



PREWORK

Back-end Developer

Cele

- ❑ Zapoznanie się z podstawowymi systemem kontroli wersji oprogramowania
- ❑ Nauka podstawowych technik programowania pozwalających na samodzielną realizację prostego bloga.
- ❑ Wprowadzenie w wirtualny świat programów, projektów w którym nauczysz się komunikować z maszyną.



Wprowadzenie do HTML

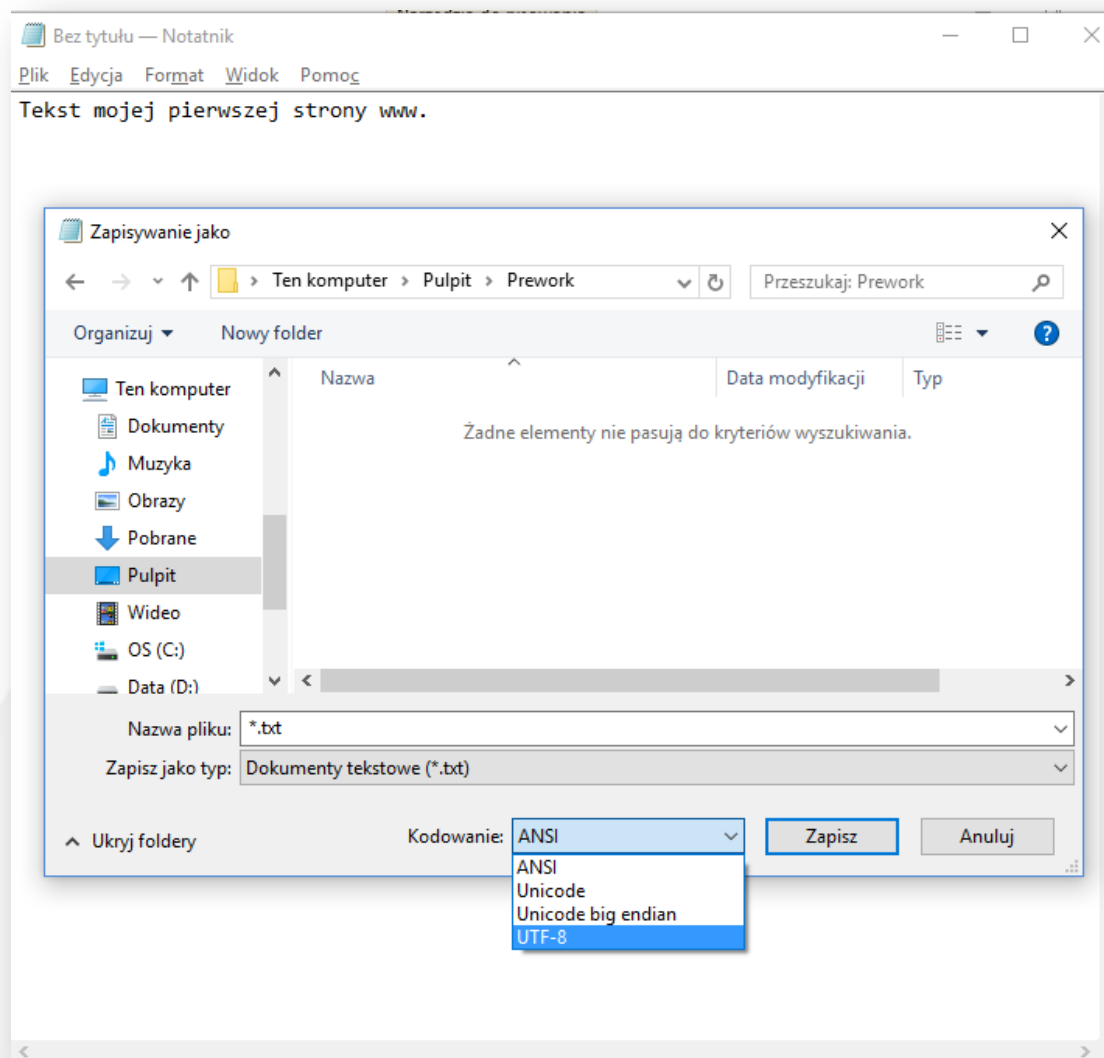
Czyli aby sprawnie realizować zadania back-endu
zaczynamy od poznania podstaw technik front-endowych.

- ❑ HTML (ang. Hyper Text Markup Language) to najbardziej popularna technologia w sieci, absolutnie konieczna do projektowania stron internetowych.
 - ❑ Hyper Text – dokumenty są interpretowane w ramach protokołu HTTP
 - ❑ Markup – dokumenty zawierają znaczniki które określają treści
 - ❑ Language – dokument jest interpretowany przez komputer
- ❑ W tym kursie będziemy omawiali zagadnienia związane z wersją najnowszą, czyli HTML5.

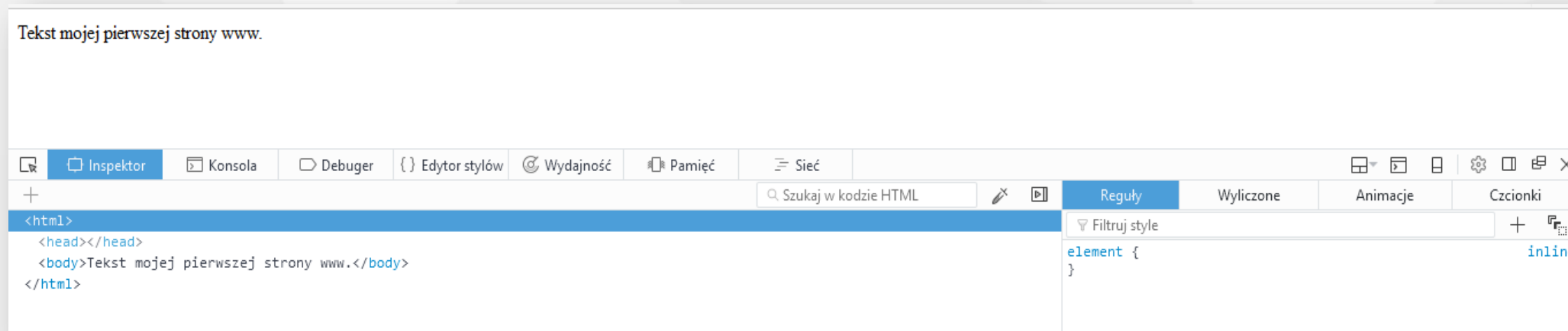
Jak zacząć przygodę z HTML

- ❑ Otwieramy dowolny edytor tekstu np.: Notatnik dostępny w systemach Windows bez konieczności instalacji.
- ❑ Wpisujemy do niego jakąś frazę np.: Tekst mojej pierwszej strony www i zapisujemy go jako plik z rozszerzeniem .html (np.: Stronka.html). Warto zmienić kodowanie na UTF-8!
- ❑ Takie rozszerzenie jest interpretowane przez przeglądarkę internetową i w ten sposób możesz otworzyć swoją pierwszą stronę internetową.

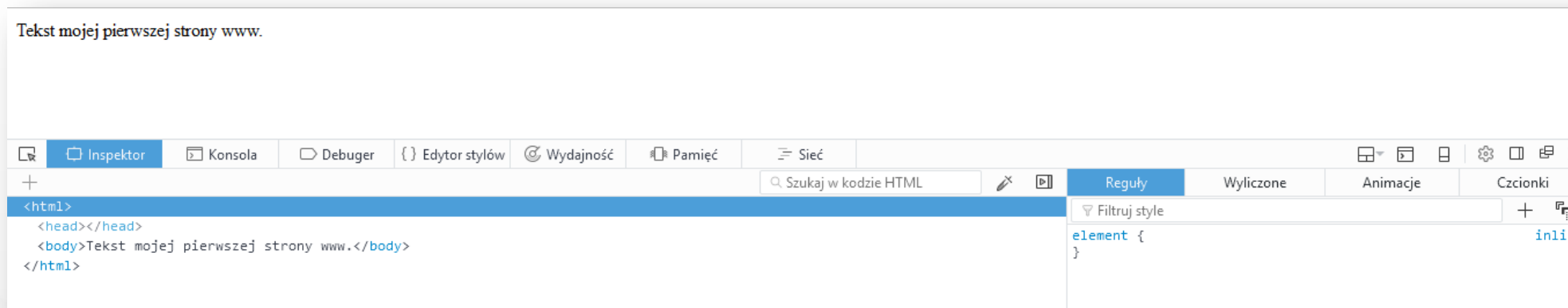
Jak zacząć przygodę z HTML



- ❑ W przeglądarce internetowej zawsze możemy zbadać źródło wyświetlanej strony internetowej klikając prawym klawiszem myszy na jej treści i wybierając polecenie: *Zbadaj*.



