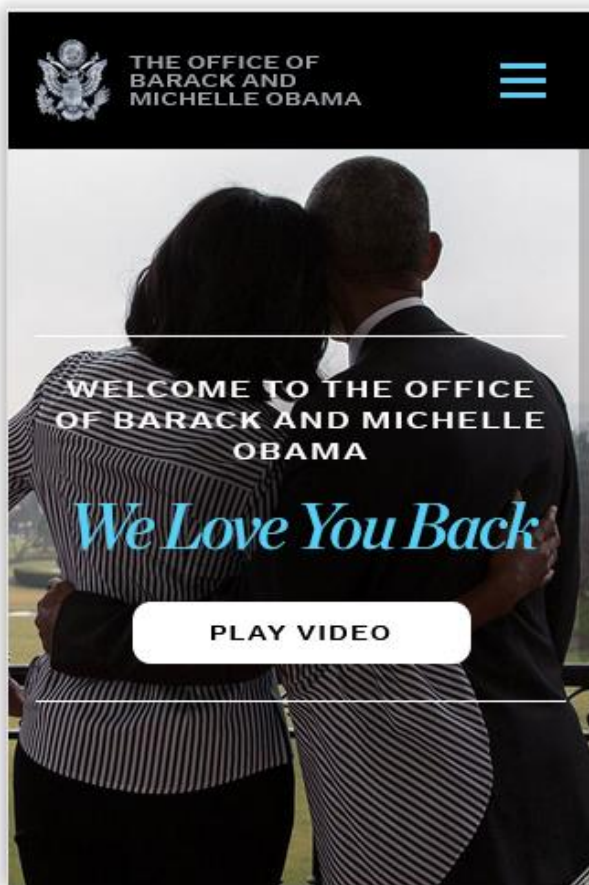
margin: 0  
border: 0  
padding: 0  
1704x966

1704x966   static

Własności modelu pudełkowego

box-sizing	border-box
display	block
float	none
line-height	19.5px
position	static
z-index	auto

html > body.page-landing



## Ograniczenia technik elastycznych.

- skalowalności obrazów

obrazy dostosowuje się do szerokości ekranu poprzez zastosowanie deklaracji CSS **max-width: 100%** albo w ogóle się je ukrywa przy użyciu deklaracji **display: none**, gdy są niepotrzebne. Jednakże w takim przypadku, niezależnie od wielkości urządzenia z którego korzystamy obraz zostaje pobrany i przeskalowany do wielkości urządzenia.

Istnieje inna możliwość:






## HTML <picture> Tag

[https://www.w3schools.com/tags/tag\\_picture.asp](https://www.w3schools.com/tags/tag_picture.asp)

```
<h1>The picture element</h1>

<picture>
  <source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
  <source media="(min-width: 465px)" srcset="img_white_flower.jpg">
  
</picture>
```

Obraz o nazwie img\_pink\_flowers.jpg o wyższej rozdzielczości i zapewne większej szerokości, zostanie wysłany do przeglądarki tylko wtedy, gdy rozmiar jej okna będzie nie mniejszy niż 650 pikseli. Natomiast obraz img\_white\_flower.jpg, o mniejszej rozdzielczości i szerokości, a także o mniejszym rozmiarze, zostanie wysłany do przeglądarki o szerokości nie mniejszej niż 465 pikseli. Na końcu listy znajduje się dodatkowy element <img>, który dostarcza obraz do przeglądarek nieobsługujących elementu <picture>.

Browser Support					
The numbers in the table specify the first browser version that fully supports the element.					
Element					
<picture>	38.0	13.0	38.0	9.1	25.0

## Systemy szkieletowe i inne systemy ułatwiające nam pracę

- system Bootstrap dostarcza style siatki, podstawowe reguły stylistyczne i style interfejsu użytkownika. Skeleton jest jednym z tych systemów, jest on dostępny jako motyw do Wordpressa <http://themes.simplethemes.com/skeleton/>

# skeleton

A Responsive WordPress Theme

Home

Tutorials

Sample Blog

WooCommerce

## Skeleton is a Responsive, Mobile-friendly Theme for WordPress



Skeleton is a simple, responsive ([video](#)) theme from [Simple Themes](#). If you like this theme, we encourage you to try some of our other [WordPress themes](#).

