

ACCESSIBILITY DLA PROGRAMISTÓW FRONTEND

OLA WARZECHA

warzecha.ola@gmail.com | github.com/olawar/accessibility-workshop

SPIS TREŚCI

- **Czym jest dostępność i dlaczego jest ważna?**
- **Regulacje prawne**
- **Jak osoby niepełnosprawne korzystają z internetu?**
- **WCAG 2.0 - omówienie wymogów i dobre praktyki**
- **Audyty dostępności**
- **Dostępność a systemy CMS**
- **Materiały i przydatne linki**

CZYM JEST DOSTĘPNOŚĆ I DLACZEGO JEST WAŻNA?

Dostępność / accessibility to zapewnienie dostępu do prezentowanych treści
jak najszerszemu gronu użytkowników

UWZGLĘDNIĄ RÓŻNE OGRANICZENIA:

- niepełnosprawność zmysłową
- niepełnosprawność ruchową
- niepełnosprawność umysłową
- wiek i różnice wynikające z innej percepcji
- korzystania z innych urządzeń niż te, które chcielibyśmy widzieć jako domyślne

SKALA ZAPOTRZEBOWANIA NA DOSTĘPNE ROZWIĄZANIA:

- **10-15%** - odsetek niepełnosprawnych ogółem w Polsce
- **1,8 mln** dysfunkcje wzroku z czego **100 tys.** to niewidomi
- **900 tys.** z dużym uszczerbkiem słuchu, a **50 tys.** to niesłyszący
- **6 mln** - niedosłuch w Polsce, **80 mln** w Europie i rośnie!

Źródła: **Rynek Zdrowia**, **Raport Kongresu Kobiet**, **BPRON**

Z DOSTĘPNYCH USPRAWNIEN KORZYSTA KAŻDY Z NAS:

- Responsive Web Design
- crossbrowser compatibility
- transkrypcja w wideo
- możliwość powiększenia tekstu
- a nawet - błędy podczas ładowania strony

REGULACJE PRAWNE: POZIOM ODPOWIADAJĄCY AA Z WCAG 2.0

PODMIOTY REALIZUJĄCE ZADANIA PUBLICZNE

Podstawa prawna: Rozporządzenie w sprawie Krajowych Ram Interoperacyjności obowiązujące od czerwca 2015 - podmioty realizujące zadania publiczne

Obejmuje: jednostki administracji państwowej - ministerstwa, urzędy, policję, straż pożarną, jednostki służby zdrowia, przyjmuje się także, że NGOsy

DOSTAWCY TELEKOMUNIKACYJNI

Postawa prawna: Rozporządzenie Ministra Administracji i Cyfryzacji obowiązujące od 10 kwietnia 2016 roku

Obejmuje: duże (Orange, Play, T-Mobile, Plus) i małe firmy telekomunikacyjne (dostawcy lokalni) oferujące dostęp do internetu / telefonu dla konsumentów

JAK OSOBY NIEPEŁNOSPRAWNE KORZYSTAJĄ Z INTERNETU?

PORUSZANIE SIĘ PO STRONIE:

- "skanowanie" nagłówków - skróty czytnika
- przejście po interaktywnych elementach strony - linki, buttony, inputy i elementy z `tabindex >= 0` - **Tab** i **Shift + Tab**
- poruszanie się po wybranych elementach skrótami, np. listy, paragrafy, etc. - skróty czytnika

Więcej informacji na **WebAIM**

WYBÓR NARZĘDZI

- czytniki software'owe: **NVDA**, **JAWS**, **Window-Eyes (współpraca z urządzeniami brailowskimi)**, VoiceOver (Apple)
- programy powiększające
- linijka brajlowska
- wsparcie w smartfonach
- klawiatura i jej wariacje!



WEB CONTENT ACCESSIBILITY GUIDELINES 2.0

Standard od W3C, podzielony na 3 poziomy

- A - najbardziej podstawowy
- AA - optymalny
- AAA - najbardziej wymagający, rzadko spotykany

Uzyskanie certyfikacji na danym poziomie wymaga spełnienia wszystkich wytycznych z danego i niższych poziomów

Skrócone zestawienie poziomów

TECHNIKI POTRZEBNE DO SPEŁNIENIA TYCH WYMOGÓW

- dobre przygotowanie UX i GUI
- semantyczny HTML
- WAI-ARIA
- 'odpowiedzialny' CSS i możliwość obejścia się bez niego
- dostosowany JS

DOBRE PRZYGOTOWANIE UX I GUI

- świadome przygotowanie logicznych treści, w tym alternatyw dla czytnika
- przygotowanie projektów graficznych zgodnie z wymogami WCAG
- komunikacja z deweloperami!