Znaczniki pierwszej strony



- ❑ Widzimy, że oprócz napisu widocznego w przeglądarce pojawiły się dodatkowe elementy – znaczniki HTML.
- ❑ Przeglądarka wygenerowała domyślnie znaczniki do tekstu zawartego w pliku Stronka.html, ale oczywiście nie powinniśmy tak robić!
- ❑ Powinniśmy skorzystać ze struktury HTML po to aby wyświetlić nasz kawałek tekstu.

Znacznik paragrafu

- ❑ Modyfikujemy tekst w pliku: Stronka.html opakowując go w znacznik `<p>` - paragraf (ang. paragraph) i `</p>` - zakończenie paragrafu.

`<p>Tekst mojej pierwszej strony www</p>`

- ❑ Zapisujemy plik otwieramy w przeglądarce i patrzymy co się stało oraz co ważniejsze badamy źródło – jak wcześniej.

Znaczniki nagłówków

- ❑ Tym razem dodajmy sobie jakieś nagłówki do naszej Stronki.html opakowując je znacznikami typu `<h>` - nagłówek (ang. header):

```
<h1> Nagłówek typu 1 </h1>
```

```
<h2> Nagłówek typu 2 </h2>
```

```
<p>Tekst mojej pierwszej strony www</p>
```

Nagłówek typu 1

Nagłówek typu 2

Tekst mojej pierwszej strony www

- ❑ W efekcie zauważamy, że każdy z nagłówków posiada inne style!? Dlatego że nasza przeglądarka posiada wbudowane style dla tak zdefiniowanych znaczników.

HTML



Edytor HTML

Wybór edytora

- ☐ Istnieje wiele różnych – mniej lub bardziej zaawansowanych edytorów HTML takich jak: Notepad ++, Sublime Text, PSPad, Aptana Studio, Brackets, czy niegdyś popularny Pajączek.
- ☐ Na tym kursie będziemy korzystali z *Notepad ++*. Dlaczego?
 - ☐ Prostota i wygodna.
 - ☐ Przejrzysty interfejs, szybkie uruchamianie i stabilna praca ułatwią tworzenie kodu źródłowego.
 - ☐ Podświetlanie składni wielu języków programowania.
 - ☐ Pozwala na wyszukiwanie i zmianę ciągów znaków za pomocą wyrażeń regularnych, tworzenie makr i własnych wtyczek.
 - ☐ Darmowy edytor oparty na licencji GNU GPL.
- ☐ Instalacja edytora jest dokładnie opisana w prezentacji dotyczącej oprogramowania potrzebnego na kursie Back-end Developer.

HTML



Składnia HTML

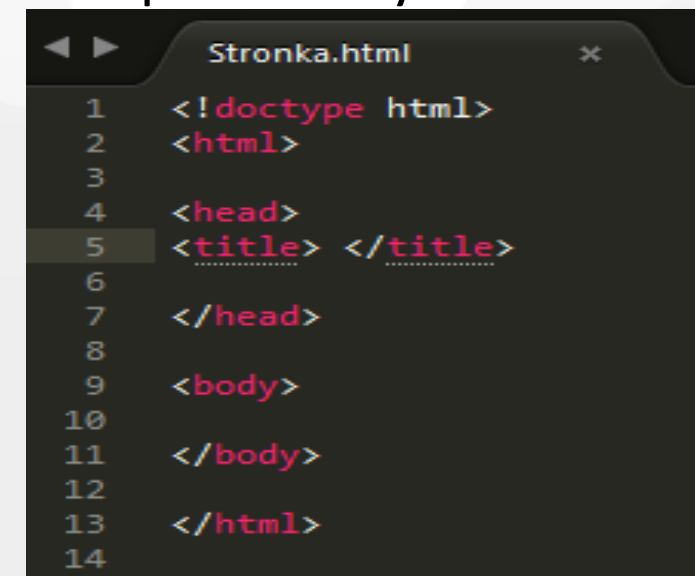
Składnia dokumentu HTML

☐ Zachowanie odpowiedniej składni pozwala na jego:

- | | |
|--------------------------------------|-----------------------|
| <input type="checkbox"/> Walidację | <code>!doctype</code> |
| <input type="checkbox"/> Debugowanie | <code>html</code> |
| <input type="checkbox"/> Czytelność | <code>head</code> |
| <input type="checkbox"/> Utrzymanie | <code>body</code> |

Podstawowa składnia dokumentu

- ❑ Każdy poprawny dokument HTML powinien zawierać następujące sekcje (określone znacznikami):
 - ❑ `<!doctype html>` informuje, że dokument ma być interpretowany w języku HTML
 - ❑ `<html>` cały dokument HTML
 - ❑ `<head>` sekcja nagłówka
 - ❑ `<title>` tytuł
 - ❑ `<body>` treść dokumentu



```
Stronka.html
1  <!doctype html>
2  <html>
3
4  <head>
5  <title> </title>
6  .....
7  </head>
8
9  <body>
10
11 </body>
12
13 </html>
14
```

Sekcja <head>

- ❑ W sekcji meta występują znaczniki:
 - ❑ <title>
 - ❑ <meta>
 - ❑ <style>
 - ❑ <link>

Kodowanie znaków w HTML

- ❑ Do kodowania znaków w HTML wykorzystywane jest polecenie:

<meta charset = "typ kodowania">

- ❑ W naszym przypadku wybieramy kodowanie utf-8 i znacznik ten dodajemy w pierwszej linii sekcji nagłówka (<head>) – wyłącznie tam on może występować:

<meta charset="utf-8">

Nazwa autora strony

Za pomocą znacznika meta możemy dodać nazwę autora strony

```
<meta name= " author" content= "nazwa autora">
```

Słowa kluczowe strony

- ❑ Niegdyś bardzo ważne w celu pozycjonowania wyników wyszukiwania strony przez przeglądarki.

<meta name="keywords" content="słowa kluczowe">

- ❑ Aktualnie słowa kluczowe straciły na wartości.

Opis strony

- ❑ Za pomocą znacznika meta możemy dodać opis strony, który pojawi się w przeglądarce podczas wyszukiwania (max. ok. 150 znaków)

<meta name="description" content="opis strony">

Home - Reaktor - Wydawnictwo Naukowe PWN

reaktor.pwn.pl/ ▼

Content

Kurs w Reaktor PWN to nie tylko bardzo intensywne warsztaty pod okiem trenera. ... Dwie osoby z każdego kursu, które przygotowują najciekawsze projekty, ...

Elementy blokowe

- ❑ Elementy blokowe to najważniejsze elementy struktury dokumentu na stronie.

Określają one ramy dokumentu:

- ❑ Paragrafy/akapity `<p>`
- ❑ Listy `` i ``
- ❑ Nagłówki `<h1>` ... `<h6>`
- ❑ Sekcje `<section>`
- ❑ Artykuły `<article>`

Elementy liniowe

- ❑ Elementy liniowe to najczęściej elementy znajdujące się wewnątrz elementów blokowych zawierające jakiś fragment tekstu, któremu nadają jakieś konkretne znaczenie.

