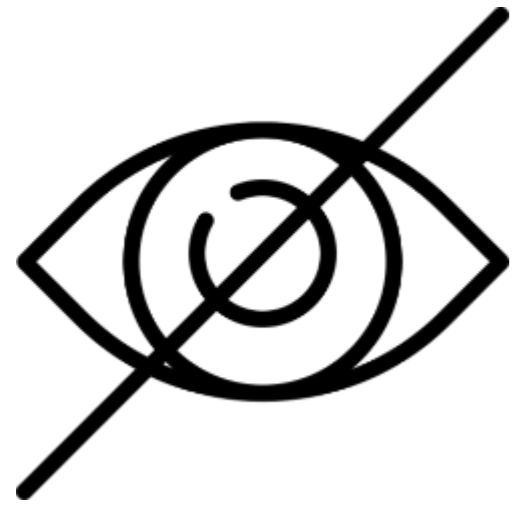


A 1 1 Y

W P R A K T Y C E

d l a p r o g r a m i s t y

Kto jest narażony na wykluczenie cyfrowe?



NIEWIDOMI,
NIEDOWIDZĄCY
I Z DYSFUNKCJAMI
WZROKU



GŁUSI,
GŁUCHONIEWIDOMI



OSOBY Z
NIEPEŁNOSPRAWNOŚCIĄ
FIZYCZNĄ KOŃCZYN



OSOBY Z
NIEPEŁNOSPRAWNOŚCIĄ
INTELEKTUALNĄ



OSOBY
KORZYSTAJĄCE ZE
STARYCH TYPÓW
KOMPUTERÓW



OSOBY STARSZE



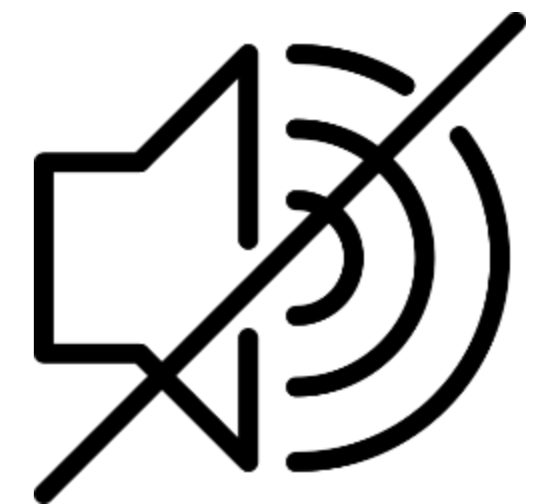
OSOBY KORZYSTAJĄCE
Z URZĄDZEŃ
MOBILNYCH



SPOŁECZNOŚCI
Z BIEDNYCH
REGIONÓW



DYSLEKTYCY



OBCOKRAJOWCY I
KORZYSTAJĄCY ZE
STRON W
GŁOŚNYCH/CICHYCH
MIEJSCACH

Co to jest dostępność?

Zapewnienie dostępu do informacji możliwe największej liczbie użytkowników niezależnie od:



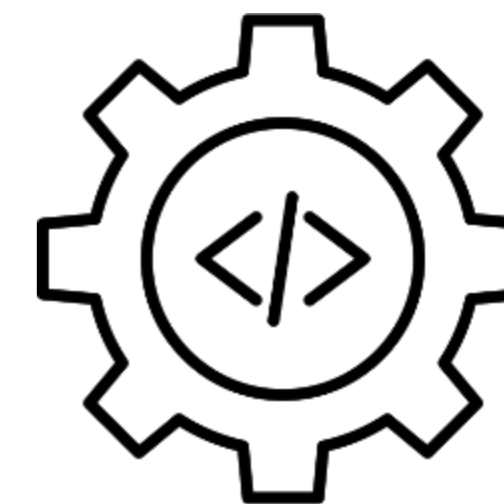
WIEKU



NIEPEŁNOSPRAWNOŚCI



SPRZĘTU



OPROGRAMOWANIA

Samolubna dostępność (selfish accessibility)

projektowanie dla siebie z przyszłości, gdy już będziemy mniej sprawni niż obecnie

Często nie czujemy potrzeby tworzenia dostępnych stron. Pewnie sami jesteśmy zdrowi. Pewnie nasze najbliższe otoczenie też nie ma problemu z poruszaniem się po Internecie. Nie rozumiemy, jak ważne są dostępne strony internetowe. Czasem jednak nadchodzi ten moment, kiedy uświadamiamy sobie, że problem dostępności może dotyczyć także nas – dziś, jutro, za rok, za 50 lat. Nieważne kiedy. Istotne, że to będzie NASZ problem. Jeśli to myślenie motywuje nas do projektowania stron zgodnych z WCAG – mamy do czynienia z samolubną dostępnością.

