

Tworzenie Stron Internetowych

odcinek 9

The background of the slide is black, featuring a complex pattern of glowing, multi-colored lines. These lines, in shades of orange, red, and white, radiate from a central point on the right side of the frame, creating a sense of dynamic movement and depth. The lines vary in thickness and brightness, some appearing as sharp, bright streaks while others are more diffuse and ethereal.

CSS – menu nawigacyjne

Przyjazne dla użytkownika menu nawigacyjne to jedna z najważniejszych cech dobrej strony/witryny/portalu internetowego. Menu nawigacyjne (lista odnośników) można zbudować wykorzystując listę nieuporządkowaną `` (poprawność semantyczna) lub bez użycia listy.

Przykład 1 (z ``):

```
<ul>
  <li><a href="index.html">strona główna</a></li>
  <li><a href="metody.html">metody pomiarowe</a></li>
  <li><a href="odnosniki.html">odnośniki</a></li>
  <li><a href="kontakt.html">kontakt</a></li>
</ul>
```

Przykład 2 (bez ``):

```
<div>
  <a href="index.html">strona główna</a>
  <a href="metody.html">metody pomiarowe</a>
  <a href="odnosniki.html">odnośniki</a>
  <a href="kontakt.html">kontakt</a>
</div>
```

CSS – menu nawigacyjne

Wgląd menu ustalamy wykorzystując CSS (menu poziome).

```
#menu {
```

```
  background-color: silver;
```

```
  overflow: hidden;
```

```
  width: 100%;
```

```
}
```

kolor tła

ukrywanie zawartości niemieszczącej się

szerokość menu

```
#menu a {
```

```
  display: block;
```

```
  float: left;
```

```
  color: black;
```

```
  text-align: center;
```

```
  padding: 14px 16px;
```

```
  text-decoration: none;
```

```
  font-size: 15px;
```

```
  border-right: 1px solid white;
```

```
}
```

wyświetlanie elementów <a> jako blokowych

ustawienie elementów listy w jednej linii

kolor tekstu

ustawienie tekstu

odstęp tekstu od obramowania

usunięcie domyślnego podkreślenia odnośników

rozmiar czcionki

prawy margines oddzielający przyciski menu

```
#menu a:hover {
```

```
  background-color: #ddd;
```

```
  color: maroon;
```

```
}
```

kolor tła i tekstu dla przycisku menu po wskazaniu go myszą

CSS – menu nawigacyjne

Wgląd menu ustalamy wykorzystując CSS (menu pionowe).

```
#menu {  
    background-color: silver;  
    overflow: hidden;  
    width: 100%;  
}
```

kolor tła
ukrywanie zawartości niemieszczącej się
szerokość menu

```
#menu a {  
    display: block;  
    float: none;  
    color: black;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;  
    font-size: 15px;  
    border-right: 1px solid white;  
}
```

wyświetlanie elementów <a> jako blokowych
ustawienie elementów listy w pionie
kolor tekstu
ustawienie tekstu
odstęp tekstu od obramowania
usunięcie domyślnego podkreślenia odnośników
rozmiar czcionki
prawy margines oddzielający przyciski menu

```
#menu a:hover {  
    background-color: #ddd;  
    color: maroon;  
}
```

kolor tła i tekstu dla przycisku menu po wskazaniu go myszą

CSS – menu nawigacyjne

A jeśli menu poziome jest za szerokie przy wyświetlaniu na małym urządzeniu?

Dodajemy do <head>:

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

a do CSS:

```
@media screen and (max-width: 970px)  
{ #menu a {float: none;} }
```

Wyjaśnienie za chwilę.

