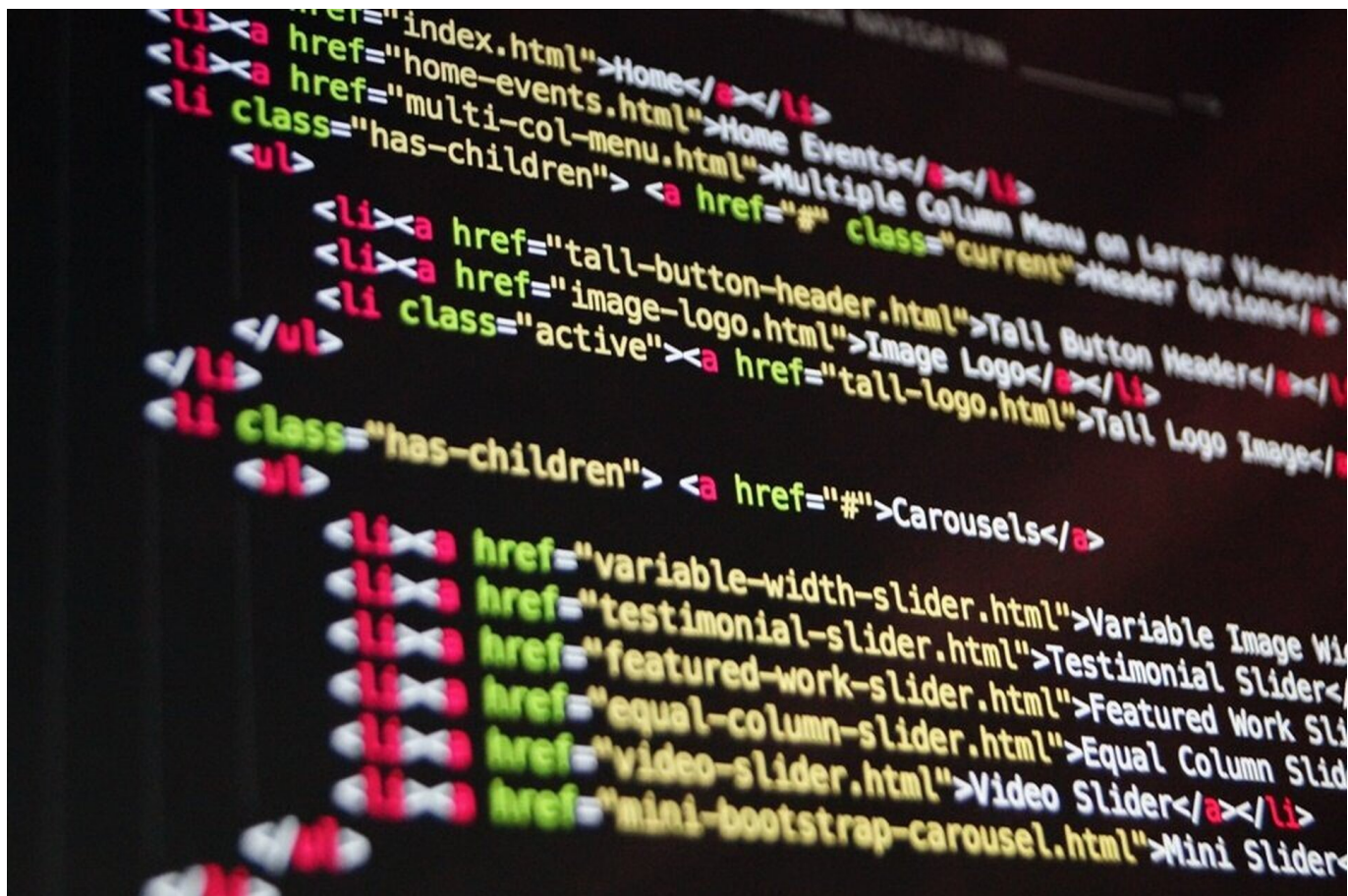