Projektowanie inkluzywne (inclusive design)

tworzenie treści użytecznych dla jak największej grupy osób, z naciskiem na etap ich projektowania

Termin projektowania inkluzywnego można analizować w kontekście stron internetowych, ale też np. architektury, czy wzornictwa.

Dla nas, ludzi tworzących Internet, ważny jest jeden wniosek – nie zostawiamy kwestii WCAG developerom. O dostępności powinny myśleć wszystkie zespoły zaangażowane w tworzenie finalnego produktu: od projektantów, przez redaktorów, developerów, managerów, konsultantów, aż po testerów. I każdą jednostkę pomiędzy nimi.

Idea projektowania inkluzywnego zaczyna się od ciekawości – jak widzą dany produkt użytkownicy inni niż ja? Jak z niego korzystają? Jak możliwości technologiczne, językowe, poznawcze, fizyczne pozwalają im na wygodne poruszanie się po stronie internetowej? Czy jest ona dla nich w pełni funkcjonalna?

Podstawowe pojęcia

W3C

W3C (World Wide Web Consortium) to organizacja, która zajmuje się ustanawianiem standardów tworzenia stron WWW. Dodatkowo dba o szereg innych okołointernetowych kwestii, np. prawa autorskie, bezpieczeństwo w sieci. W3C oferuje bezpłatne walidatory HTML i CSS, dzięki którym możemy sprawdzić czy dana strona jest zgodna ze standardami.

W konsorcjum działają grupy tematyczne zajmujące się różnymi zagadnieniami. Jedną z grup tematycznych jest WAI.

WAI

WAI (Web Accessibility Initiative) to wydzielona grupa tematyczna W3C, która przygotowuje wytyczne dotyczące szeroko pojętej dostępności. Najważniejsze wskazówki zawarte są w dokumentacjach określanych jako:

- **WCAG 2.0** (Web Content Accessibility Guidelines)
zbiór zasad projektowania dostępnych stron, dla twórców, projektantów i redaktorów stron www
- **ATAG 2.0** (Authoring Tool Accessibility Guidelines)
wytyczne dla twórców oprogramowania stron, narzędzi służących do tworzenia i publikacji treści na stronach www (np. edytory HTML, systemy CMS).
- **UAAG 2.0** (User Agent Accessibility Guidelines)
zbiór reguł dla twórców aplikacji klienckich (np. przeglądarki internetowe, odtwarzacze multimedialne czy technologie asystujące).



Podstawowe pojęcia

WCAG

WCAG (Web Content Accessibility Guidelines) to dokument opisujący standard dostępności treści internetowych.

Specyfikacja WCAG 2.0 ma ściśle określoną strukturę. Całość wytycznych została podzielona na 4 główne zasady:

- **PERCEPCJA** – dotyczy zapewnienia dostępu do informacji za pośrednictwem co najmniej jednego ze zmysłów użytkownika,
- **FUNKCJONALNOŚĆ** – komponenty interfejsu (zwłaszcza nawigacja) muszą być funkcjonalne dla wszystkich użytkowników (powinny pozwalać na interakcję),
- **ZROZUMIAŁOŚĆ** – wskazuje, że poza dostępną i funkcjonalną formą strony, również czytelność i zrozumiałość interfejsu oraz samej treści jest niezbędna,
- **RZETELNOŚĆ** – treść musi być wystarczająco rzetelna, aby mogła być poprawnie interpretowana przez wielu różnych klientów użytkownika, włączając technologie asystujące.

Cała specyfikacja WCAG 2.0 została ponadto podzielona na trzy poziomy dostępności:

- A – wymóg konieczny, minimalny,
- AA – wymóg rekomendowany, średni,
- AAA – wymóg możliwy, maksymalny.