❑ Elementy	<code></code>
❑ Linki	<code><a></code>
❑ Ważne treści	<code></code>
❑ Pochylenie	<code><i></code>
❑ Pogrubienie	<code></code>

Domykanie elementów

	Znacznik otwierający i zamykający	Samozamykanie
Elementy blokowe	<code><p> </p></code> <code> </code> <code> </code>	Nieemożliwe
Elementy liniowe	<code><a> </code> <code> </code> <code> </code>	<code><input></code> <code>
</code> <code></code>

Często stosowane znaki specjalne

Znak	Encja	Komentarz
&	&	ang. ampersand
<	<	ang. less than
>	>	ang. greater than
©	©	ang. copyright
®	®	ang. registered
™	™	ang. trademark
€	€	
£	£	
§	§	ang. section
¼	¼	ang. fraction
½	½	
¾	¾	

Znak	Encja	Komentarz
¹	¹	ang. superscript
²	²	
³	³	
°	°	ang. degree
α	α	
β	β	
γ	γ	
δ	δ	
ε	ε	
μ	μ	
π	π	
Ω	Ω	

HTML



Treści na stronie

Model DOM

- ❑ DOM, czyli obiektowy model dokumentu (ang. Document Object Model) jest to sposób reprezentacji złożonych dokumentów XML i HTML w postaci modelu obiektowego.
- ❑ Model ten jest niezależny od platformy i języka programowania.
- ❑ Standard W3C DOM definiuje zespół klas i interfejsów, pozwalających na dostęp do struktury dokumentów oraz jej modyfikację poprzez tworzenie, usuwanie i modyfikację tzw. węzłów (ang. nodes).

Dokumentacja do poczytania:

- <https://www.w3.org/DOM/>
- <https://www.w3.org/TR/html5/dom.html#kinds-of-content>

Najczęściej używanie znaczniki

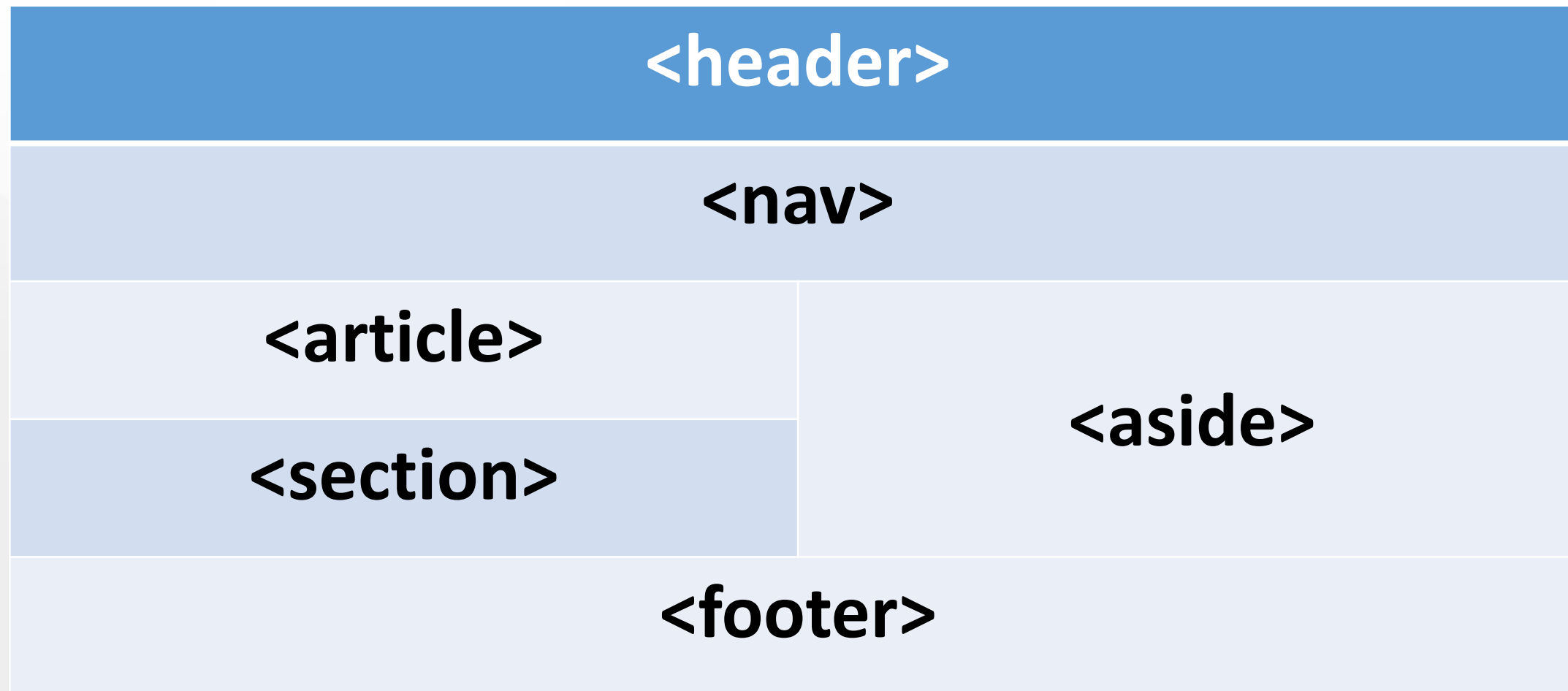
- ❑ Dokładniejszy opis znaczników HTML znajduje się pod adresem:

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

- ❑ Poniższa tabela zawiera najczęściej wykorzystywane znaczniki HTML:

Struktura	Treść	Linowe
header – nagłówek strony h1 h2 div nav – nawigacja strony footer – stopka strony article – artykuł strony section – sekcja strony	p ul ol li blockquote	a strong em q abbr span

Semantyka znaczników struktury



Znaczniki treści

- ☐ Otwórz plik o nazwie znaczniki treści.
- ☐ Występują tam następujące typy znaczników:
 - ☐ `<p>` akapit
 - ☐ `` lista nieuporządkowana
 - ☐ `` lista nieuporządkowana
 - ☐ `` element listy
 - ☐ `<blockquote>` cytata

Znaczniki liniowe

- ☐ Otwórz plik o nazwie znaczniki liniowe.
- ☐ Występują tam następujące typy znaczników:
 - ☐ `<a>` hiperłącze
 - ☐ `` pogrubienie
 - ☐ `` wyróżnienie semantyczne
 - ☐ `` wyróżnienie stylistyczne
 - ☐ `<abbr>` zaznaczenie skróconych form wyrazów
 - ☐ `<q>` cytaty w cudzysłowie

HTML



Obrazy na stronie

Ścieżki bezwzględne

- Aby jakiś obraz znajdujący się na już istniejącej stronie w sieci załączyć do naszej strony należy zastosować ścieżkę bezwzględną tego obrazu w znaczniku img:

``

Np.:

``

Ścieżki względne

- ❑ Aby załączyć obraz znajdujący się w tym samym miejscu na dysku co nasz plik .html:

**

- ❑ Obraz jednak może się znajdować w odrębnym katalogu:

**

- ❑ Obraz również może się znajdować w katalogu wyżej w hierarchii katalogów (przed plikiem .html):

**

Podstawowe atrybuty obrazów

- ❑ Obraz poza atrybutem źródła (src) może zawierać również inne atrybuty:
 - ❑ alt – opis obrazu
 - ❑ width - szerokość obrazu (w px lub % szerokości strony)
 - ❑ height - wysokość obrazu (w px lub % wysokości strony)
- ❑ Gdy zdjęcie pełni jakąś ważną funkcję objaśniającą na stronie warto opakować je w znacznik *<figure>*.

reaktor

Hiperłącza wychodzące poza domenę

- ❑ Hiperłącze wychodzące:

**

- ❑ Hiperłącze wychodzące z wymuszeniem otwarcia nowej zakładki:

**

Hiperłącza w obrębie domeny

- ❑ Hiperłącze do identyfikatora elementu elementu strony:

``

- ❑ Hiperłącze do pliku znajdującego się o w miejscu pliku na którym pracujemy:

``

- ❑ Hiperłącze do pliku w katalogu znajdującym się w miejscu pliku na którym pracujemy:

``

- ❑ Hiperłącze do pliku w katalogu znajdującym się wyżej w strukturze katalogów niż plik na którym pracujemy:

``



CSS

Czyli praca z wyglądem strony internetowej.

- ❑ CSS (ang. Cascading Style Sheets) to kaskadowy arkusz definiujący style dokumentów HTML.
- ❑ W chwili gdy dokument HTML jest już logicznie podzielony na części do nadawania stronom atrakcyjnego wyglądu służy CSS.
- ❑ Pozwala on określić, jak ma wyglądać dany element.
- ❑ Pozwala oddzielić strukturę (HTML) od jego wyglądu (warstwa prezentacji).
- ❑ CSS nie jest językiem programowania ani markupem.