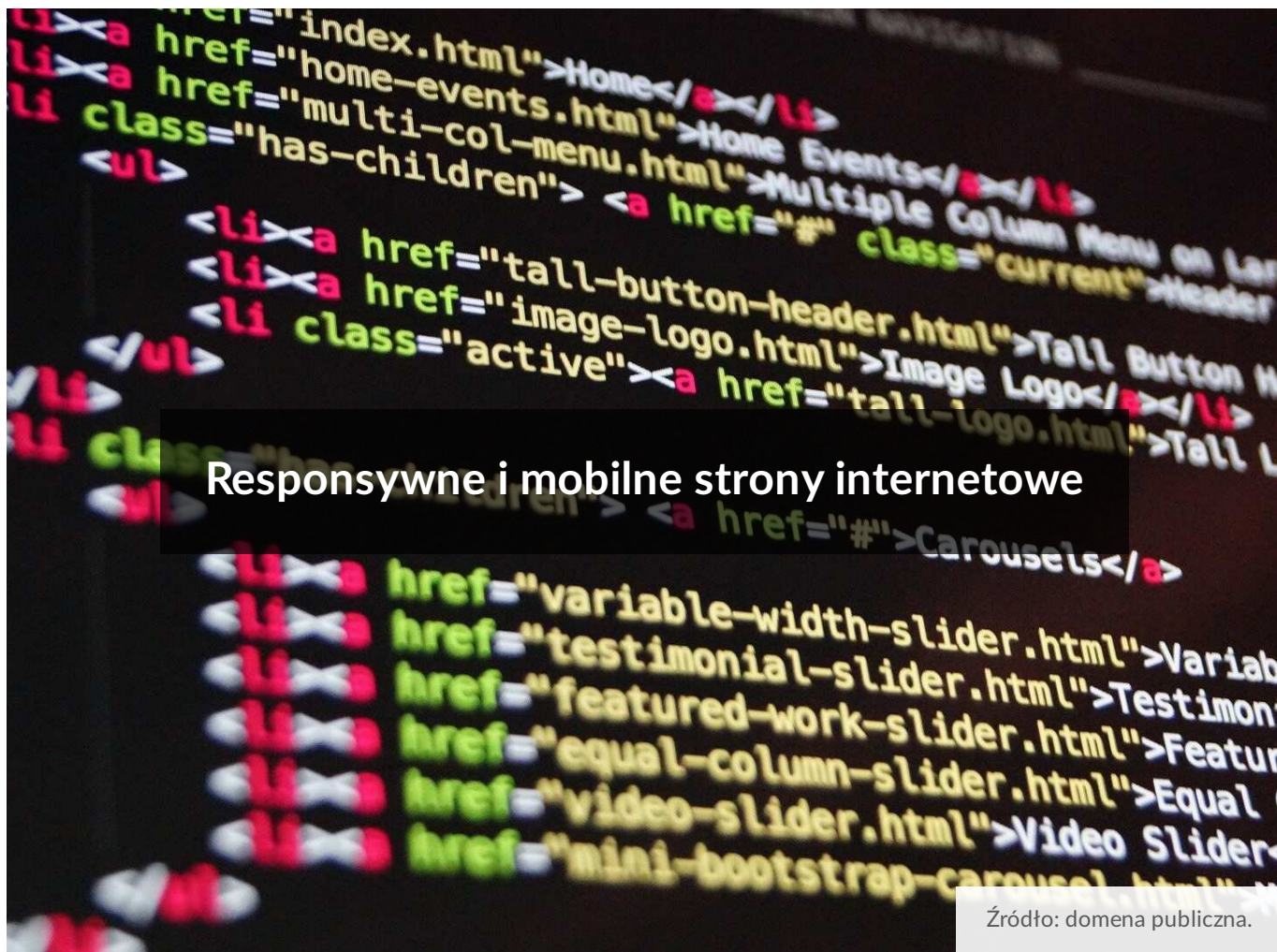