CSS – układ strony

Strona internetowa często dzielona jest na kilka obszarów o różnej zawartości

Przykładowy układ strony

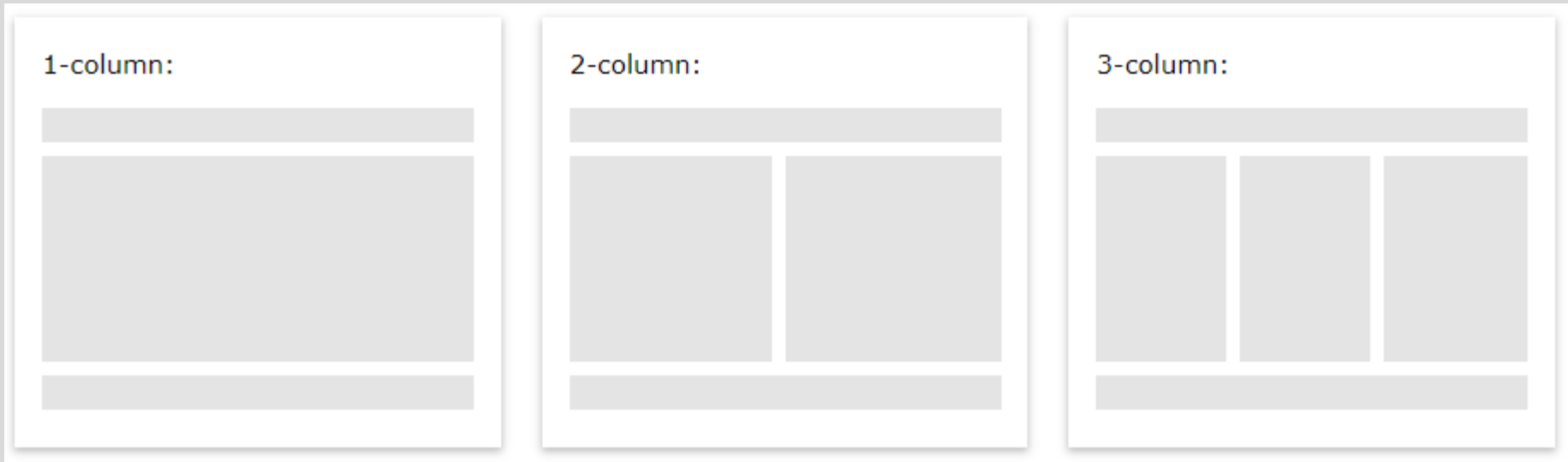


układ na duży wyświetlacz (układ desktopowy)

CSS – układ strony

Przykładowe układy strony.

Układ danej strony może zmieniać się w zależności od urządzenia wyświetlającego.



smartfon

tablet / laptop

PC

CSS – układ strony

Wykorzystując znacznik **<div>** z atrybutem **id** lub **class** lub nowe elementy blokowe (patrz następny slajd) możemy podzielić stronę na obszary takie jak: menu nawigacyjne, zawartość główna, zawartość poboczna, stopka, itd, czyli zbudować układ strony. Następnie w CSS nadajemy styl poszczególnym obszarom, układając je przy tym w odpowiednią siatkę.

Przykładowy prosty układ czteroobszarowy (jednokolumnowy) na blokach div:

```
<div id="header"> nagłówek </div>  
<div id="menu"> menu nawigacyjne </div>  
<div id="content"> zawartość strony </div>  
<div id="footer"> stopka </div>
```

Nagłówek

Umieszczony na samej górze lub pod menu nawigacyjnym. Dobre miejsce na logo lub nazwę strony WWW, duże zdjęcie.

Menu nawigacyjne

Zawiera poziome/pionowe menu nawigacyjne umożliwiające poruszenie się po pomiędzy poszczególnymi stronami tworzącymi serwis internetowy.

Zawartość strony

Tu znajduje się główna zawartość strony. Zawartość ta może być podzielona na mniejsze części.