- ❑ CSS składa się z regół wyglądających następująco:



- ❑ kto jaki element strony chcemy stylizować
- ❑ co co takiemu elementowi chcemy przypisać
- ❑ jak jak to chcemy zrobić (jaką chcemy przypisać wartość)

- ❑ Przypisanie do napisu nagłówka h1 czcionki czerwonej:

```
h1 { color: red }
```

- ❑ Przypisanie do identyfikatora o nazwie stopka fontu czcionki, marginesu (10px – góra i dół, 15px – prawy i lewy) i koloru czcionki:

```
#stopka {
```

```
font-family: 'Helvetica', 'Arial';
```

```
margin: 10px 15px
```

```
color: #09988
```

```
}
```

Więcej znajdziesz w dokumentacji CSS:

<https://www.w3schools.com/cssref/default.asp>

https://www.w3.org/standards/techs/css#w3c_all

reaktor

Wsparcie dla nowych technologii HTML i CSS w różnych przeglądarkach

- ❑ HTML i CSS jest cały czas rozwijamy standardem i czasem jest konieczne upewnienie się czy jakiś właściwość jest wspierana w danej przeglądarce.
- ❑ W tym celu możemy wykorzystać stronę www.caniuse.com gdzie możemy wpisać np.: flex i sprawdzić czy ta stosunkowo nowa właściwość CSS jest wspierana przez wszystkie przeglądarki.

Flexible Box Layout Module CR										Global
Method of positioning elements in horizontal or vertical stacks. Support includes the support for the all properties prefixed with flex as well as display: flex , display: inline-flex , align-content , align-items , align-self , justify-content and order .										unprefixed
										83.29% + 13.63% = 96.92%
										82.3% + 4.4% = 86.69%
Current status: Usage relative Show all										
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android	
			49					4.4		
8	13	47	51			9.2		4.4.4		
11	14	48	52	9.1	39	9.3	all	51	51	
		49	53	10	40					
		50	54	TP	41					
		51	55							



Połączenie HTML i CSS

Czyli pracujemy nad pierwszą stroną internetową.

Osadzanie stylów inline

- ❑ Wykorzystując atrybut style w kodzie HTML:

style="właściwość: wartość;"

Np.:

<p style="font-size: 50px; color: red;"> Tekst </p>

- ❑ Nie jest potrzebny selektor, ponieważ styl używamy wewnątrz konkretnego znacznika i nie ma sensu wyszczególniać do czego ma być zastosowany ten styl.

Osadzanie stylów w sekcji <head>

- ❑ Wykorzystując atrybut style w sekcji <head> dokumentu HTML:

```
<head>
```

```
<style type="text/css">
```

```
p { właściwość: wartość; }
```

```
</style>
```

```
</head>
```

- ❑ Tak zdefiniowane style będą działały do każdego (w tym przypadku) akapitu <p> w dokumencie HTML, który może występować w sekcji <body>.

- ❑ Zaczynamy od podpięcia pliku .css do naszego dokumentu HTML poprzez dodanie w sekcji <head> następującej komendy:

<link rel="stylesheet" type="text/css" href=„nazwa_pliku.css">

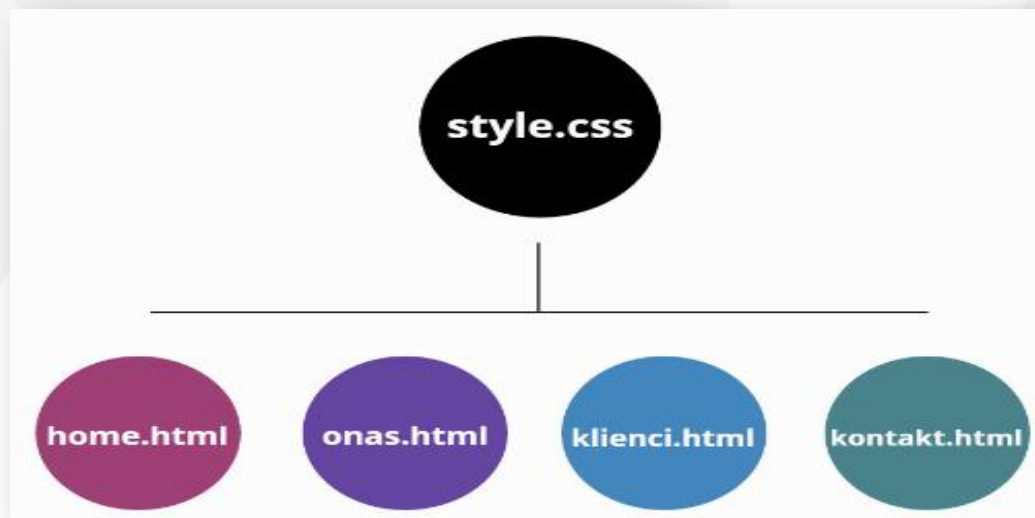
- ❑ Następnie tworzymy nowy plik z rozszerzeniem .css najlepiej w tym samym folderze co dokument HTML w którym osadzamy przykładową regułę:

znacznik { właściwość: wartość; }

Reaktor

Który sposób osadzania stylów jest najlepszy?

- ❑ Oczywiście najlepszym rozwiązaniem jest trzeci sposób osadzania stylów, czyli podpięcie osobnego pliku .css wraz ze zdefiniowanymi stylami dla znaczników HTML.
- ❑ Dzięki temu możemy jeden plik .css podłączyć do wielu podstron, a ewentualne zmiany w stylach będą wprowadzane automatycznie do wszystkich plików połączonych z naszym CSS.





Połączenie HTML i CSS

Selektory i cechy.

- ❑ Element HTML który chcemy sformatować za pomocą CSS.

Selektor znacznika (tagu)

- ❑ Znacznik (tag)
 - ❑ Pozwala na sformatowanie stylu dla konkretnego znacznika i wszystkie użycia tego znacznika w kodzie HTML są sformatowane identycznie.

- ❑ Plik .html

<znacznik> tekst <znacznik>

- ❑ Plik .css

znacznik { właściwość: wartość; }

Selektor klasy

- ❑ Klasa
 - ❑ Najważniejszy i najczęściej używany typ selektora w którym określamy nazwę do konkretnego elementu HTML, czyli taką nazwę roboczą dla której następnie przypisujemy style.

- ❑ Plik .html

<znacznik class="nazwa"> tekst <znacznik>

- ❑ Plik .css

.nazwa { właściwość: wartość; }

Selektor identyfikatora

- ❑ Identyfikator
 - ❑ Działający podobnie do selektora klasy z tą różnicą, że nazwa klasy może być taka sama dla wielu elementów, a identyfikator musi być unikatowy dla każdego elementu w którym jest zastosowany.

- ❑ Plik .html

<znacznik id="nazwa_id"> tekst </znacznik>

- ❑ Plik .css

#nazwa { właściwość: wartość; }

Sposoby tworzenia selektorów

☐ Prosty:

znacznik { właściwość: wartość; }

☐ Złożony:

znacznik.nazwa_klasy { właściwość: wartość; }

☐ Zagregowany:

h1, h2, h3, h4 { właściwość: wartość; }

Hierarchia stylów CSS



Hierarchia stylów ogólnie

Style w pliku HTML
w sekcji <body>

Style w pliku HTML w
sekcji <head>

Style w pliku zewnętrznym CSS

reaktor

Dlaczego warto o hierarchii pamiętać?

- ❑ Czasem programiści chcą wprowadzić jakąś tymczasową zmianę tylko dla jednego konkretnego przypadku. Wtedy wystarczy odnaleźć element w sekcji body i nadpisać go nowym stylem.
- ❑ Dzięki temu reszta podstron posiada nadal ujednolicone style, a nie ma potrzeby dla takiej zmiany tworzyć nowego pliku CSS i go podłączać do HTML.
- ❑ Oczywiście nie jest to zbyt dobra praktyka, ale czasem bywa użyteczna.