Responsywne i mobilne strony internetowe

- Wprowadzenie
- Przeczytaj
- Infografika
- Mapa myśli
- Dla nauczyciela



Nowoczesne strony internetowe wyświetlane na telefonach i komputerach osobistych wyglądają nieco inaczej. Jest to świadome rozwiązanie twórców, które wiąże się z responsywnością. Przeglądając strony internetowe za pomocą telefonu, robimy to inaczej, niż używając komputera. Różnice pojawiają się już na poziomie wielkości ekranu, na którym wyświetlamy treści. Jak zatem stworzyć stronę, by zarówno użytkownicy urządzeń mobilnych, jak i komputerów osobistych, mogli korzystać z niej bez problemów?

Przypomnij sobie najważniejsze informacje dotyczące stylów CSS, które znajdują się w e-materiale: [Język HTML i podstawy CSS](#).

Pozostałe e-materiały z serii:

- [Dobre praktyki związane z tworzeniem stron internetowych](#),
- [Kreatory stron internetowych](#),

- [Etapy projektowania strony internetowej](#),
- [Fonty i ikony w projekcie strony internetowej](#),
- [Formularze i menu w projekcie strony internetowej](#),
- [Graficzna prezentacja danych na stronie internetowej](#),
- [Nadawanie stylów elementom strony internetowej](#),
- [Tworzenie galerii zdjęć na stronie internetowej](#),
- [Elementy interaktywne w projekcie strony internetowej](#).

Twoje cele

- Wyjaśnisz, na czym polega responsywność.
- Scharakteryzujesz, w jaki sposób stworzyć stronę responsywną.
- Porównasz responsywne i nieresponsywne strony internetowe.

Przeczytaj

[Responsywność](#) (z ang. *Responsive Web Design*) to technika projektowania stron internetowych, w taki sposób, aby ich układ i wygląd dostosowywały się do rozmiaru ekranu, na którym są wyświetlane.

Wcześniej tworzone były osobne strony na każdy rodzaj ekranu, współcześnie pisze się jedną, która będzie czytelna i zapewni wygodę użytkownikom wszystkich urządzeń.

Elementy składające się na responsywność strony internetowej

Jest wiele sposobów, które pozwalają na osiągnięcie responsywności, w tym e-materiale przeanalizujemy kilka z nich.

Na początek pobierz przygotowany projekt zawierający pliki HTML i CSS Posłużą one do testowania przedstawionych informacji.

Plik o rozmiarze 536.00 B w języku polskim

Zastosowanie wszystkich omawianych elementów możesz zobaczyć też w większym projekcie w e-materiale [Tworzenie galerii zdjęć na stronie internetowej](#).

Viewport

Pojęcie to określa obszar ekranu, na którym użytkownik wyświetla stronę internetową. W celu odwołania się do tego elementu w dokumencie HTML umieszczamy zapis w sekcji <head>:

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

Zobaczmy, co oznaczają poszczególne elementy zdefiniowane w metatagu viewport:

- `width=device-width` – ustawia szerokość strony zgodnie z szerokością ekranu (która będzie się różnić w zależności od urządzenia),
- `initial-scale=1.0` – ustala początkowy poziom powiększenia, gdy strona jest ładowana po raz pierwszy przez przeglądarkę.

W naszym pliku HTML wygląda to następująco:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Responsywna strona internetowa</title>
6     <link rel="stylesheet" href="css/style.css">
7     <meta name="viewport" content="width=device-width, initial-sc
8 </head>
9 <body>
10     <h1>hello world!</h1>
11 </body>
12 </html>
```

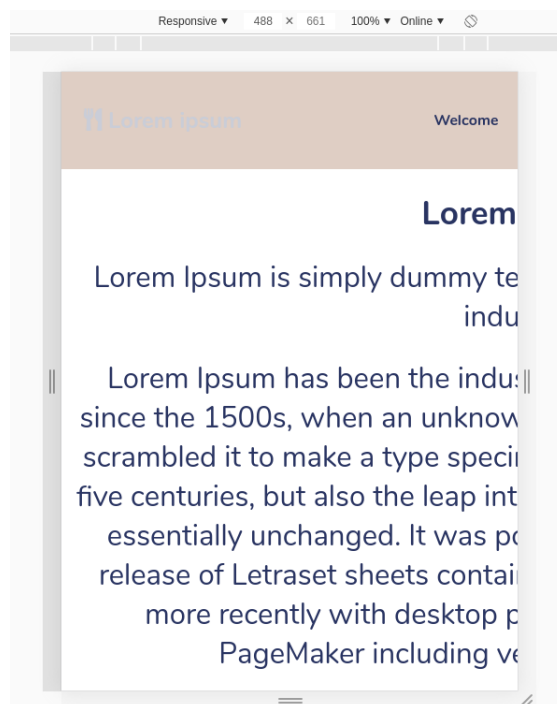
Jeśli nie zostanie zdefiniowany metatag `viewport`, strona na małych ekranach telefonów komórkowych zostanie wyświetlona w rozdzielczości zastępczej (między 800 px a 1024 px). W takim wypadku użytkownik, chcąc zapoznać się z jej zawartością, będzie musiał powiększać stronę albo przesuwając ją palcem.

Przykład 1

Porównajmy, jak wyświetli się przykładowa strona internetowa, gdy w dokumencie HTML został zdefiniowany metatag `viewport` oraz w momencie, kiedy tego tagu zabrakło (najlepiej widoczne będzie to już w gotowym projekcie strony internetowej).



Wersja, w której zastosowano zapis: `<meta name="viewport" content="width=device-width, initial-scale=1.0">` .



Wersja bez metatagu viewport.

Jednostki relatywne (rem, em, %, vw, vh) zamiast px

Jednostka em jest względna i jej wartość zależy od rozmiaru czcionki użytej w dokumencie. Jeżeli na nadrzędnym elemencie HTML ustawimy `font-size: 16px`,

to dla każdego elementu wewnątrz wartość 1em również będzie wynosić 16px. Przeanalizujmy implementację tego rozwiązania.

W pliku HTML tworzymy div o klasie hello. A w środku tego elementu paragraf zawierający tekst hello everyone :).