Podstawowe pojęcia

WAI-ARIA

WAI-ARIA (WAI Accessible Rich Internet Applications) – dokumentacja stworzona przez WAI; zestaw konkretnych narzędzi, których stosowanie ma na celu poprawę dostępności aplikacji internetowych przez stworzenie dodatkowej warstwy niosącej informacje istotne dla technologii wspomagających.

WAI-ARIA definiuje zestaw atrybutów dodawanych do znaczników HTML, które wzbogacają go o semantykę i różnego rodzaju metadane, pozwalając na lepsze zrozumienie zawartości.

Atrybuty ARIA nadają semantyczne znaczenie niesemantycznym elementom HTML, bądź też zmieniają semantyczne znaczenie elementów, które zostały “przerobione” przy użyciu JavaScript na komponenty służące do innych celów niż jest to opisane w dokumentacji HTML. Dzięki nim stosowne, dodatkowe informacje są przekazywane technologiom asystującym.

Wsparcie dla WAI-ARIA jest zróżnicowane, ale obecnie ani przeglądarki, ani technologie wspomagające nie wspierają tej specyfikacji w 100%. Mimo to stosowanie atrybutów WAI-ARIA w kodzie jest zalecane. Jednocześnie, jeśli to możliwe, należy używać natywnych semantycznych znaczników HTML zamiast stosowania atrybutów ARIA. Elementy WAI-ARIA dodane do kodu HTML nie powodują żadnych zmian w sposobie wyświetlania się strony.

Atrybuty WAI-ARIA można podzielić na:

- **role** – opisany element jest czymś innym niż wynika to z jego znacznika. Generalnie, jeśli nie ma uzasadnionej potrzeby, nie należy zmieniać natywnego znaczenia znaczników HTML. Role stosujemy najczęściej, kiedy HTML nie posiada odpowiednika dla danej roli.
- **właściwości** – cechy lub dodatkowe znaczenia danego elementu, które nie są dostępne w standardzie HTML. Posiadają prefix aria-, ich wartości są różne.
- **statusy** – aktualny stan elementu. Posiadają prefix aria-, ich wartości to “true” albo “false”.

WCAG w praktyce

DLA PROGRAMISTÓW

10 dobrych praktyk
tworzenia dostępnych aplikacji,
które nie zabrają Ci wiele czasu i energii.

Semantyczny HTML

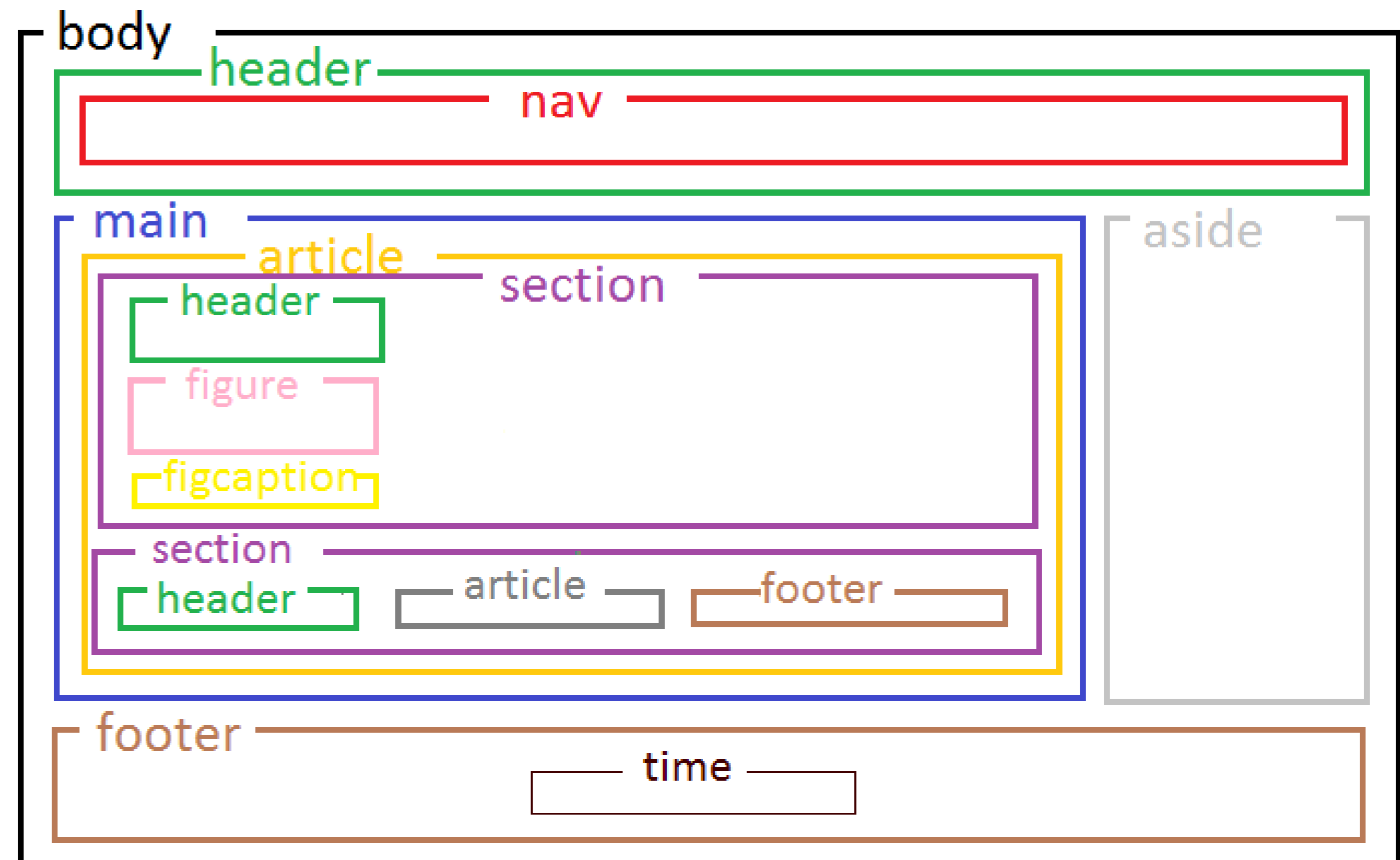
Przestań używać divów i zacznij pisać semantyczny (samoopisujący się) HTML