Konflikt w pliku CSS

- ❑ Konflikt między regułami w pliku CSS występuje wtedy gdy do tych samych selektorów przypisane zostały różne style:

Np.:

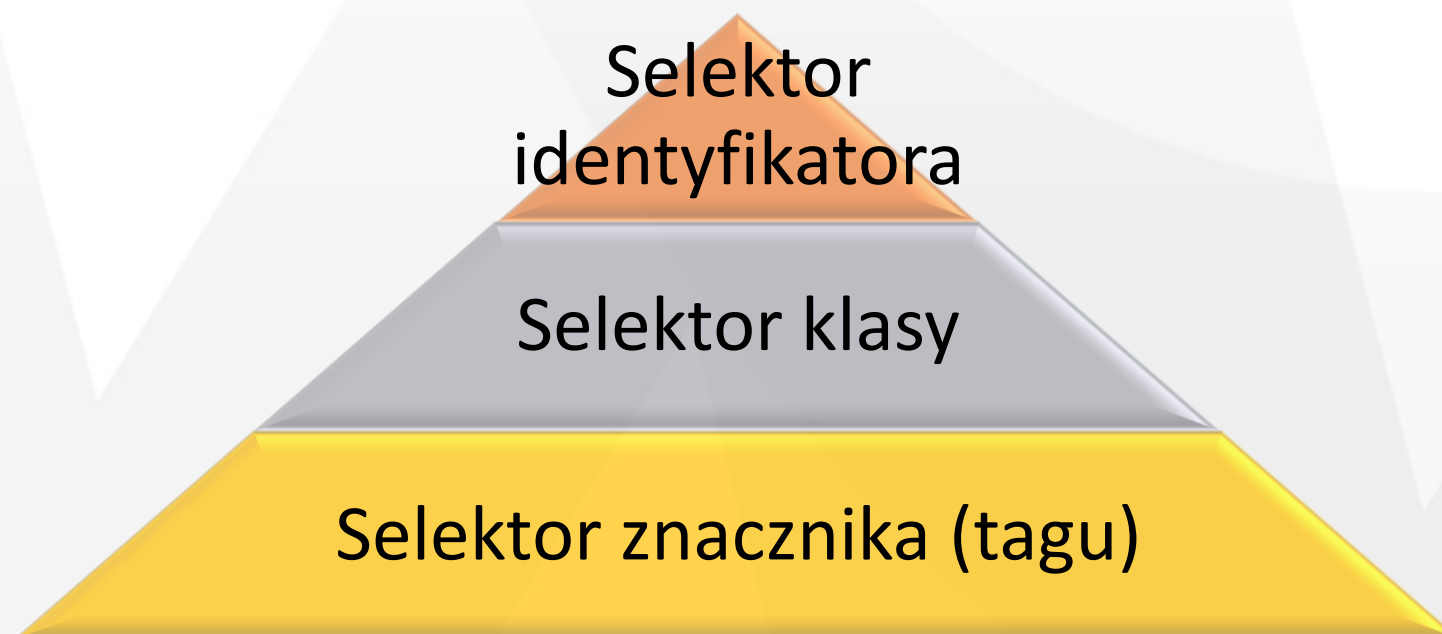
```
p { color: blue; }
```

```
p { color: red; }
```

- ❑ W sytuacjach konfliktowych pierwszeństwo ma reguła znajdująca się niżej w kodzie, ponieważ plik CSS jest interpretowany linijka po linijce od góry do dołu.
- ❑ Wobec tego niebieski kolor tekstu akapitu zostanie nadpisany kolorem czerwonym i ostatecznie tekst zostanie wyświetlony w tym kolorze. Ale czy zawsze tak jest?

Specyficzność

- ❑ W przypadku gdy występują specyficzne znaczniki, klasy czy identyfikatory to mamy do czynienia z następującą hierarchią ważności:



reaktor

Podsumowując hierarchie stylów CSS

WAŻNE STYLE UŻYTKOWNIKA

WAŻNE STYLE AUTORA

STYLE AUTORA

STYLE UŻYTKOWNIKA

STYLE PRZEGLĄDARKI

Style Inline

Style w <head>

Style w .css

#identyfikator

.klasa

tag

Importowanie stylów pomiędzy plikami CSS

- ❑ Aby zaimportować styl z pliku CSS (style2.css) do innego pliku CSS (style1.css), który jest podłączony do naszego dokumentu HTML, należy w pliku style1.css dodać polecenie:

```
@import url('style2.css');
```

- ❑ Teraz cała zawartość pliku style2.css została zaimportowana do pliku style1.css.

Prosta kalkulacja hierarchii

```
#news .header .lead p { color: black; }
```

121

```
#news .lead p a span { color: black; }
```

113

```
.sekcja .historie .news .lead p a span { color: black; }
```

043

Komentarze w CSS

- ❑ Przydatnym narzędziem są komentarze, które powodują, że zakomentowana część kodu nie jest interpretowana:

- ❑ **Komentarz jednowierszowy //**

Wszystko co znajduje się za znakiem // do końca linii nie jest interpretowane.

- ❑ **Komentarz blokowy (/* */)**

Wszystko co znajduje się pomiędzy znakami /* i */ (może być położone w wielu następujących po sobie wierszach kodu) nie jest interpretowane.

Dobre praktyki CSS

- ☐ Pisz jak najkrótsze selektory
- ☐ Pisz nazwy klas i identyfikatorów małymi literami, a wyrazy oddzielaj znakiem _
- ☐ Nie używaj nadmiernie identyfikatorów – są bardzo mocne w hierarchii. Lepiej jest pisać same klasy.
- ☐ Używaj intuicyjnych nazw klas
- ☐ Osadzaj style w pliku CSS
- ☐ Używaj resetowania stylów CSS (www.csstricks.com)



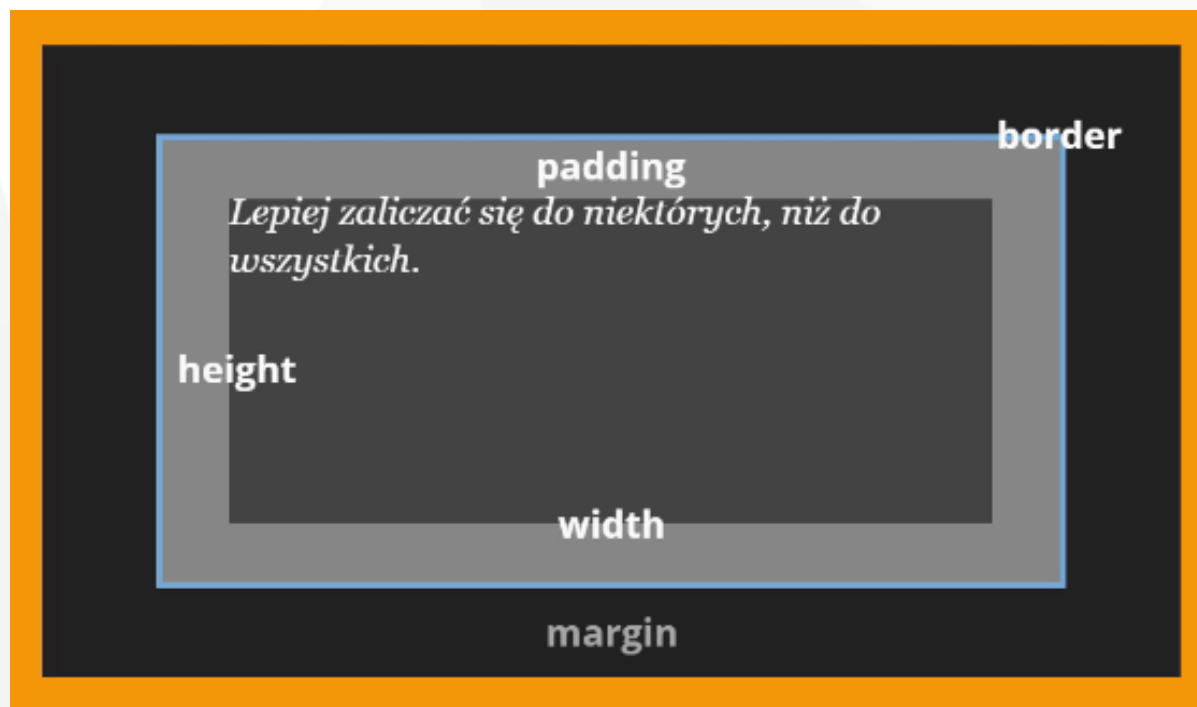
Połączenie HTML i CSS

Model pudełkowy.

reaktor

Czym jest strona internetowa w modelu pudełkowym?

- ❑ Zestaw bloków na ekranie - każdy znacznik który dodasz tworzy taki blok.
- ❑ Za pomocą CSS możemy dowolnie ustawiać wybrane właściwości takich blok.