```
1 <body>
2   <h1>hello world!</h1>
3   <div class="hello">
4     <p>hello everyone :)</p>
5   </div>
6 </body>
```

Po wprowadzeniu kodu, zapisaniu zmian i odświeżeniu strony, uzyskujemy następujący widok.

hello world!

hello everyone :)

Sprawdźmy teraz zastosowanie jednostki em. W pliku CSS zdefiniujemy wielkość fontu dla div o klasie hello oraz paragrafu, który znajduje się w środku.

```
1 .hello {
2   font-size: 24px;
3 }
4
5 p {
6   font-size: 2em;
7 }
```

Zobaczmy efekt:

hello world!

hello everyone :)

Widok strony internetowej

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Jednostka `rem` działa niemalże identycznie jak `em`, z tą różnicą, że wartość nie jest dziedziczona od elementu nadrzędnego (tzw. rodzica). Jej wartość jest brana z najwyższego elementu w drzewie dokumentu HTML. Najczęściej jest to tag `<html>`.

Zmieńmy zatem w pliku CSS wielkość fontu w paragrafie na `2rem` i zobaczmy efekt.

```
1 .hello {  
2     font-size: 24px;  
3 }  
4  
5 p {  
6     font-size: 2rem;  
7 }
```

hello world!

hello everyone :)

Widok strony internetowej.

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Jednostki `viewport` są przeliczane w zależności od aktualnej szerokości ekranu (przy przykładowej szerokości 1200 px wartość 1 vw jest równa 1% razy 1200 px, czyli będzie

to 12 px).

1vw = 1% szerokości viewportu 1vh = 1% wysokości viewportu

Ostatnia z podstawowych jednostek relatywnych to %. Ustawia ona rozmiar względem najbliższego rodzica.

Polecenie 1

Podobnie, jak w przypadku testowania jednostek em i rem, ustaw paragrafowi:

- `font-size = 50%;`
- `font-size = 50vw;`

Zobacz, jak zmienia się wielkość fontu i pomyśl, z czego wynikają te zmiany.

Ważne!

Jeśli stosujesz jednostki relatywne, zwróć uwagę, czy ostatecznie wyliczana wartość jest wyrażona liczbą całkowitą.

Przykładowo, jeśli ustawiasz na body `font-size: 16px`, to stosując jednostki relatywne typu rem, staraj się używać wartości takich jak `font-size: .5rem / .75rem / .875rem / 1rem / 1.125rem`; itd, aby przeliczając na piksele, uzyskać liczbę całkowitą 8px / 12px / 14px / 16px / 18px.

Stosowanie reguły @media

Dzięki zastosowaniu reguły @media możemy m.in. wskazywać, jaka część kodu będzie wyświetlona przez urządzenie. Regułę tę definiujemy w pliku CSS, a najpopularniejszą właściwością, do której się odnosi, jest szerokość urządzenia, na którym strona jest wyświetlana.

Dokładne działanie przetestujemy, określając, że na ekranie poniżej 567 px nie będzie wyświetlał się div o klasie hello.