Tagi nie-semantyczne:

div, span

Tagi semantyczne:

body, header, nav, main, article, section, figure, figcaption, footer, aside, cite, blockquote, abbr, h1-h6, kbd



Semantyczny HTML

Przykład - akordeon

```
<details>
<summary>Languages Used</summary>
<p>This page was written in HTML and CSS. The CSS was compiled from
SASS. Regardless, this could all be done in plain HTML and CSS</p>
</details>
```

```
<details>
<summary>How it Works</summary>
<p>Using the sibling and checked selectors, we can determine the
styling of sibling elements based on the checked state of the
checkbox input element. </p>
</details>
```

▼ Languages Used

This page was written in HTML and CSS. The CSS was compiled from SASS. Regardless, this could all be done in plain HTML and CSS

► How it Works

Semantyczny HTML

Przykład – pasek postępu


```
<label for="upload">Upload progress:</label>

<meter id="upload" name="upload"
  min="0" max="100"
  low="33" high="66" optimum="80"
  value="50">
  at 50/100
</meter>

<hr/>

<label for="file">File progress:</label>

<progress id="file" max="100" value="70"> 70% </progress>
```

Upload progress: 

File progress: 

Semantyczny HTML

Przykład – input

```
<label for="date">Enter date:</label>  
<input type="date" id="date"/>
```