Kolory w CSS

- ❑ Kolory w CSS są zestawem nazw (keyword) i odpowiadających im wartości (RGB hex values), które reprezentują poszczególne barwy.
- ❑ Dokładny opis znajduje się pod linkiem:
https://developer.mozilla.org/pl/docs/Web/CSS/color_value
- ❑ Wartości kolorów w CSS są zapisane w trybie szesnastkowym, tj. #123456 z czego pierwsze dwie cyfry odpowiadają za kolor czerwony (R), kolejne dwie za kolor zielony (G) i ostatnie dwie za kolor niebieski (B). W efekcie powstaje mieszanina trzech barw podstawowych, czyli kolor wypadkowy.

Palety barw

- ❑ W celu pomocy w doborze odpowiednich kolorów podczas projektowania stron internetowych warto odwiedzić stronę:

<https://color.adobe.com/pl/create/color-wheel/>

Definiowanie koloru tła

- Aby zdefiniować kolor tła dla naszej strony internetowej w pliku CSS w sekcji <body> wpisujemy poniższe polecenie:

<body>

background-color: keyword lub wartość;

</body>

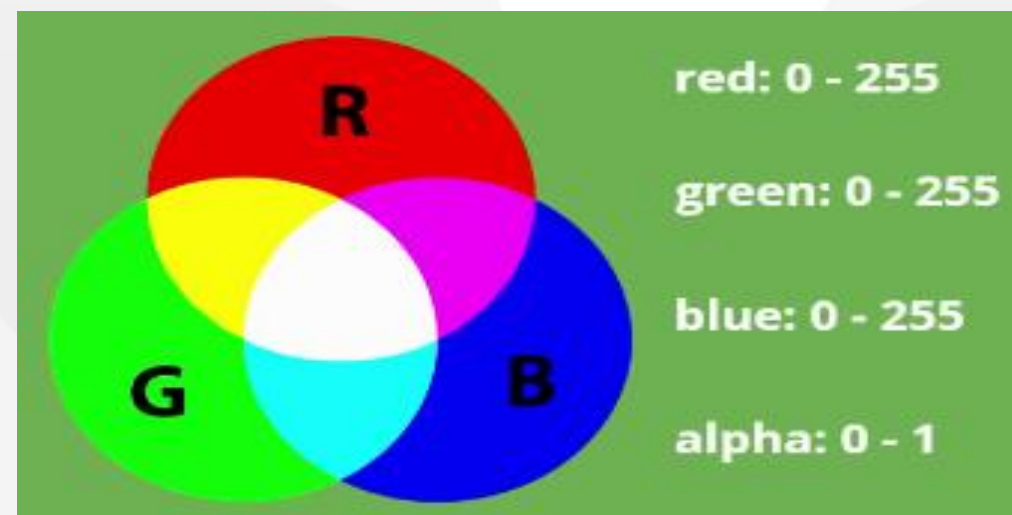
Efekt przezroczystości kolorów RGBA

- Aby dodać efekt przezroczystości musimy zdefiniować wartości dla kolorów w standardzie RGB oraz wartość przezroczystości A z przedziału 0-1, gdzie 1 to brak przezroczystości, a 0 to 100% przezroczystość.

```
<body>
```

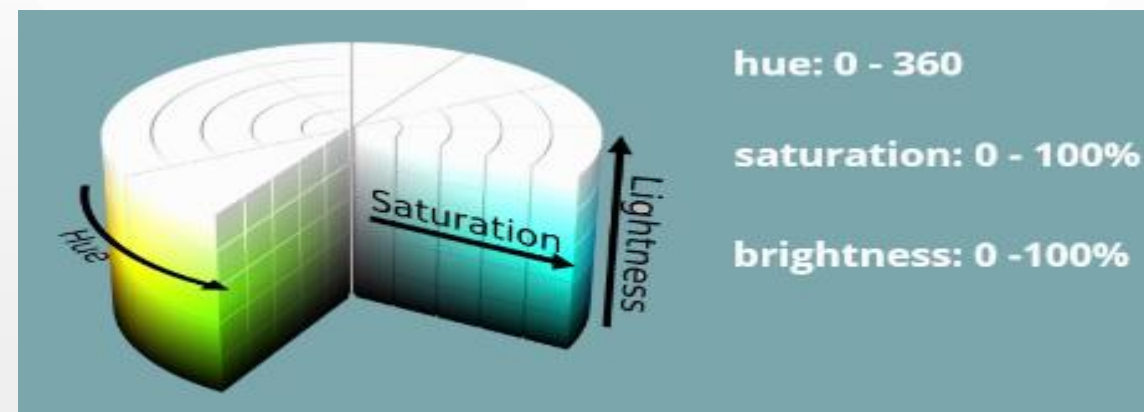
```
    color: rgba(20, 20, 10, 0.5);
```

```
</body>
```



Notacja HSLA

- ❑ Notacja HSLA definiuje kolory na podstawie 4 atrybutów:
 - ❑ Kolor
 - ❑ Nasycenie
 - ❑ Jasność
 - ❑ Przezroczystość



`<body>`

`color: hsla(100, 10%, 50%, 0.5);`

`</body>`

Generyczne rodziny krojów pisma

- ❑ Typy rodzin krojów pisma:
 - ❑ Serif
 - ❑ Sans-serif
 - ❑ Monospace
 - ❑ Cursive
 - ❑ Fantasy
- ❑ Nie jest to może bardzo bogata lista wyboru, ale zapewnia poprawność wyświetlania po stronie użytkownika.
- ❑ Pewnie mniej spotykane fonty mogą nie być zainstalowane w systemie użytkownika!

Właściwości tekstu

```
body {  
  font-family: Arial, Verdana, sans-serif;  
}
```

```
body {  
  font-size: 16px;  
}
```

```
body {  
  text-indent: 1em;  
}
```

```
body {  
  font-style: italic;  
}
```

```
body {  
  text-indent: 1em;  
}
```

```
body {  
  font-weight: bold;  
}
```

```
body {  
  text-align: right;  
}
```

```
body {  
  line-height: 1.5;  
}
```

```
body {  
  text-transform: uppercase;  
}
```

<https://www.gridlover.net/try>

Właściwości fontów

☐ Właściwości fontów:

- ☐ font-style
- ☐ font-variant
- ☐ font-weight
- ☐ font-size
- ☐ line-height
- ☐ font-family

Np.:

<body>

font: italic small-caps bold 12px/1.6 Arial, sans-serif;

</body>

Kolejność właściwości fontów

❑ /* size | family */

font: 2em "Open Sans", sans-serif;

❑ /* style | size | family */

font: italic 2em "Open Sans", sans-serif;

❑ /* style | variant | weight | size/line-height | family */

font: italic small-caps bolder 16px/1.6 cursive;

❑ /* style | variant | weight | stretch | size/line-height | family */

font: italic small-caps bolder condensed 16px/3 cursive;

Dodawanie cienia pod tekstem

- Aby dodać cień pod tekstem w akapicie:

```
p{ text-shadow: 1px 1px 0 rgba(0,0,0,0.5); }
```

gdzie:

1px

przesunięcie cienia w poziomie

1px

przesunięcie cienia w pionie

0

rozmycie cienia

rgba()

kolor w notacji rgba

Bezpieczne fonty

- ☐ Bezpieczne fonty możesz bez przeszkód używać na stronie i w każdej aplikacji, ponieważ są one preinstalowane w systemie użytkownika na każdej platformie, tj.:
 - ☐ Arial
 - ☐ Courier New, Courier
 - ☐ Garamond
 - ☐ Georgia
 - ☐ Lucida
 - ☐ Tahoma
 - ☐ Times New Roman, Times
 - ☐ Trebuchet
 - ☐ Verdana
 - ☐ Palatino

Font face

- ☐ Podstawowe fonty są bardzo oklepane, ale można równie bezpiecznie wykorzystywać inne ciekawe fonty.
- ☐ Reguła font face pozwala osadzić w naszej witrynie dowolny font. Jak?
 - ☐ Generuje specjalne pliki z fontem i przechowuje na serwerze
 - ☐ Przeglądarka użytkownika nawet jeśli nie zna tych fontów będzie korzystała z plików z serwera i renderowała napisy we wskazanym foncie – bez instalacji.
 - ☐ Ważne! Muszą na to zezwalać zapisy licencyjne fonta.