Today, people visit your website from a variety of devices ranging from desktop computers, tablets, and smart phones. The average user spends only a few seconds before making a decision to stay on your site or not. Skeleton is a WordPress theme aimed at helping you build simple, uncluttered, useable, and mobile-friendly WordPress sites.

Simple

Lightweight

Mobile

The tabs are clean and simple unordered-list markup and basic CSS..

# Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

Get started

Download

Currently v4.4.1



# Foundation

The most advanced responsive front-end framework in the world.

Download Foundation 6

★ 21.4k GitHub stars

🐦 @ZURBfoundation

### Wady i zalety systemów szkieletowych:

Może zaczniemy od zalet:

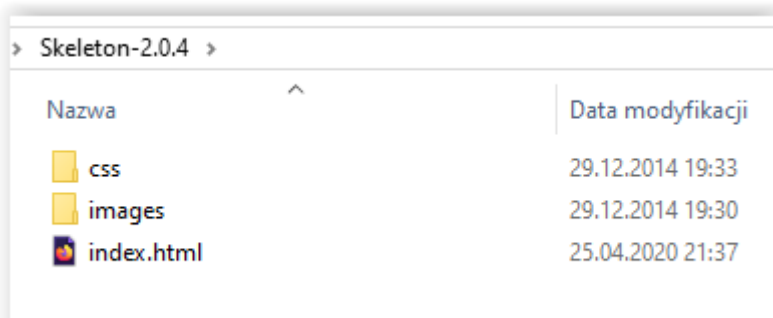
- Oszczędność czasu
- Społeczność i rozszerzalność
- Zgodność z różnymi przeglądarkami
- Dokumentacja

Są też i wady:

- Nadmiarowość kodu
- Czas nauki
- Zmniejszona elastyczność

Zadanie: **Stworzenie elastycznego portfolio przy użyciu systemu Skeleton**

Należy pobrać i rozpakować do wybranego folderu archiwum ze strony [www.getskeleton.com/](http://www.getskeleton.com/) (pobrany przeze mnie pakiet to 2.0.4). Po wypakowaniu zobaczymy taki układ folderów jak na fot. poniżej:



W opisywanym szablonie metaznacznik viewport ma parametry initial-scale ustawione na 1, jak widać poniżej, co spowoduje że przy pierwszym otwarciu strona zajmuje 100 procent dostępnego obszaru w oknie przeglądarki.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Po uruchomieniu pliku index.html zobaczymy taki oto widok:

# Basic Page

This index.html page is a placeholder with the CSS, font and favicon.  
It's just waiting for you to add some content! If you need some help hit  
up the [Skeleton documentation](#).



Skeleton jest wyposażony w skalowalną siatkę, dzięki której można szybko zbudować elastyczny układ strony.

Kolumny definiuje się za pomocą klasy `columns` i dodatkowej klasy określającej liczbę kolumn, np.: `one`, `three`, `four` itd., wyznaczającej szerokość tworzonej kolumny. Klasy te zawarte są w pliku `skeleton.css`.

```
.one.column,
.one.columns      { width: 4.66666666667%; }
.two.columns      { width: 13.3333333333%; }
.three.columns    { width: 22%;           }
.four.columns     { width: 30.6666666667%; }
.five.columns     { width: 39.3333333333%; }
.six.columns      { width: 48%;           }
.seven.columns    { width: 56.6666666667%; }
.eight.columns    { width: 65.3333333333%; }
.nine.columns     { width: 74.0%;         }
.ten.columns      { width: 82.6666666667%; }
.eleven.columns   { width: 91.3333333333%; }
.twelve.columns   { width: 100%; margin-left: 0; }

.one-third.column { width: 30.6666666667%; }
.two-thirds.column { width: 65.3333333333%; }

.one-half.column  { width: 48%; }
```



Treści na stronę możemy wygenerować za pomocą <https://www.lipsum.com/feed/html>



W systemie Skeleton wykorzystano zapytania medialne CSS3

```
/* Larger than mobile */
@media (min-width: 400px) {}

/* Larger than phablet (also point when grid becomes active) */
@media (min-width: 550px) {}

/* Larger than tablet */
@media (min-width: 750px) {}

/* Larger than desktop */
@media (min-width: 1000px) {}

/* Larger than Desktop HD */
@media (min-width: 1200px) {}
```

Nasza strona jest już elastyczna, ponieważ klasy i style kolumn są już zdefiniowane w zapytaniach medialnych w pliku skeleton.css.