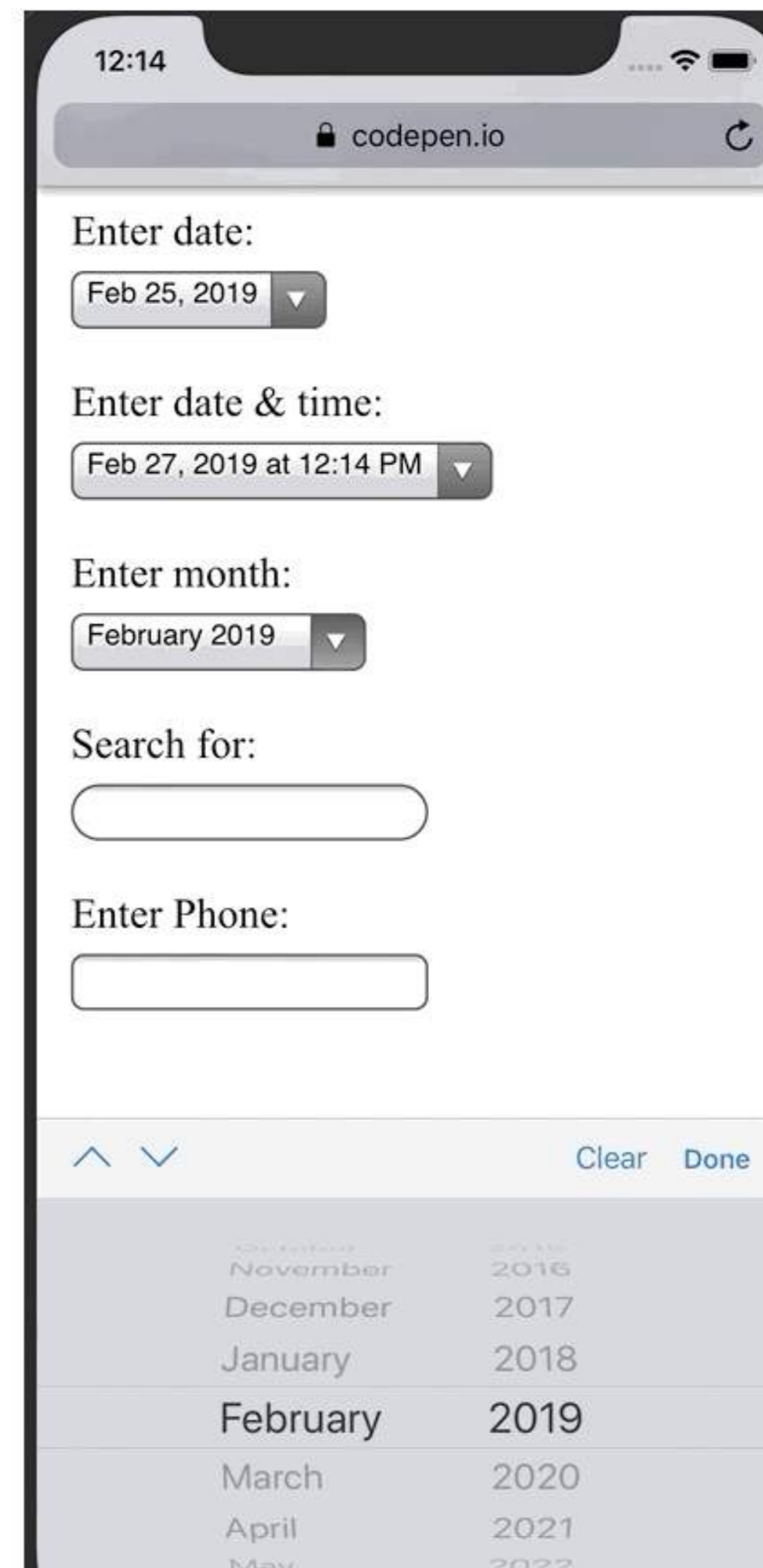
```
<label for="datetime">Enter date & time:</label>  
<input type="datetime-local" id="datetime"/>
```

```
<label for="month">Enter month:</label>  
<input type="month" id="month"/>
```

```
<label for="search">Search for:</label>  
<input type="search" id="search"/>
```

```
<label for="tel">Enter Phone:</label>  
<input type="tel" id="tel">
```

[Źródło](#)



Nagłówki

W prawidłowy sposób korzystaj z nagłówków

- ✓ Przygotuj system czcionek z odpowiednimi wielkościami umożliwiającymi ich prawidłowe użycie w kodzie HTML
- ✓ możliwość czytania strony za pomocą czytników

Problem:

```
<h1>Game of Thrones</h1>  
<p>First paragraph</p>
```

```
<h2>Season 1</h2>  
<p>Paragraph</p>
```

```
<h3>Episode 1</h3>  
<p>Paragraph</p>
```

```
<h2>Episode 2</h2> // This one is incorrect  
<p>Paragraph</p>  
<h3>Episode 3</h3>
```



```
Game of Thrones  
  Season 1  
    Episode 1  
  Episode 2  
    Episode 3
```

Nagłówki

W prawidłowy sposób korzystaj z nagłówków

Rozwiązanie: Zmiana guideline albo...

```
h2,
.h2 {
  font-size: 3rem;
}

h3,
.h3 {
  font-size: 2rem;
}
```

```
<h2>Season 1</h2>
<p>Paragraph</p>

<h3>Episode 1</h3>
<p>Paragraph</p>

<h3 class="h2">Episode 2</h3>
<p>Paragraph</p>
```

Outline

Pod żadnym pozorem nie usuwaj wskaźnika elementu, na którym jest focus



NIE dla **outline: none;**



Brzydki, domyślny outline – Chrome



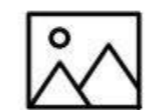
ładny, zgodny z identyfikacją wizualną projektu, customowy outline

Obrazki

Dekoracyjne obrazki staraj się umieszczać na stronie za pomocą CSS.