```
1 .hello {
2     font-size: 24px;
3 }
4
5 p {
6     font-size: 2em;
7 }
8
9 @media (max-width:567px) {
10     .hello {
11         display: none;
12     }
13 }
```

W celu określenia szerokości możemy też wskazać wartość `min-width`, czyli minimalną liczbę pikseli. Dobrą praktyką jest jednak trzymanie się w projekcie jednego z rozwiązań.

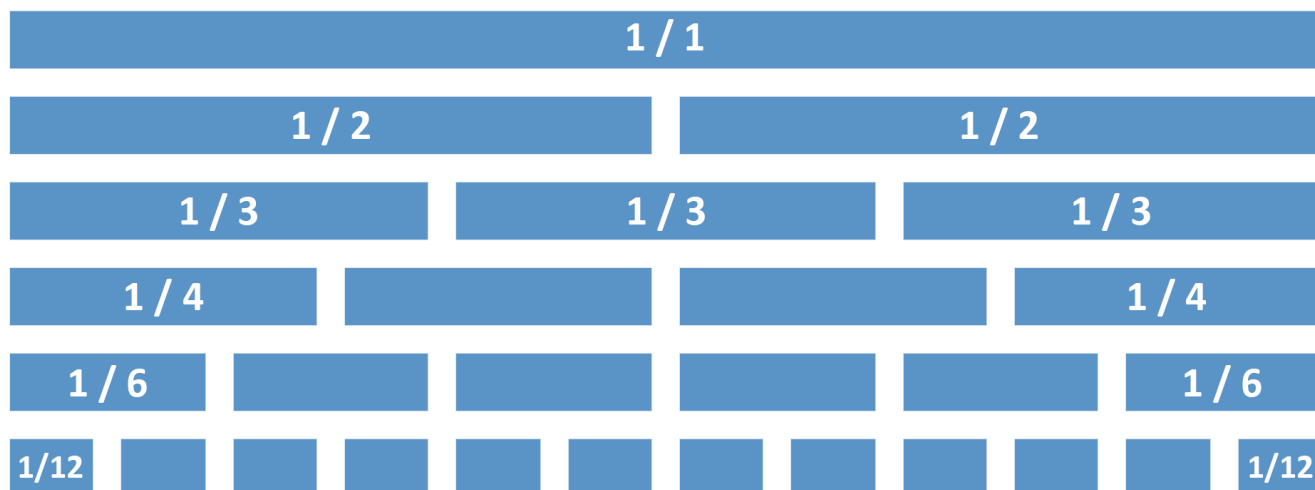
Polecenie 2

W pliku CSS wskaż kolor niebieski dla elementu `body`. Następnie użyj reguły `@media`, która będzie zmieniać ten kolor w zależności od szerokości ekranu zgodnie z poniższą informacją:

- dla ekranów poniżej 567 px tło będzie miało kolor żółty,
- dla ekranów powyżej 992 px tło będzie miało kolor czerwony.

Zastosowanie grid jako siatki układającej stronę w kolumnach i rzędach

Wykorzystanie siatki układającej całą treść na stronie internetowej przyspiesza przygotowanie strony i dostosowanie jej do wyświetlania na różnej wielkości ekranach. Można sformułować ją na różne sposoby, choć jedną z najpopularniejszych praktyk jest podział strony oparty na 12 kolumnach. Jest to praktyczne rozwiązanie, wynikające z faktu, że 12 jest nie tylko liczbą parzystą, ale dzieli się również przez 1, 2, 3, 4, 6 i 12.



Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Rozwiązanie to zapewnia elastyczność w projektowaniu układu strony.

Zasady samodzielnego przygotowania siatki dotyczą przede wszystkim ustalenia szerokości kolumn. Jeśli całość ma 100%, to jedna kolumna będzie miała 100/12, czyli 8,33%.

Możemy także podobne rozwiązanie znaleźć, sprawdzając dokumentację biblioteki Bootstrap. Jest to popularny framework CSS, stosowany przy projektowaniu stron. Możemy go umieścić w swoim projekcie albo umieścić odpowiednie źródła w tagu `<head>`. Dzięki temu rozwiązaniu nie musimy martwić się o definiowanie stylów na stronie. Wystarczy odpowiednie przygotowanie pliku HTML i korzystanie z klas, które są przygotowane przez ten framework.