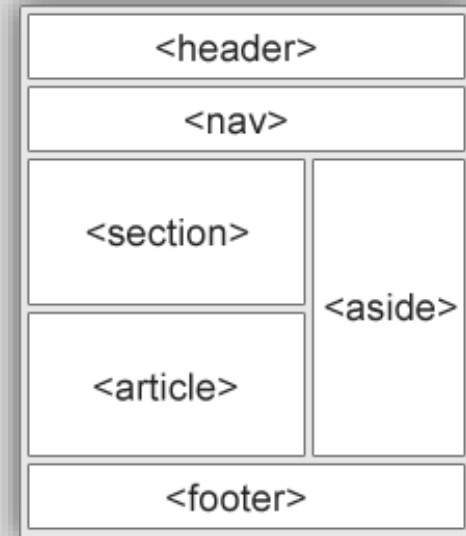
Stopka

Miejsce na notkę o prawach autorskich, kontakt z autorem, info o polityce prywatności, itp.

CSS – układ strony

W HTML5 dostępne są nowe (blokowe) elementy semantyczne służące budowaniu układu stron internetowych. Te elementy to:

- **<header>** – zawiera nagłówek strony, wprowadzenie do niej (może być tekst lub obraz)
- **<nav>** – definiuje zestaw odnośników będących menu nawigacyjnym
- **<main>** – element z główną, unikalną zawartość strony
- **<section>** – element definiujący części dokumentu (np. rozdział)
- **<article>** – element zawierający treść niezależną, mogącą istnieć bez reszty strony (np. wpis na forum, komentarz, artykuł prasowy)
- **<aside>** – definiuje zawartość poboczną, ale związaną tematycznie z resztą strony (np. przypisy)
- **<footer>** – zawiera informację na temat samej strony (np. podaje autora, zasady wykorzystania treści, dane kontaktowe, ...)



CSS – układ strony

Przykładowy układ dwukolumnowy. Na początek kod HTML.

```
<div id="header">  
  <h1>Nagłówek</h1>  
  <p>To jest strona testowa z układem dwukolumnowym</p>  
</div>
```

```
<div id="navmenu">  
  <a href="#...">Link1</a>  
  <a href="#...">Link2</a>  
  <a href="#...">Link3</a>  
</div>
```

```
<div id="row">  
  <div id="column1">  
    <h2>Aktualności</h2>  
    <p>Zawartość aktualności...Lorem ipsum dolor sit amet, consectetur adipiscing elit... </p>  
  </div>
```

```
<div id="column2">  
  <h2>Treść główna</h2>  
  <p>Zawartość kolumny głównej...Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>  
</div>  
</div>
```

CSS – układ strony

Przykładowy układ dwukolumnowy. Reguły CSS.

```
body { margin: 0; }
```

zerowanie marginesów (usunięcie domyślnych ustawień przeglądarki)

```
/* header */  
#header { background-color: #f1f1f1;  
padding: 20px;  
text-align: center; }
```

blok kodu dla obszaru nagłówka

```
/* navmenu */  
#navmenu { overflow: hidden;  
background-color: #333; }
```

1 blok kodu dla menu nawigacyjnego

```
/* navmenu linki*/  
#navmenu a { float: left;  
display: block;  
color: #f2f2f2;  
text-align: center;  
padding: 14px 16px;  
text-decoration: none; }
```

2 blok kodu dla menu nawigacyjnego

```
/* navmenu linki dla hover */  
#navmenu a:hover { background-color: #ddd;  
color: black; }
```

3 blok kodu dla menu nawigacyjnego

CSS – układ strony

Przykładowy układ dwukolumnowy. Reguły CSS.

```
/* dwie kolumny obok siebie */
```

```
#column1 { float: left;  
width: 25%;  
padding: 15px; }
```

```
#column2 { float: left;  
width: 75%;  
padding: 15px; }
```

```
/* usunięcie float po kolumnach */
```

```
#row::after {  
content: "";  
display: block;  
clear: both; }
```

blok kodu dla kolumny 1 o szerokości 25% szerokości całkowitej strony

blok kodu dla kolumny 2 o szerokości 75% szerokości całkowitej strony

zdeklarowanie float: left umożliwia ustawienie kolumn obok siebie

blok kodu usuwający ustawianie obszarów strony obok siebie (obok powyższych dwóch kolumn)

responsywność

Responsive Web Design (RWD) – technika projektowania strony WWW tak, aby jej wygląd i układ dostosowywał się automatycznie do rozmiaru okna urządzenia, na którym jest wyświetlany.