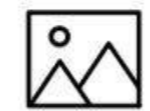
Jeśli używasz tagu ``,
uzupełniaj tekst alternatywny
dla obrazków – atrybut `alt`



dekoracyjny

Pozostaw pusty atrybut lub użyj CSS.

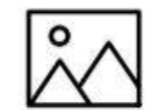
```
alt=""
```



przedstawia tekst

Powtórz tekst z obrazka.

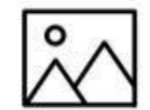
```
alt="Ten sam tekst, jaki przedstawia obrazek"
```



przedstawia akcję

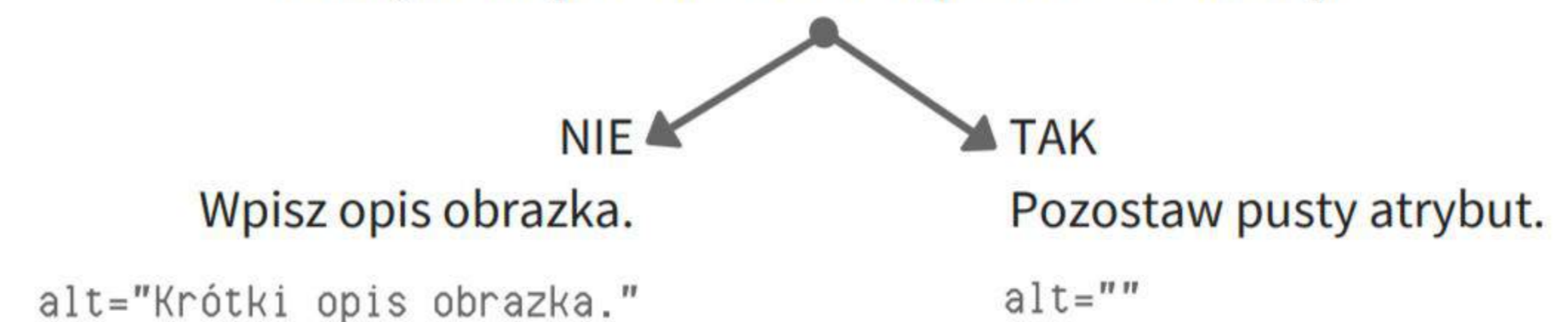
Wpisz rezultat interakcji z danym elementem.

```
alt="Dodaj do koszyka"
```



informacyjny i mogę opisać go 1 zdaniem

Czy informacja przekazywana przez obrazek pojawia się
w innym miejscu na stronie w postaci tekstowej?



Obrazki

informacyjny i NIE mogą opisać go 1 zdaniem (np.: graf, wykres...)

Wprowadź 2 opisy obrazka:

1. krótki – za pomocą atrybutu alt
2. długi – za pomocą atrybutów longdesc/aria-labelledby lub elementów figure oraz figcaption

```

```

```
<p id="desc">Długi opis obrazka.</p>
```

```
<figure>
```

```
  
```

```
  <figcaption>Długi opis obrazka.</figcaption>
```

```
</figure>
```

mapa

Uzupełnij opis zarówno dla elementu , jak i każdego elementu <area>.

```

```

```
<map name="picture">
```

```
  <area shape="[...]" coords="[...]" href="[...]" alt="Opis fragmentu obrazka.">
```

```
  [...]
```

```
</map>
```

... kilka obrazków

Wprowadź opis do jednego z obrazków, dla pozostałych pozostaw pusty atrybut.

```

```

```

```

```
[...]
```


Formularze - labels

Zawsze używaj <label> przy <input>

```
<label>  
  Your name:  
  <input type="text" name="name" placeholder="Michael Scott" />  
</label>
```

```
<label htmlFor="company">Your company</label>  
<input type="text" name="company" id="company" placeholder="ACME Inc." />
```