```
1 <!-- CSS -->
2 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap
3
4 <!-- jQuery and JS bundle w/ Popper.js -->
5 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" in
6 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js
```

Warto zaznaczyć, że w programowaniu stosuje się także reguły dotyczące określania wielkości ekranów. Możesz z nich skorzystać lub zmodyfikować je i dostosować do

swoich potrzeb.

extra small	small	medium	large	extra large
mnijšie niż 576 px	większe równe 576 px	większe równe 768 px	większe równe 992 px	większe równe 1220 px
.col-xs	.col-sm	.col-md	.col-lg	.col-xl

Przeanalizujmy nazwę klas przedstawionych w ostatnim wierszu tabeli. Zapis `col` wskazuje, że jest to kolumna, natomiast `xs` wskazuje, że omawiana klasa będzie odczytywana przez przeglądarkę w momencie wyświetlenia strony na małym ekranie. Kolejne określenia `sm`, `md`, `lg`, `xl` odnoszą się do odpowiednio większych ekranów.

Warto szukać inspiracji w takich źródłach jak Bootstrap. Inną biblioteką, z której można korzystać, jest Semantic UI. Stosując ją, możesz ostylewać tekst, a także umieścić ikony w swoim projekcie.

Ciekawostka

Jednym z rozwiązań, które pozwoli na zastosowanie Semantic UI, jest skorzystanie ze źródła open-source CDN (z ang. *Content Delivery Network*). Aby pominąć proces instalacji, możesz w wyszukiwarce wpisać `semantic ui cdn`. Otrzymasz wtedy możliwość dołączenia tej biblioteki poprzez wskazanie w `<head>` źródła do wybranej wersji.

Polecenie 3

Korzystając z dokumentacji Bootstrap i klas jego biblioteki, stwórz grid zgodne ze wzorem wskazanym poniżej.

col-sm	col-sm
col-sm	col-sm

Dla zainteresowanych

Korzystając z klas Bootstrapa, możesz także przygotowywać siatki, które zmieniają szerokość kolumn w zależności od tego, na jakim ekranie jest

wyświetlana strona. Przykładowo zdjęcie, które wyświetla się na desktopie na połowie szerokości ekranu, na urządzeniu poniżej 768 px będzie zajmowało całą szerokość. To tylko jedno z możliwych zastosowań, o których można przeczytać w dokumentacji w części dotyczącej layoutu

Polecenie 4

W oparciu o dokumentację Bootstrapa przygotuj poniższy schemat i zaobserwuj, jak zmienia się szerokość kolumn w zależności od szerokości ekranu.

.col-md-8		.col-6 .col-md-4	
.col-6 .col-md-4	.col-6 .col-md-4		.col-6 .col-md-4
.col-6	.col-6		

Słownik

jednostki relatywne

jednostki, które są uzależnione od rozmiaru innego elementu, np. wielkości czcionki w aktualnym elemencie lub wielkości okna przeglądarki

kreator stron WWW

narzędzie do tworzenia stron internetowych; znajdziesz tam gotowe szablony

responsywność

automatyczne dostosowanie rozmiaru strony do urządzeń, na których będzie wyświetlana; responsywna strona to taka, która poprawnie wyświetla się na wszystkich monitorach i wyświetlaczach