Stosujemy nietypowy font

- ❑ Pobieramy font z licencją np. ze strony: www.fontsquirrel.com
- ❑ Kopiujemy plik z fontem do katalogu gdzie znajduje się nasz plik HTML.
- ❑ Importujemy font za pomocą font face w pliku CSS:

```
@font-face{
```

```
    font-family: 'MojFont';
```

```
    src: url('nazwa_fonta.ttf');
```

```
}
```

- ❑ Odwołujemy się do tego fonta za pomocą nadanej mu nazwy 'MojFont'

Hostowane fonty

- ❑ Fonty znajdujące się w specjalnych serwisach (www.fonts.google.com, www.dafont.com, www.tipekit.com), które pozwalają podłączenie konkretnego fontu do naszej strony.
- ❑ Korzystają najczęściej z fragmentów kodu JavaScript na naszej stronie.
- ❑ Plik .html:

```
<link href="https://fonts.googleapis.com/css?family=Playfair+Display" rel="stylesheet">
```

- ❑ Plik .css:

```
p { font-family: 'Playfair Display', serif; }
```

Tło w CSS

- ❑ Wypełnienie tła

- ❑ Jednolity kolor

- background-color: kolor;***

- ❑ Gradient

- background-image: linear-gradient(kolor1, kolor2);***

- ❑ Obraz (uwaga na rozmiar!)

- background-image: url(adres_obrazu);***

❑ Odsyłam do zapoznania się z dokumentacją:

www.colorzilla.com/gradient-editor

Obrazek w tle

- ❑ Notacja obrazka w tle
 - ❑ `background-image: url(adres_obrazu);`
 - ❑ `height: wartość;`
 - ❑ `weight: wartość;`
 - ❑ `background-repeat [no-repeat, repeat-x, repeat-y];`
 - ❑ `background-position [center, left, right, top, bottom] lub [wartości px];`
 - ❑ `background-size: [cover, contain];`

- ❑ Istnieje także skrócona notacja



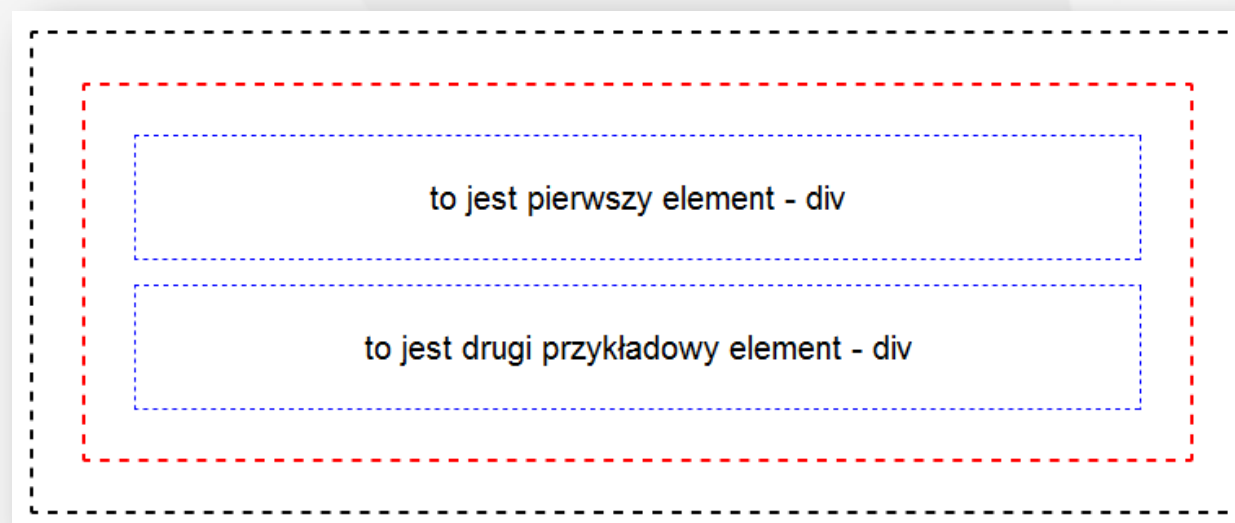
Pozycjonowanie elementów na stronie

Pozycjonowanie elementów strony

- ❑ W realnym świecie, gdy układamy jedną kartkę papieru na drugiej kartce papieru, gdy powtarzamy tę czynność z kilkoma kartkami papieru, to tworzymy **stos** z kartek papieru ułożonych na sobie nawzajem.
- ❑ W świecie wirtualnym, gdy umieszczamy jeden element HTML w drugim, gdy powtarzamy tę czynność z kilkoma elementami HTML, to tworzy się coś, co moglibyśmy nazwać **stosem elementów HTML**.

Przykład stosu HTML

- W przykładzie dołączonym do prezentacji *Pozycjonowanie.html*, do elementu body został dodany element *div*, do którego z kolei zostały dodane kolejne dwa elementy *div*. Do niektórych elementów zostały dodane marginesy wewnętrzne *padding* oraz różne obramowanie *border*. Do wszystkich elementów *div* został dodany biały kolor tła *background-color:white*; oraz wartość wysokości wierszy *line-height:60px*; a do elementu *div#dziecko1* dodatkowo został dodany dolny margines zewnętrzny *margin-bottom:15px*; dzięki czemu element *div#dziecko2* został odsunięty o 15 pikseli od elementu *div#dziecko1*.



Zmiana pozycji elementów HTML

- ❑ Domyślnie, elementy HTML są wyświetlone w pozycji statycznej, co oznacza, że **nie** możemy zmienić ich pozycji początkowej. Aby zmienić pozycję początkową elementu HTML, wykorzystuje się do tego celu następujące właściwości CSS:
 - ❑ **top** - odsunięcie od góry
 - ❑ **right** - odsunięcie od prawej
 - ❑ **bottom** - odsunięcie od dołu
 - ❑ **left** - odsunięcie od lewej
- ❑ Jednak aby wyżej wymienione właściwości CSS mogły przesunąć dany element HTML, nie może on być wyświetlony w domyślnej pozycji statycznej.

Właściwości pozycji

- ❑ Do zmiany pozycji, w jakiej ma być wyświetlony element HTML, wykorzystuje się właściwość *position* oraz jedną z następujących wartości:
 - ❑ **relative** - pozycja relatywna
 - ❑ **absolute** - pozycja absolutna
 - ❑ **fixed** - pozycja ustalona

- ❑ Gdy element HTML będzie wyświetlony w pozycji innej niż statyczna, czyli w pozycji relative, absolute lub fixed, to nie będzie on wpływał, po przesunięciu go, na położenie sąsiednich elementów HTML, tzn. nie będzie on odsuwał ich od siebie.

Pozycja relatywna

- Spójrzmy co się stanie z układem elementów HTML, gdy do elementu *#rodzic* zostanie dodana właściwość *position:relative;* oraz właściwość *right:100px;*

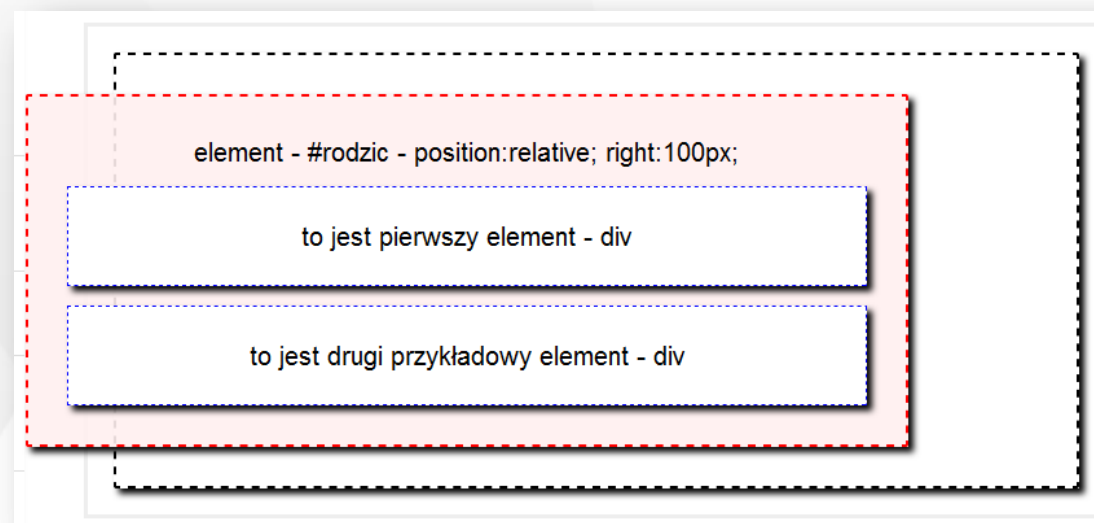
```
#rodzic {
```

```
border:2px dashed red;
```

```
position:relative;
```

```
right:100px;
```

```
}
```