Zaprojektowanie strony w RWD wymaga użycia HTML i CSS.

Układ strony przy wyświetleniu na danym urządzeniu powinien mieścić się w szerokości wyświetlacza. Użytkownik przewija stronę tylko w pionie, a nie w poziomie. Nie ma konieczności pomniejszania w celu obejrzenia całości. Dostępne jest powiększanie. Ogólne reguły przy projektowaniu RWD:

- Nie stosuj dużych elementów o ustalonej szerokości.
- Nie dostosowuj strony do jednej szerokości, przy której będzie dobrze ona wyglądać.
- Używaj reguł @media dla nadania różnych stylów zależnie od wielkości urządzenia.
- Używaj rozmiarów względnych (np. 100%) i uważaj na pozycjonowanie absolutne (może ono spowodować wypadnięcie elementu poza widoczny obszar).



Ważna ciekawostka

Algorytm wyszukiwania Google faworyzuje strony zgodne z RWD. Strony nieresponsywne pozycjonują się znacznie gorzej dla użytkowników wyszukujących informacji na urządzeniach mobilnych.

przesuń, powiększ/pomniejsz, pokaż/ukryj

responsywność

Krok 1 – **viewport**

Viewport to obszar strony internetowej, który widoczny jest na urządzeniu użytkownika. Rozmiar viewport zależy zatem od rozmiaru wyświetlacza.

W celu kontroli viewport należy dodać do kodu HTML w części <head> następujący element <meta>

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

width=device-width – wskazuje, że szerokość strony powinna naśladować szerokość wyświetlacza,

initial-scale=1.0 – ustawia powiększenie strony przy pierwszym jej załadowaniu.

Krok 2 – **box-sizing**

Cecha box-sizing umożliwia dołączenie do zadanych wymiarów width i height odstępu (padding) i obramowania (border). Ułatwia to nadawanie rozmiarów poszczególnym elementom oraz obszarom układu strony.

```
* { box-sizing: border-box; }
```

responsywność

Krok 3 – **width**

Cecha `width` pozwala ustawić względne rozmiary sąsiadujących kolumn/obszarów w układzie strony. Np.:

```
div#menu {width: 25%; float: left;}  
div#tresc {width: 75%; float: left;}
```

W kodzie HTML należy wyróżnić bloki `div` będące różnymi obszarami strony.

Krok 4 - **@media**

Reguła `@media` pozwala uwzględnić dany zestaw deklaracji CSS tylko jeśli spełniony jest zadany warunek. Np. w poniższym przypadku tło staje się niebieskie, jeśli strona jest wyświetlana na ekranie (`screen`) o szerokości mniejszej niż 500 pikseli.

```
@media only screen and (max-width: 500px) { body { background-color: lightblue; } }
```

Inny przykład: przy małej szerokości okna, poniżej zadanego punktu (tzw. **breakpoint**, tu 500px), strona będzie wyświetlana jako jedno kolumnowa z pionowym menu.

```
@media screen and (max-width: 500px)  
    { div#menu {width: 100%; float: none;} div#tresc {width: 100%; float: none;} }
```

responsywność

@media

Reguła @media pozwala dostosować styl do możliwości urządzenia wyświetlającego. Reguła pojawiła się w CSS2 (nieco inne zastosowanie, patrz opis <link> w prezentacji poświęconej <head>). W CSS3 nazywana jest „media queries”. Działa na zasadzie instrukcji warunkowej. Pozwala na sprawdzenie następujących własności:

- szerokość i wysokość viewport/wyświetlacza
- orientacja ekranu (pionowa/pozioma)
- rozdzielczość
- ...

