CodersLab Coders Lab

CSS

17.12.2020 Kursy > WRO\_FER\_S\_24 > Nauka > SassiRWD > Dzień 5 - Warsztat > Rozwiązanie > Nawigacja Moje Kursy Career Lab Front End sygnatura: WAR\_CFE\_D\_06 Nawigacja JavaScript Developer: React sygnatura: WRO\_FER\_S\_24 0. Prework 1. Zaawansowany HTML i CSS 2. Sass i RWD Praca samodzielna - Podstawy NodeJS i npm Dzień 1 - Konfiguracja środowiska i Sass Dzień 1 - Praca domowa Dzień 2 - Sass i RWD Dzień 2 - Praca domowa Dzień 3 - RWD Dzień 3 - Praca domowa • Dzień 4 - RWD i Animacje Dzień 4 - Praca domowa Dzień 5 - Warsztat - Omówienie Rozwiązanie Egzamin próbny • Egzamin próbny - rozwiązanie 3. Praca własna 1

4. JavaScript 5. ES6 i React Podstawy 6. Praca własna 2 7. React

Scrum Lab Front End

ldź w góre Poprzedni Nastepny

U Jarosław Krynicki EN/PL U Wyloguj

Sekcja nawigacji jest pierwszą sekcją z którą się zmierzymy. Rozpocznijmy od struktury HTML. Jest to jedyna sekcja podczas tego warsztatu która ulegnie zmianom jeśli chodzi o HTML! Musimy zmienić lekko strukturę HTML ze względu na "hamburger icon" który ma się pojawić w wersji mobilnej naszej strony. Po kliknięciu menu powinno się otworzyć zaraz pod logotypem, przysłaniając sekcję otwierającą. W elemencie nav w klasie .header\_\_container wstawmy input[type="checkbox"] dzięki któremu będziemy wiedzieć czy nasze menu powinno być otwarte czy nie.

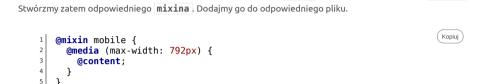
Umieścimy tam również element [label] który będzie powiązany z checkboxem (atrybut [for i ]id), w którym znajdą się 3 elementy span . Będą one imitowały ikonę hamburgera a także "X" w zależności od stanu naszego menu. Cały kod sekcji nawigacyjnej będzie wyglądał następująco:

Kopiuj | <header class="header">
| <div class="header\_\_container container">
| <a href="#" class="logo">BestShop</a> <nav>
<input type="checkbox" class="menu\_btn" id="menu\_btn">
<label for="menu\_btn" class="menu\_toggle"> </nav> 19 </header>

scss/elements/\_header.scss Stwórzmy plik **\_header.scss** w folderze **elements** . Do pliku tego przekopiujmy wszystkie style dotyczące **headera** naszej strony, tzn. wszystkie klasy rozpoczynające się od słowa **header** a także **nav** . Po przekopiowaniu, zadbajmy o poprawne zagnieżdżenie elementów zgodnie ze sztuką SCSS. Pamiętamy

również podmienieniu zmiennych. Kopiuj .header {
 height: 70px;
 box-shadow: 0 3px 6px rgba(0, 0, 0, .16); &\_\_container {
 display: flex;
 justify-content: space-between;
 align-items: center;
 height: 100%;
} .nav {
 &\_\_element {
 padding: 0 26px; &:last-child { padding-right: 0; &\_\_link {
 color: \$color-grey;
 font-weight: 700;
 font-size: 0.6875rem;
 text-transform: uppercase;
 padding: 28px 18px 22px;
 border-bottom: 4px solid transparent;
 letter-spacing: 0.1875rem; &:hover { color: \$color-black; border-color: \$color-primary; &\_\_nav {
 display: flex;

scss/settings/\_mixins.scss Nasze menu nie mieści się na ekranie mniej więcej od szerokości **792px** . Wartość tą określimy jako **mobile** .



Wróćmy do pliku scss/elements/\_header.scss. Dodajmy do naszego kodu style opisujące nowe klasy: menu\_btn i menu\_toggle . Chcemy ukryć menu\_btn ze względu na to, że jest to checkbox. Nie zależy nam na tym aby w nagłówku taki element się znajdował - chodzi nam tylko o jego specyficzne właściwości.