strona mobilna

strona dedykowana tylko i wyłącznie urządzeniom mobilnym; posiada m.in. inny adres niż odpowiadająca jej strona desktopowa

Infografika

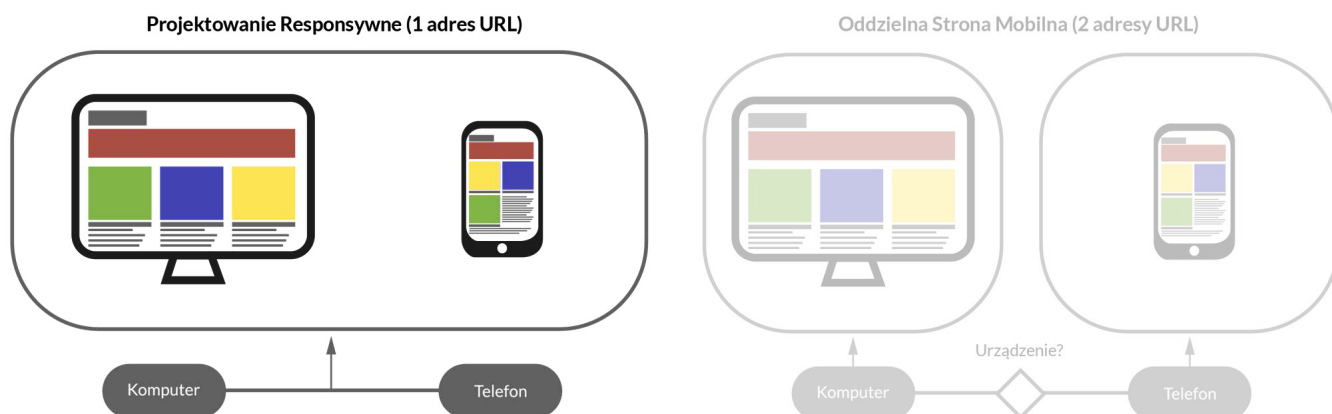
Strona responsywna a strona mobilna

W momencie pojawienia się potrzeby dostosowywania stron do wyświetlania na urządzeniach mobilnych pojawiły się dwa podejścia (a jednocześnie pojęcia): strona responsywna i strona mobilna. Często bywają one zestawiane jako synonimy. Jednak znaczenie ich jest odmienne.

Polecenie 1

Przeanalizuj infografikę, a następnie wykonaj ćwiczenie, aby zapoznać się z różnicami pomiędzy pojęciami: strona responsywna i strona mobilna.

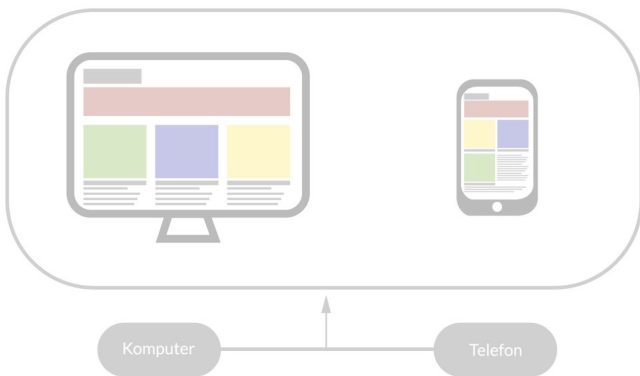
Strona responsywna



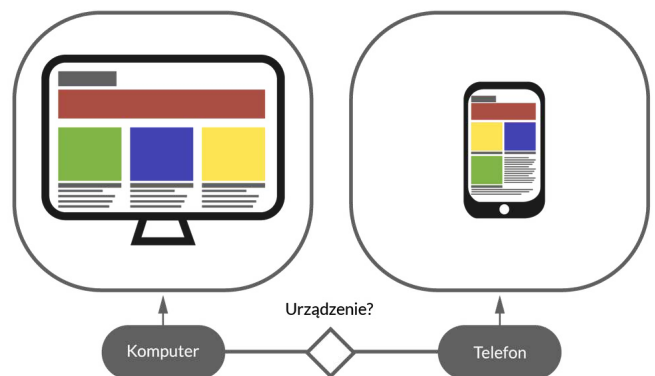
Dopasowuje się do różnych wielkości i rozdzielczości ekranów dzięki zastosowaniu jednego rozwiązania. Jest uniwersalna, ta sama strona działa na desktopie, jak i na każdym innym urządzeniu mobilnym. Warto zainwestować w stronę responsywną, jeśli dopiero planujesz stworzyć swój serwis. Jedną z zalet takiego rozwiązania jest fakt, że nie musisz aktualizować dwóch witryn. Inną natomiast, że wyszukiwarka Google preferuje strony responsywne, więc lepiej je pozycjonuje.

Strona mobilna

Projektowanie Responsywne (1 adres URL)



Oddzielna Strona Mobilna (2 adresy URL)



Rozwiązanie opierające się na umieszczeniu wersji mobilnej strony pod osobnym adresem. Od wersji desktopowej różni się kodem HTML. Strona mobilna jest popularnym rozwiązaniem w sytuacji, kiedy nie przeprowadza się częstych zmian na stronie internetowej. Może być dobrym wyjściem, jeśli twoja strona już istnieje.

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Ćwiczenie 1

Poszukaj informacji dotyczących charakterystyki statystyk stron responsywnych oraz mobilnych. Następnie przyporządkuj elementy w odpowiednie miejsca.

Wybrane cechy	Strona responsywna	Strona mobilna
wygląd strony		
adres URL		
treść		
CMS		
prędkość strony		
koszt i czas realizacji		

niski, szybki

inny adres URL

wspólny z wersją desktopową

wolniejsza

odmienna od wersji desktopowej

jednakowa z desktopową

taki sam jak desktopowej

szybsza

nowa wersja strony

wyższy, wolniejszy

oddzielny od wersji desktopowej

jeden adres URL

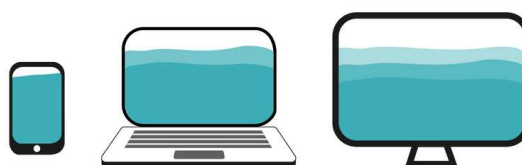
Mobile First



> >

Responsywne Dopasowywanie

próba dopasowania zawartości aktualnej strony desktopowej do strony mobilnej



> >

Projektowanie responsywne w pierwszej kolejności dla urządzeń mobilnych

wykorzystywanie przestrzeni ekranowej do stopniowego zwiększania komfortu i zawartości