Składnia:

```
@media not|only mediatype and (właściwości urządzenia) { deklaracje CSS; }
```

Szczegółowa lista właściwości dostępna jest tu: www.w3schools.com/cssref/css3_pr_mediaquery.asp

Przykładowe to:

- max-width – maksymalna szerokość obszaru wyświetlania
- orientation – orientacja urządzenia
- aspect-ratio – iloraz wysokości i szerokości viewport

CSS – układ strony

Wracamy do przykładowego układu dwukolumnowego. Do reguł CSS dodajemy responsywność.

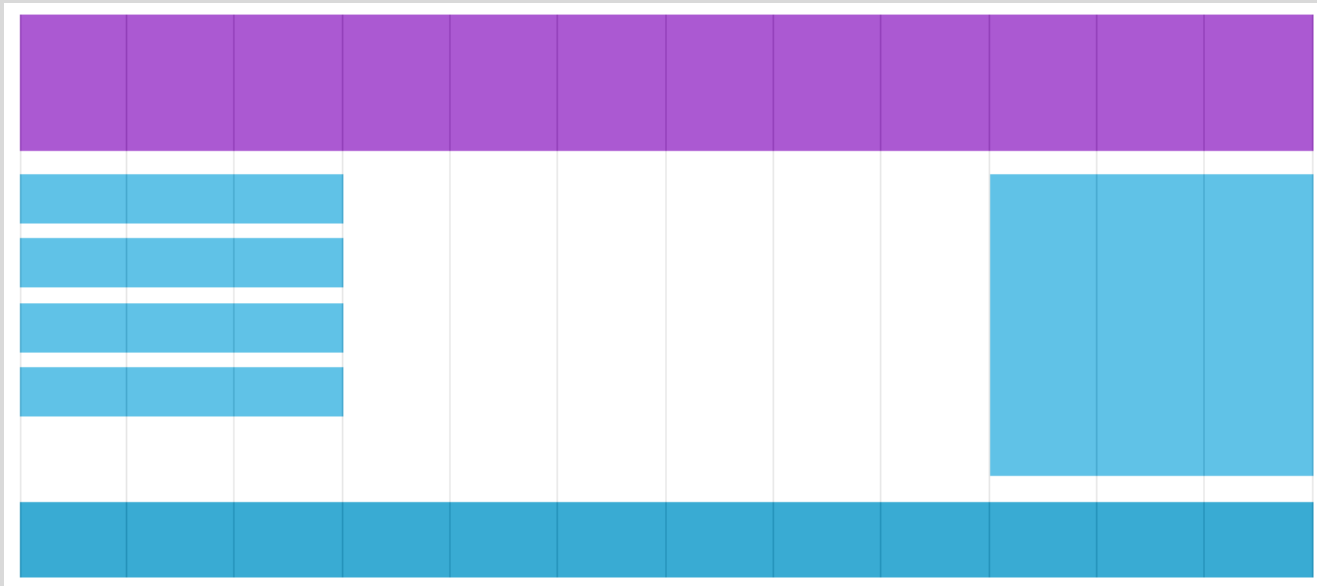
```
/* kolumny 1 i 2 umieszczane jedna pod drugą, jeśli strona wyświetlana jest na szerokości  
mniejszej niż 600 */
```

```
@media only screen and (max-width:600px) {  
  div[id*="column"] { width: 100%; }  
}
```

CSS – układ strony

Kilka uwag:

- Projektując układ strony warto najpierw zrobić układ mobilny, a później układ desktopowy.
- Często projektowanie układu strony opierane jest na 12 kolumnach, których sumaryczna szerokość wynosi 100%.
- Istnieje wiele różnych rozmiarów ekranów. Dla prostoty można przyjąć następujący zestaw breakpoints:
 - @media only screen and (max-width: 600px) {...} - do 600 px, małe smartfony
 - @media only screen and (min-width: 600px) {...} - ponad 600 px, większe smartfony, tablety w pionie
 - @media only screen and (min-width: 768px) {...} - ponad 768 px, tablety
 - @media only screen and (min-width: 992px) {...} - ponad 992 px, laptopy i kom. biurkowe
 - @media only screen and (min-width: 1200px) {...} - ponad 1200 px, duże laptopy i kom. biurkowe