W klasie **header** stwórzmy następującą konstrukcję która ukryje checkboxa: Kopiuj &\_btn {
display: none;
position: absolute;
top: -9999px;
left: -9999px; opacity: 0;

Ostylujmy elementy **span** znajdujące się w klasie **menu\_toggle**. Dodatkowo ukryjemy domyślnie ten element ze względu na to, że chcemy aby był widoczny tylko w zakresie **mobile**. Dodamy więc odpowiedniego mixina do tej klasy. Nasze elementy **span** , będą prostokątami o wymiarach **20px** x **2px** z odpowiednim kolorem. Dodamy im również **transition** aby uzyskać ciekawy efekt po kliknięciu (o tym za chwilę). Dodajmy zatem do klasy **menu** następujące style:

1 & toggle {
2 display: none; span {
 width: 20px;
 height: 2px;
 display: block;
 margin better: 2 margin-bottom: 3px;
background-color: \$color-primary;
transition: .3s all ease-in-out; &:last-child { margin-bottom: 0; @include mobile {
 display: block;

Jak teraz sprawić, że menu zniknie w widoku mobilnym a także zmieni diametralnie swoją strukturę kiedy będzie otwarte? Do selektora &\_\_nav dodajmy kod który to opisze. Cały musi być zawarty w mixinie mobile bo dane style chcemy zaaplikować tylko i wyłączne w takim widoku. Nawigacja ma przykrywać sekcję otwierającą (position: absolute), być pod belką nagłówka (top: 95px) a jej elementy mają być umieszczone pod sobą (flex-direction: column). Oczywiście domyślnie pozostaje ona ukryta ( display: none ).

Same elementy nawigacji, będą miały zmieniony **padding** . Kopiuj 1 @include mobile { position: absolute; left: 0; top: 95px; flex-direction: column; background-color: white; width: 100%; padding-bottom: 20px; display: none; .nav\_\_element {
 padding: 0; .nav\_\_link {
 display: block; padding: 14px 20px;

Pozostała nam do wykonania ostatnia rzecz. Jak połączyć kliknięcie w checkboxa z otwarciem lub zamknięciem Jak wiemy element <code>label</code> razem z atrybutem <code>for</code> wskazuje na odpowiedni checkbox z wartością <code>id</code> równą tej z atrybutu 🚺 for . Kiedy klikamy w taki 🔝 label modyfikujemy stan przypisanego doo niego elementu input (zaznaczamy bądź odznaczamy). Tak więc głównym naszym założeniem będzie rozpoznanie kiedy checkbox jest zaznaczony i zareagowanie na to, zmieniając styl nawigacji na **display: flex** a także modyfikując elementy **span** tak aby ułożyły się w znak "X".

transform: rotate(45deg) translate(2px, 5px); &:nth-child(2) { opacity: 0; }
&:nth-child(3) {
 transform: rotate(-45deg) translate(2px, -5px);

Nakładamy tutaj pseudoklasę **:checked** w której szukamy rodzeństwa o klasie **header\_\_nav** (~ . header\_\_nav ) i ustawiamy odpowiedni styl. Szukamy również rodzeństwa **.menu\_\_toggle** a w nim elementów **span**. Każdemu z nich nadajemy

1. obracamy o 45 stopni i przesuwamy **2px** w poziomie i **5px** pionie 2. nastawiamy opacity: 0; aby element znikł 3. obracamy o -45 stopni i przesuwamy **2px** w poziomie i **-5px** pionie Dzięki temu, po kliknięciu w nasz hamburger icon, trzy poziome kreski płynnie zmienią się w znak "X" a pod spodem pojawi się menu.

Import styli Dodajmy odpowiedni import do pliku main.scss na jego koniec:

1 @import "elements/header";

Pełne rozwiązanie Oto pełen kod pliku scss/elements/\_header.scss: Kopiuj .header {
 height: 70px;
 box-shadow: 0 3px 6px rgba(0, 0, 0, .16); &\_container {
 display: flex;
 justify-content: space-between;
 align-items: center;
 height: 100%;
} .nav {
 &\_\_element {
 padding: 0 26px; &:last-child { padding-right: 0;
} &\_\_link {
 color: \$color-grey;
 font-weight: 700;
 font-size: 0.6875rem;
 text-transform: uppercase;
 padding: 28px 18px 22px;
 border-bottom: 4px solid transparent;
 letter-spacing: 0.1875rem; &:hover {
 color: \$color-black; border-color: \$color-primary; &\_\_nav { display: flex; @include mobile { position: absolute;
left: 0;
top: 95px;
flex-direction: column;
background-color: white;
width: 100%;
padding-bottom: 20px;
display: none; .nav\_\_element {
 padding: 0; .nav\_\_link {
 display: block;
 padding: 14px 20px; .menu {
 &\_\_btn {
 display: none;
 position: absolute;
 top: -9999px;
 left: -9999px;
 opacity: 0: opacity: 0; ~ .menu\_\_toggle {
 span {
 &:nth-child(1) {
 transform: rotate(45deg) translate(2px, 5px);
 }
} &:nth-child(2) { &:nth-child(3) {
 transform: rotate(-45deg) translate(2px, -5px); &\_\_toggle {
 display: none; span {
 width: 20px;
 height: 2px;
 display: block; margin-bottom: 3px;
background-color: \$color-primary;
transition: .3s all ease-in-out; &:last-child { margin-bottom: 0; @include mobile {
 display: block;