UX I GUI - SPÓJNA ORGANIZACJA BLOKÓW TREŚCI W SERWISIE WWW

Nawigacja główna | wyszukiwarka

Ścieżka okruszków > informuje gdzie jesteśmy

Główna treść

Stopka

SEMANTYCZNY HTML

- korzystanie ze znaczenia tagów HTML5, np. article, section, nav, header, main, footer, ul, itp.
- poprawność ze specyfikacją
- listy zagnieżdżone, odpowiednia kolejność focusu, korzystanie z tabindex, itp.

ATRYBUTY WAI-ARIA

Web Accessibility Initiative - Accessible Rich Internet Applications - pozwalają zwiększyć dostępność dynamicznych treści i elementów interfejsu użytkownika

Definiują role, właściwości i stany elementów na stronie

WAI-ARIA: ROLE

- abstract
- widget: button, checkbox, dialog, tabpanel, checkbox
- document structure: article, document, list
- landmark - regiony nawigacyjne na stronie: banner, form, navigation, search

WAI-ARIA: WŁAŚCIWOŚCI I STANY

- widget - do elementów UI: aria-haspopup, aria-label, aria-hidden, aria-expanded
- live region - dynamiczne zmiany w treści: aria-atomic, aria-relevant, aria-live
- drag and drop
- zależności: aria-controls, aria-labelledby, aria-describedby

PRZYKŁAD ZASTOSOWANIA - MODAL / OKNO DIALOGOWE

```
<div id="modal" role="dialog" aria-labelledby="modal-title" aria-describedby="modal-content">  
  <h2 id="modal-title">To jest modal</h2>  
  <p id="modal-content">Ten modal jest dostępny</p>  
</div>
```

PRZYKŁAD ZASTOSOWANIA - TABY / AKORDEON

```
<ul id="tabs-controls" role="tablist">
  <li role="presentation">
    <a href="#content-1" id="tab-1" role="tab" aria-controls="content-1" aria-expanded=""
  </li>
  <li role="presentation">
    <a href="#content-2" id="tab-2" role="tab" aria-controls="content-2" aria-expanded=""
  </li>
</ul>

<div class="tabs-content-wrapper">
  <div id="content-1" aria-labelledby="tab-1" role="tabpanel" aria-hidden="">
    <p>To jest treść pierwszego taba</p>
  </div>
  <div id="content-2" aria-labelledby="tab-2" role="tabpanel" aria-hidden="">
    <p>To jest treść drugiego taba</p>
  </div>
</div>
```


PRZYKŁAD ZASTOSOWANIA - WIELOPOZIOMOWA NAWIGACJA

```
<nav role="navigation">
  <ul class="first-level">
    <li><a href="#" aria-haspopup="true">O firmie</a>
      <ul class="second-level">
        <li><a href="#" aria-haspopup="true">Historia</a>
          <ul class="third-level">
            <li><a href="#">Rodzinne dziedzictwo</a></li>
            <li><a href="#">Firma dziś</a></li>
          </ul>
        </li></ul>
      </li>
    <li><a href="#">Strategia</a></li>
  </ul>
  <li><a href="#">Oferta</a></li>
</ul>
</nav>
```

'ODPOWIEDZIALNY' CSS

- `display:none`, i `visibility:hidden` ukrywają tekst przed czytnikiem
- stosowane w animacjach `opacity:0` wciąż jest dostępne dla czytnika!
- strona musi być dostępna również bez CSS
- ciekawostka: **CSS2 Aural Style Sheets**, ale praktycznie bez wsparcia

DOSTOSOWANY JS

- przyjmuje się, że wszystkie czytniki obsługują JS
- niezbędne jest zakodowanie odpowiednich stanów ARIA przy zmianach elementów, np. aria-hidden, aria-expanded, aria-selected
- trzeba też pamiętać o dostępie z klawiatury oraz odpowiednim zarządzaniu focusem

PODZIAŁ WCAG 2.0

1. Percepcja - sposób przedstawienia treści
2. Funkcjonalność - używalność zastosowanych komponentów
3. Zrozumiałość - zrozumiałe treści i obsługa
4. Rzetelność - treść przygotowana tak, aby umożliwiała zastosowanie technologii wspomagających

1. PERCEPCJA

1.1 Tekst alternatywny dla informacji nietekstowych; np. alty dla obrazków

1.2 Media zależne od czasu (audio / video) - zapewnienie alternatywy jak transkrypcji, języka migowego

1.3 Możliwość adaptacji - zakodowanie elementów tak, aby było możliwe poruszanie się skrótami czytnika i zrozumiale przedstawić treści w sposób uproszczony (np. bez CSS), nieodnoszenie się do właściwości zmysłowych (np. kolorów)