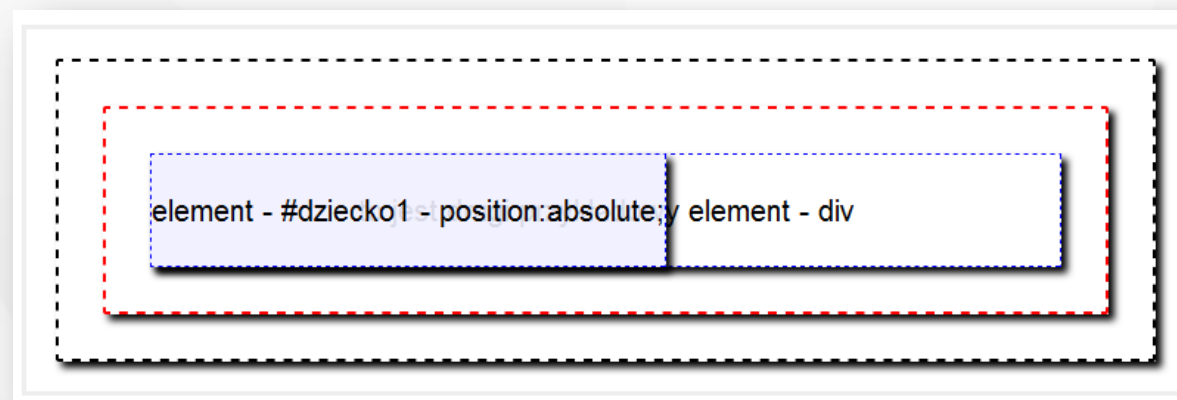


- Pozycję elementu relatywnego możemy kontrolować za pomocą właściwości *top*, *right*, *bottom*, *left*.

Pozycja absolutna

- Spójrzmy co się stanie z naszym przykładowym elementem *#dziecko1*, gdy dodamy do niego właściwość *position:absolute*;

```
#dziecko1 {  
  
    border:1px dashed blue;  
  
    margin-bottom:15px;  
  
    position:absolute;  
  
}
```



- Domyślnie, element HTML, do którego zostanie dodana właściwość *position:absolute*; nie pozostawia po sobie żadnej dodatkowej przestrzeni, dzięki czemu elementy, które go otaczały, nie reagują w żaden sposób na niego oraz na jego pozycję początkową.

Pozycja absolutna - właściwości

- ❑ Domyślnie, element wyświetlony z pozycji absolutnej będzie znajdował się w miejscu swojej pozycji początkowej, chyba że zmienimy to za pomocą właściwości CSS, które służą do ustalenia pozycji takiego elementu HTML, czyli za pomocą właściwości *top*, *right*, *bottom* lub *left*.
- ❑ Domyślnie, pozycja elementu absolutnego będzie liczona względem krawędzi okna naszej przeglądarki internetowej.
- ❑ Domyślnie, szerokość elementu pozycjonowanego absolutnie, czyli takiego elementu HTML, do którego została dodana właściwość *position:absolute*; jest uzależniona od zawartości jaka w tym elemencie zostanie umieszczona.
- ❑ Jeżeli chcemy, aby szerokość elementu pozycjonowanego absolutnie zajmowała całą szerokość okna naszej przeglądarki internetowej, to do takiego elementu pozycjonowanego absolutnie należy dodać właściwość *width:100%*; natomiast sam element pozycjonowany absolutnie musi być ustawiony w lewym górnym rogu okna przeglądarki oraz nie może posiadać w sobie takich właściwości CSS jak: *margin*, *padding*, *border*, po swojej prawej lub lewej stronie.

Pozycja ustalona

- ❑ Element wyświetlony w pozycji *fixed* zachowuje się podobnie jak element HTML, który jest wyświetlony w pozycji *absolute*, lecz jego pozycja zawsze jest liczona względem okna przeglądarki internetowej, a sam element wyświetlony w pozycji *fixed* zawsze znajduje się w tym samym miejscu, w oknie naszej przeglądarki internetowej, nawet jeżeli będziemy przesuwali nasze okno przeglądarki internetowej, np. za pomocą poziomego lub pionowego suwaka.
- ❑ Ponadto gdy umieścimy jakiś element HTML poza prawą krawędzią lub dolną krawędzią okna naszej przeglądarki internetowej, np. za pomocą właściwości *top*, *right*, *bottom*, *left* oraz wartości ujemnych, np. -30px, to do okna naszej przeglądarki internetowej zostaną dodane poziome bądź pionowe suwaki, które służą do przewijania niemieszczącej się zawartości, która pojawiła się w oknie przeglądarki internetowej, lecz opisana sytuacja nie będzie miała miejsca, gdy elementem, który został umieszczony poza granicami okna przeglądarki internetowej, jest element wyświetlony w pozycji *fixed*.



Praktyczny projekt

Zacznijmy od prostej strony domowej

- ❑ Utwórz stronę o sobie lub interesującej Cię tematyce. Może to być również prosty blog – liczę na Twoją kreatywność!
- ❑ Możesz wykorzystać dowolne poznane techniki, ale nie korzystaj z gotowych templatek stron.
- ❑ Do realizacji tego projektu wygeneruj plik lub pliki HTML do budowy strony oraz plik CSS do zdefiniowania stylów na stronie.
- ❑ Podczas pracy nad stroną wykorzystaj Gita i umieść finalne pliki źródłowe swojej pierwszej strony w repozytorium GitHub.

Przykładowa strona



PWN rusza z kursami programowania! W ciągu kilku tygodni przygotuje Cię do pracy w nowym zawodzie.



Wydawnictwo Naukowe PWN właśnie ruszyło ze szkołą oferującą kursy programowania opartą o metodę boot camp. To okazja dla wszystkich zainteresowanych branżą IT. Mentorzy **Reaktora PWN** podczas kilkutygodniowego kursu nauczą ich programować. Średnie zarobki na stanowisku programisty w Polsce sięgające ponad sześciu tysięcy złotych brutto czy ogromna potrzeba rynkowa zatrudnienia takich specjalistów to tylko kilka z wielu argumentów za tym, by nauczyć się kodowania. Jeśli nie jesteś zadowolony z obecnej pracy, a interesujesz się programowaniem i nigdy nie wiedziałeś, jak zacząć, to oferta **Reaktora PWN** jest dla Ciebie.

Metoda BootCamp, czyli szybka i intensywna nauka programowania?



Najlepszym sposobem na naukę programowania jest intensywny trening pod okiem ekspertów. Taki właśnie jest boot campowy kurs, dzięki któremu poznasz tajniki pracy programisty i nauczysz się tworzyć strony internetowe i aplikacje. Jak kurs przygotowany przez specjalistów z **PWN** wygląda od środka? Zajęcia odbywają się stacjonarnie, dzięki czemu szybciej i łatwiej przyswoisz wiedzę niż na kursach online. Każdy dzień rozpoczyna się od wstępu merytorycznego, który jest niezbędny do wykonywania poszczególnych projektów. Podczas zajęć kursanci skupiają się na kodowaniu, by najszybciej przyswoić przekazaną im wiedzę. Na tym właśnie polega metoda boot camp, z k



**Dodatkowe ćwiczenia
przygotowujące do kursu**

Propozycje do przećwiczenia baz danych

MySQL:

<http://www.w3resource.com/mysql/mysql-tutorials.php>

Ćwiczenia:

<http://www.w3resource.com/mysql-exercises/>

Propozycje do przećwiczenia podstaw programowania

Python:

<http://www.w3resource.com/python/python-tutorial.php>

Ćwiczenia:

www.w3resource.com/python-exercises/