CSS – układ strony

Kilka uwag:

- W RWD również rozmiar obrazów powinien być skalowany.

```
img { width: 100%; height: auto; }
```

obraz skalowany poniżej i powyżej rozmiaru oryginalnego

```
img { max-width: 100%; height: auto; }
```

obraz skalowany tylko poniżej rozmiaru oryginalnego

- Podobnie można skalować wyświetlanie plików video.
- Oprócz użycia float do uzyskania responsywności, można też użyć **flexbox**. Aby użyć modelu flexbox należy zdefiniować kontener zawierający elementy, które mają być flex.

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
</div>
```

Następnie należy ustawić dla tego kontenera cechę display na wartość flex:

```
.flex-container {display: flex;}
```

Do dyspozycji mamy zestaw cech, które można przypisać do kontenera i uzyskać odpowiednie wyświetlanie zawartych w nim elementów, np.

```
.flex-container { display: flex; flex-direction: column; }
```

wyświetlenie w kolumnie

```
.flex-container { display: flex; flex-direction: row; }
```

wyświetlenie w rzędzie

```
.flex-container { display: flex; flex-wrap: wrap; }
```

wyświetlenie z zawijaniem zależnym od szer. ekranu

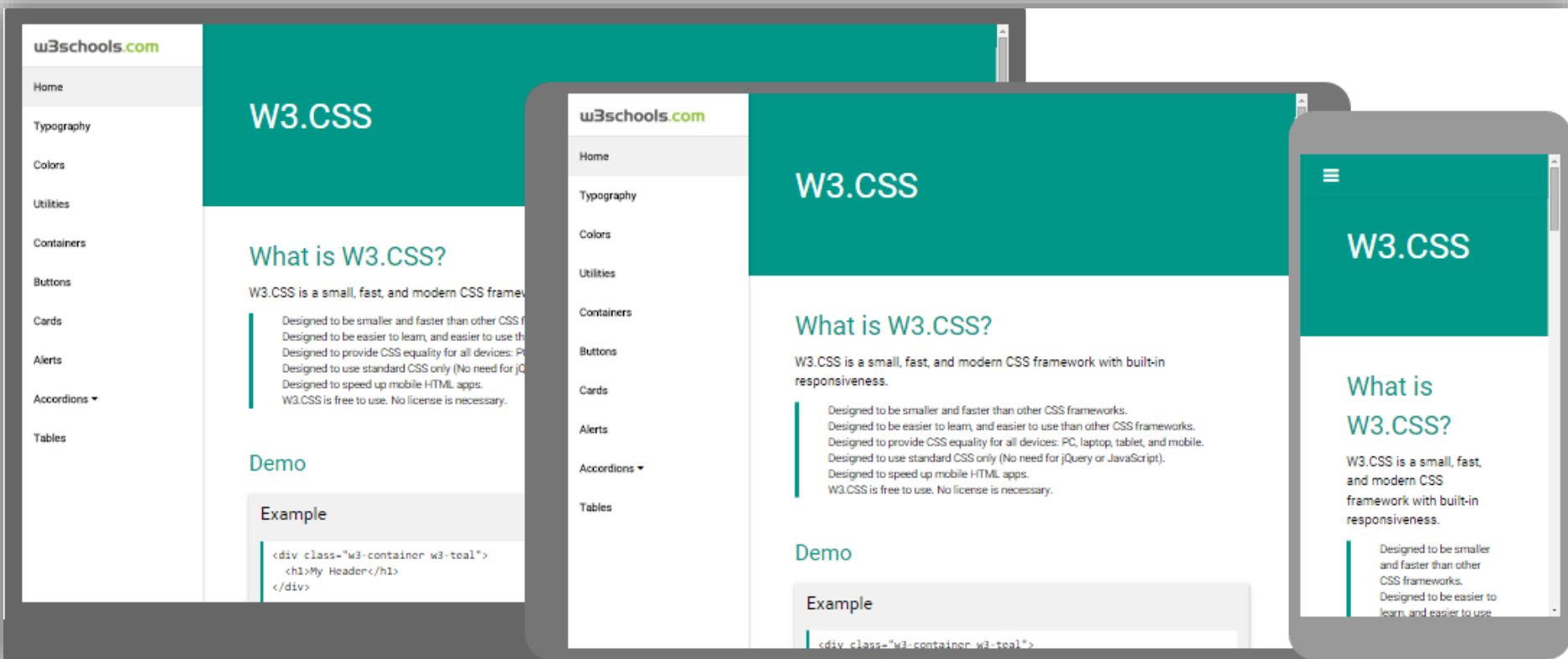
biblioteki CSS

Biblioteki CSS (CSS Frameworks) oferują wsparcie dla tworzenia nowoczesnych stron www z uwzględnieniem responsywności. Dostępnych jest wiele takich bibliotek, np.:

- **W3.CSS** (www.w3schools.com/w3css/default.asp) – lekka, łatwa w użyciu, oparta tylko na CSS
- **Bootstrap** (www.w3schools.com/bootstrap/bootstrap_ver.asp) – najpopularniejsza biblioteka

Obie biblioteki są darmowe.

Dostępne są też darmowe szablony stron www zbudowane na W3.CSS.



co dalej z gotową stroną WWW?

Zdobycie miejsca na serwerze WWW (hosting)

- zakup konta od godnej zaufania firmy zajmującej się hostingiem WWW