1.4 Rozróżnialność

- na poziomie AA wymagany kontrast podstawowy tekst vs tło to 4,5:1 (coraz rzadziej rekomenduje się osobną wersję serwisu z podwyższonym kontrastem, choć wciąż zdarzają się **ciekawe realizacje**)
- powiększanie czcionek możliwe do 200% (rekomendowane są jednostki rem, em, %, px)
- brak tekstu w postaci grafiki bez alternatywy
- unikanie justowania tekstu do prawej

1.4 Rozróżnialność (c.d.)

Należy też unikać zakłóceń ze strony multimediiów - domyślnie wyłączone odtwarzanie wideo i dźwięku, możliwość sterowania z klawiatury

2. FUNKCJONALNOŚĆ

2.1 Dostępność z klawiatury

- domyślne klawisze: Tab, Shift + Tab, Enter, w elementach formularzy też spacja i strzałki
- główna nawigacja - dodatkowe wsparcie jak zapewnienie poruszania się strzałkami - przykład na **Play.pl**
- (powiązane z 1.3) uwzględnienie poruszania się skrótami czytelnika: H - nagłówki, L - listy, G - grafiki, F - formularze
- brak pułapek dla klawiatury

2.1 Dostępność z klawiatury (c.d.)

Ważne jest odpowiednie zarządzanie focusem - np. przeniesienie focusu na otwarty modal i 'zamknięcie' go w nim - przykład z **Bootstrapa**, albo korzystanie z tabindex do zapewnienia rekomendowanej kolejności poruszania się np. w formularzu

2.2 Zapewnienie wystarczająco dużo czasu na wykonanie akcji

2.3 Ryzyko ataku padaczki - nie należy tworzyć treści, które migają więcej niż 3 razy na sekundę; wyjątki: miganie jest niewielkie, błyski mają mały kontrast i nie zawierają dużo czerwieni.

2.4 Nawigacja po stronie

- <title> - powinien podawać o czym jest dana strona
- tylko jeden <h1> na stronie, również bezpośrednio odnoszący się do jej najważniejszej zawartości
- skiplinki (pomijanie bloków treści)- ułatwiają nawigację po stronie;
przykład: - **nawiguj klawiszem Tab**

JAK NAPISAĆ SKIPLINKI?

```
<ul class="skiplinks">
  <li><a href="#header">Przejdź do menu</a></li>
  <li><a href="#main-search">Przejdź do wyszukiwarki</a></li>
  <li><a href="#main-container">Przejdź do treści</a></li>
  <li><a href="#footer">Przejdź do stopki</a></li>
</ul>

<style>
  .skiplinks {
    list-style: none;
  }
  .skiplinks a {
    display: inline-block;
    padding: 16px;
    border: 3px solid orange;
    position: absolute;
    left: 0;
    top: -1000px;
    width: 250px;
    z-index: 100;
  }
  .skiplinks a:focus {
    top: 0;
  }
</style>
```

2.4 Nawigacja po stronie

- breadcrumbs / ścieżka okruszków - podają lokalizację w serwisie (**przykład**)
- logiczna struktura nagłówkowa bez przeskakiwania między poziomami
- widoczny focus (**przykład**)
- cel linku w kontekście - ukryty, bardziej opisowy tekst dla czytnika np. w guzikach typu "szczegóły" lub "zobacz więcej", a także jasna informacja o odnośnikach do pobrania

JAK UKRYĆ TEKST DLA CZYTNIKA?

```
<button>Szczegóły <span class="screen-reader-only">oferty z telefonem Huawei P9</span></button>

<style>
  .screen-reader-only {
    position: absolute !important;
    clip: rect(1px 1px 1px 1px);
    padding: 0 !important;
    border: 0 !important;
    height: 1px !important;
    width: 1px !important;
    overflow: hidden;
  }
</style>
```

Frameworki często oferują gotowe rozwiązania - Bootstrap "sr-only", Foundation "show-for-sr"

Nie należy ukrywać w ten sposób dużych bloków treści, ze względu na penalizację w wyszukiwarkach

3. ZROZUMIAŁOŚĆ

3.1 Zrozumiałe treści - poprawny atrybut lang dla strony i elementów, prosty język, streszczenie dla skomplikowanych diagramów