To zalecane podejście tworzenia stron internetowych, w którym zaczyna się definiowanie elementów od najprostszych do bardziej złożonych. Od razu rozwiązuje ograniczenia urządzeń mobilnych i wykorzystuje dostępną przestrzeń. To założenie pomaga też w określeniu, co jest najważniejsze na stronie.

Mapa myśli

Przyjrzyj się mapie myśli. Przedstawia zbiór najlepszych praktyk stosowanych w przygotowaniu responsywnych stron. Przeanalizuj wskazane w niej elementy i uzupełnij o kolejne.



Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Dla nauczyciela

Autorka: Paulina Wierzbińska

Przedmiot: Informatyka

Temat: Responsywne i mobilne strony internetowe

Grupa docelowa:

III etap edukacyjny, liceum ogólnokształcące, technikum

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

5) przedstawia sposoby reprezentowania w komputerze znaków, liczb, wartości logicznych, obrazów, dźwięków, animacji;

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:

f) tworzy stronę internetową zgodnie ze standardami, wzbogaconą tabelami, listami, elementami dynamicznymi, posługuje się arkuszem stylów, korzysta z oprogramowania i serwisów przeznaczonych do tworzenia stron; potrafi opublikować własną stronę w internecie;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Wyjaśnisz, na czym polega responsywność.
- Scharakteryzujesz, w jaki sposób stworzyć stronę responsywną.
- Porównasz responsywne i nierespowsywne strony internetowe.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda.

Przebieg lekcji

Przed lekcją:

1. Uczniowie, korzystając z innych e-materiałów, przypominają sobie najważniejsze informacje dot. projektowania witryn WWW.
2. Uczniowie przygotowują przykłady mobilnych oraz responsywnych stron WWW. Oceniają korzystanie z jednych oraz drugich – sami ustalają kryteria oceny.

Faza wstępna:

1. Uczniowie dyskutują na temat tego, jak oceniają wygodę przeglądania stron WWW na telefonach.
2. Chętna lub wybrana osoba przypomina najważniejsze informacje dotyczące stylów CSS.

Faza realizacyjna:

1. Nauczyciel prosi chętne lub wybrane osoby o przedstawienie kryteriów oceny, wedle których ewaluowały swoje doświadczenie korzystania ze stron mobilnych

oraz responsywnych. Prosi klasę o zabranie głosu w dyskusji, które rozwiązanie jest wygodniejsze i dlaczego.

2. Uczniowie zapoznają się z treścią e-materiału. Nauczyciel w razie potrzeby wyjaśnia niezrozumiałe treści.
3. Uczniowie w parach próbują powtórzyć przedstawione w e-materiale działania.
4. Uczniowie zapoznają się z infografiką i wykonują ćwiczenie.
5. Uczniowie zapoznają się z mapą myśli i wykonują ćwiczenie.

Faza podsumowująca:

1. Uczniowie porównują swoje rezultaty.

Praca domowa:

1. Wykonaj wszystkie przedstawione w e-materiale kroki, ale już dla twojej właściwej witryny.

Wskazówki metodyczne:

- Nauczyciel może zdecydować, czy uczniowie realizują projekt indywidualnie czy w grupach.