Skeleton zawiera podstawowe style przycisków, które stosuje się poprzez dodanie klasy button do wybranych elementów, np. button albo a, jak poniżej:

```
<button class="button" type="submit">Element button</button>
<a href="#" class="button">Kotwica</a>
```



## Basic Page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus massa urna, luctus id bibendum et, semper sed ipsum. Vestibulum sed suscipit elit. Aliquam feugiat id neque eu dapibus. Nulla feugiat mi mi, sed congue magna semper quis. Vestibulum nec porttitor leo, sed lacinia arcu. Vestibulum ultricies sem eu consequat mollis. Mauris vel odio non justo efficitur scelerisque. In id bibendum diam. Fusce condimentum urna nec tellus porta feugiat. Nulla sed pellentesque nisl. Donec congue tristique elit at molestie. Nam aliquet quis mauris non dapibus. Curabitur lobortis placerat aliquet. Pellentesque ut mi tincidunt, lacinia lacus egestas, aliquet sapien. Suspendisse tempus tortor non erat finibus congue fermentum sed velit. Suspendisse ornare mi a nibh cursus blandit. Etiam diam ex, pulvinar sed eros id, accumsan feugiat magna. Mauris elit nisi, ultricies interdum elit in, rutrum rhoncus libero. Phasellus porttitor dictum molestie. Maecenas eget sagittis ligula, eu suscipit mauris. Nunc maximus velit ut augue rhoncus viverra. Curabitur feugiat metus ut pulvinar mollis. Nam id euismod turpis. Aliquam accumsan iaculis felis vitae accumsan. Pellentesque tristique nec massa vel pretium. .

ELEMENT BUTTON

KOTWICA

### Style formularzy

**Imię**

**Wiadomość**

**WYŚLIJ**

```
<form>
  <label for="name">Imię</label>
  <input type="text" id="name">
  <label for="message">Wiadomość</label>
  <textarea id="message"></textarea>
  <button type="submit">Wyślij</button>
</form>
```

Skeleton zawiera favikonę i ikony systemu iOS, które można bez problemu zamienić na własne.

### Zróbmy teraz coś swojego

Utwórzmy i podepnijmy własne pliki css.

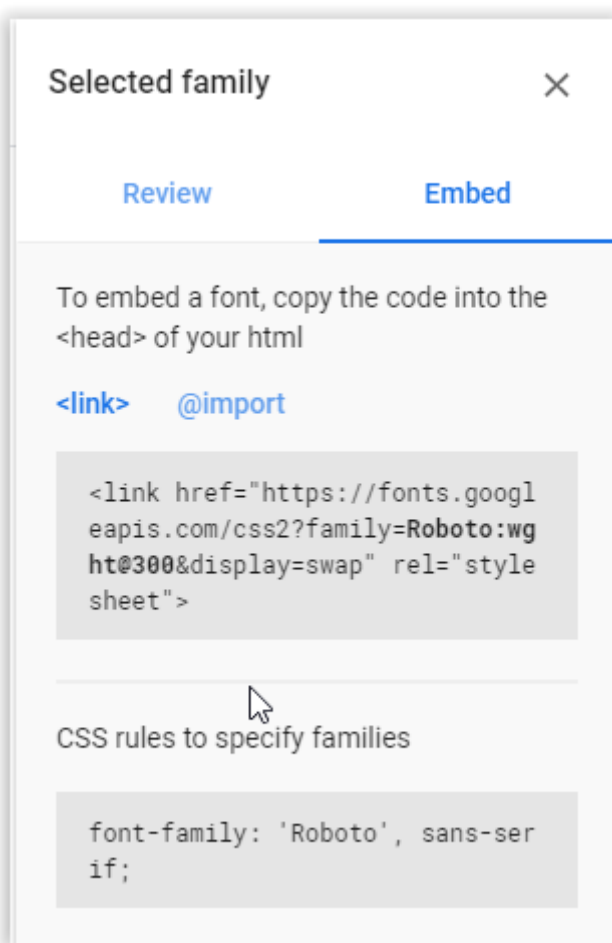
```

<!-- CSS
----- -->
<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="css/skeleton.css">

<link rel="stylesheet" href="css/base.css">
<link rel="stylesheet" href="css/skeleton.css">
<link rel="stylesheet" href="css/layout.css">
<link rel="stylesheet" href="css/styles.css">