- dla użytku niekomercyjnego można zdobyć darmowe konto na serwerze (są ograniczenia, reklamy); przykładowe darmowe serwery to Interia, Republika, GitHub Pages

Przerzucenie plików składających się na naszą stronę WWW na konto na serwerze

- wykorzystamy do tego program klient (S)FTP zewnętrzny lub wewnętrzny od dostawcy hostingu (dostępny przez panel administracyjny)

Zakup/rejestracja domeny

- domena powinna być łatwa do zapamiętania i kojarzyć się z zawartością strony
- przypisanie domeny do serwera, na którym jest katalog ze naszą stroną WWW (domena wskazuje na ten katalog)
- domenę można kupić razem z hostingiem
- lista firm sprzedające domeny (partnerów NASK) dostępna jest pod tym adresem: www.dns.pl/lista_rejestratorow

Rejestracja strony w katalogu www / wyszukiwarce

- strona musi zostać zindeksowana przez roboty sieciowe, aby była widoczna w wynikach wyszukiwania np. Google lub Bing
- przykładowy katalog WWW, na którym można zarejestrować stronę: www.dodaj-strone.com.pl
- jeśli mamy konto Google, to możemy naszą stronę sami zarejestrować w wyszukiwarce Google używając do tego narzędzi dla webmasterów

Pozycjonowanie strony

- jeśli chcemy, aby nasza strona znalazła się wyżej w wynikach wyszukiwarek musimy zadbać o kilka prostych, ale skutecznych rzeczy: treść (dużo solidnej treści), opis i słowa kluczowe w <meta>, responsywność strony, linki do niej na innych stronach (najlepiej na takich, które są wysoko pozycjonowane w przeglądarkach)
- więcej na temat pozycjonowania można poczytać na tej stronie: brandbay.pl/blog/pozycjonowanie-strony-samemu

Później...

- ...modyfikacja / odświeżanie / ulepszanie treści i plików tworzących stronę
- odnowienie/utrzymanie domeny (zwykle co roku należy zapłacić za domenę za cały rok z góry)