3.2 Przewidywalność - spójna konstrukcja powtarzających się komponentów (nawigacja, formularze), brak nagłych zmian kontekstu przy np. focusie, zaakcentowanie zmiany treści, gdy nie jest to oczywiste

3.3 Pomoc przy wprowadzaniu danych - poprawnie zakodowane formularze!

- stosowanie <label> do <input> lub innej formy podania instrukcji
- gdy <input> dostaje focus <label> wciąż musi być dostępny dla czytnika
- instrukcje i komunikaty błędów muszą być zrozumiałe oraz podane w zróżnicowany sposób
- customowy <select> też musi być dostępny z klawiatury i dla czytnika

Ciekawe opracowanie, punkt 2 i przykład formularza z wyraźnymi komunikatami błędów

4. RZETELNOŚĆ

Poprawny kod!

Nie wolno zapomnieć o:

- deklaracji html
- tagu <title> innym dla każdej podstrony
- <h1> - jeden na stronie, w ostateczności może być dostępny tylko dla czytnika
- logicznej strukturze nagłówków - stylowanie po klasach a nie tylko elementach
- logicznych altach obrazków - odzwieczających ich role

Uważaj też na:

- stosowanie pseudolinków np. `` zamiast `<a>` i problematyczne linki
 - puste, zagnieżdżone w sobie, do tego samego obok siebie
- reset CSS - na 99% wyłączy defaultowy focus. Zapewnij alternatywę!
- w jaki sposób linkujesz do materiałów do pobrania i czy pliki PDF są dostępne

Uważaj też na (c.d.):

- zapewnienie dostępności takim elementom UI jak: tooltipy, dropdowny / akordeony, taby, modale, slidery / karuzele, wielopoziomowe nawigacje
- informowanie o zmianach w dynamicznych elementach strony (aria-live)

AUDYTY DOSTĘPNOŚCI

NARZĘDZIA AUDYTORA

Automatyczny walidator:

- **WAVE (Web Accessibility Evaluation Tool)** - też jako wtyczka do Firefoxa i Chrome
- wtyczka do Chrome 'Accessibility Developer Tools'
- **TAWDIS**

Analiza kontrastu - **Colour Contrast Analyser**

Czytnik

Wsparcie niepełnosprawnego testera

WYNIK AUDYTU

Raport ułożony punkt po punkcie zgodnie z dokumentacją WCAG - **np. jak w opracowaniu na stronie Fundacji Widzialni**

DOSTĘPNOŚĆ A SYSTEMY CMS

Problem jest złożony i zależy od wybranego systemu, ale generalnie prace w kierunku zapewnienia dostępności trwają - więcej informacji **tutaj**

Szczególnie drażliwe kwestie: brak wsparcia dla klawiatury, strony na tabelkach, brak możliwości wprowadzenia poprawek, brak wsparcia dla WAI-ARIA

WORDPRESS

Społeczność pracuje nad rozszerzeniem dostępności Wordpressa, materiały:

- **jak pisać dostępne w WP**
- **plugin dla użytkowników nietechnicznych**
- **rekomendowane narzędzia WP i rekomendowane narzędzia dodatkowe**

MATERIAŁY I PRZYDATNE LINKI

ŹRÓDŁA, MATERIAŁY I PRZYKŁADY

- **Dokumentacja WCAG 2.0 (po polsku)**
- **How to Meet WCAG 2.0**
- **Understanding WCAG 2.0**
- **Techniques for WCAG 2.0**
- **ARIA in HTML**
- **WCAG 2.0 - opracowanie Fundacji Widzialni**

ŹRÓDŁA, MATERIAŁY I PRZYKŁADY (C.D.)

- **Przydatna checklista**
- **Badanie dostępności strony www w oparciu o WCAG (pdf)**
- **WCAG 2.0 Podręcznik Dobrych Praktyk (pdf)**
- **Dostępne strony - WCAG 2.0 w skrócie**
- **Internet bez barier - blog**
- **Dostępna strona - Fundacja Integracja**

PRAKTYCZNE PRZYKŁADY DOSTĘPNYCH KOMPONENTÓW

- **WebAIM - świetne źródło opracowań i przykładów**
- **Practical ARIA Examples**
- **Accessible Mega Menu**
- **Dostępny akordeon**
- **Dostępne taby**

NARZĘDZIA

- **Are My Colours Accessible**
- **Generator dostępnych palet kolorystycznych**
- **Colour Contrast Analyser**
- **WAVE (Web Accessibility Evaluation Tool)**
- **TAWDIS**
- **NVDA, JAWS, Window-Eyes**

POWODZENIA W TWORZENIU DOSTĘPNYCH STRON!

S