```

Dodajmy własne fonty z <https://fonts.google.com>



### Przygotowanie własnych obrazów.

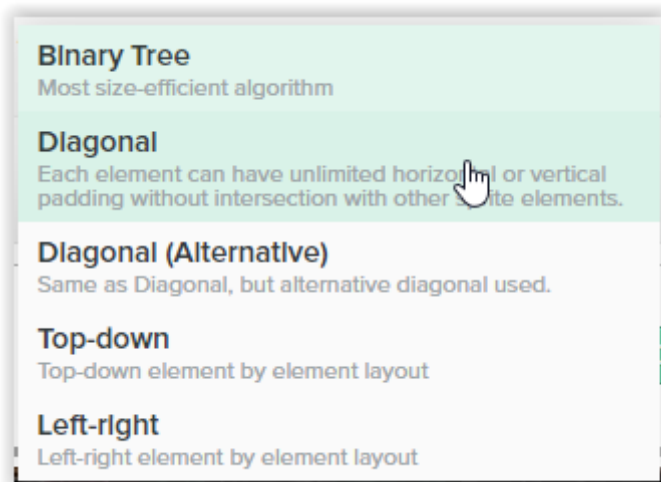
Do przygotowania własnych obrazów można wykorzystać program Photoshop lub inny do obróbki grafiki. Obrazy powinny mieć jednakowy rozmiar np. 480x480 px i ujednolicone nazewnictwo np. img-1.jpg; img-2.jpg, ..

Ikony mediów społecznościowych powinny być wykonane w ten sposób aby po najechaniu na nie kursorem zmieniały kolor na charakterystyczny dla danego portalu, czyli np. w przypadku Facebooka na niebieski 48 × 48 px.

## sprite'y

<https://www.toptal.com/developers/css/sprite-generator/>

masz tutaj do wyboru kilka opcji:



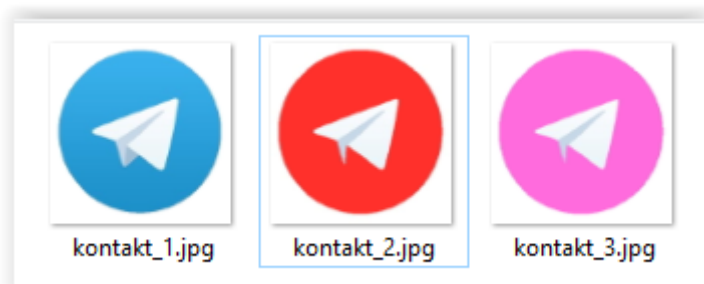
Zostanie wygenerowany obraz sprite'ów wraz z regułą CSS do jego wyświetlania.



Pobrany obraz zapisz w images a podany kod CSS do swojego pliku styles.css

## Ikony kontaktowe

ikony o wymiarach 24 × 24 pikseli



Jeśli ikony te występują oddzielnie, należy je połączyć w jeden plik i wygenerować odpowiednie reguły CSS w podobny sposób jak poprzednio.

## HTML data-\* Attributes

[https://www.w3schools.com/tags/att\\_global\\_data.asp](https://www.w3schools.com/tags/att_global_data.asp)

w HTML5 wprowadzono nowy atrybut do przechowywania danych. Za jego pomocą można osadzać dane w elementach HTML. Jego konstrukcja składa się z przedrostka data- i nazwy atrybutu.

## Konstruowanie dokumentu HTML

**Twoim zadaniem** będzie dokończenie przykładowe strony znajdującej się pod tym linkiem

[https://drive.google.com/open?id=1ZTSeKsSZyGwG0zucHPfNCaX3yKl\\_n1Lp](https://drive.google.com/open?id=1ZTSeKsSZyGwG0zucHPfNCaX3yKl_n1Lp)

Stronę trzeba dokończyć wykorzystując zagadnienia poruszone w części 1 i ostylować za pomocą CSS. Strona ma się oczywiście skalować dopasowując do różnej wielkości ekranu (**RWD**).

Po wykonaniu stron umieśćcie na GitHubie i wyślijcie do mnie link z możliwością podejrzenia kodu oraz pobrania plików.