Formularze - labels

Zawsze używaj <label> przy <input>

Również przy inpucie, który posiada placeholder, ponieważ czytniki go nie widzą!

```
let style = {
  border: 0,
  clip: 'rect(0 0 0 0)',
  height: '1px',
  width: '1px',
  margin: '-1px',
  padding: 0,
  overflow: 'hidden',
  position: 'absolute'
}

function VisuallyHidden({ Component = 'span', ...rest }) {
  return <Component style={style} {...rest} />
}
```

```
<VisuallyHidden
  Component='label'
  htmlFor='username'>
  Username:
</VisuallyHidden>
<input
  placeholder='Your username'
  id='username' />
```


Formularze - labels

Zawsze używaj <label> przy <form>

```
<form aria-labelledby="attend-title">
  <h2 id="attend-title">Attend React Zurich</h2>

  <label>
    Your name:
    <input type="text" name="name" placeholder="Michael Scott" />
  </label>

  <label htmlFor="company">Your company</label>
  <input type="text" name="company" id="company" placeholder="ACME Inc." />

  <button>RSVP</button>
</form>
```

Ikony

Jeśli tylko to możliwe – twórz elementy dekoracyjne za pomocą CSS.
W innym przypadku - ukrywaj je dla czytników.

See all Attendees →

```
<a
  href="/attendees"
>
  See all Attendees
  <svg
    aria-hidden="true"
    xmlns="http://www.w3.org/2000/svg"
    viewBox="0 0 24 24"
  >
    ...
  </svg>
</a>
```

FOLLOW US ON



```
<a
  href="https://www.facebook.com/ACME-Inc"
  aria-label="Find us on facebook"
>
  <svg aria-hidden="true">...</svg>
</a>
```

Nawigacja

Zawsze umożliwiał ominięcie głównej nawigacji podczas przeglądania strony za pomocą klawiatury, czytnika.

```
<a class="focusable visually-hidden" href="#main-content">Skip to main content</a>
...
<nav class="menu-main">
  <ul class="menu">
    <li class="menu-item"><a href="/about">About</a></li>
    <li class="menu-item"><a href="/blog">Blog</a></li>
    <li class="menu-item"><a href="/contact">Contact</a></li>
  </ul>
</nav>
...
<main id="main-content">
...
</main>
```

```
.visually-hidden {
  position: absolute;
  left: -10000px;
  top: auto;
  width: 1px;
  height: 1px;
  overflow: hidden;
}

.focusable.visually-hidden:focus {
  position: relative;
  left: 0;
  width: auto;
  height: auto;
  overflow: auto;
}
```


Linki

🔗 Linki zawsze muszą wyróżniać się wizualnie (kolor, podkreślenie).

🔗 Staraj się nadawać linkom zrozumiały opis.

```
<a href="#"> Zobacz więcej informacji o Onwelo </a>
```

🔗 Jeśli makieta wskazuje inaczej – nadal linkowi atrybut title z odpowiednim opisem.

```
<a href="#" title="Zobacz więcej informacji o Onwelo">Onwelo</a>
```

🔗 Niektóre tagi (div, span, p) nie mogą mieć stanu focus – są niedostępne z poziomu klawiatury. Jeśli tworzysz z nic odnośniki – pamiętaj o nadaniu im tabindex=0 albo opakuj je w tag <a>

```
<div onclick="#" tabindex="0">Zobacz więcej informacji o Onwelo</div>  
<a href="#"><div>Zobacz więcej informacji o Onwelo</div></a>
```

CAPTCHA

CAPTCHA OBRAZKOWA NIE JEST DOSTĘPNA!

Captcha dźwiękowa jest wg relacji osób niepełnosprawnych – bardzo trudna do zrozumienia!

Dyslektycy mają problem z każdym rodzajem captcha!

Nie używaj jej!

ALTERNATYWY

📌 Weryfikacja biometryczna – odcisk palca, skan tęczówki oka, skan twarzy

📌 Drag and drop



[Źródło](#)

📌 Honeypot — niewidoczny checkbox, który jest z reguły zaznaczany przez boty.

📌 Filtr czasowy - jeśli formularz uzupełniony zostanie nienaturalnie szybko (jak robią to automaty), system zablokuje zgłoszenie

📌 Czarne listy serwerów rozsyłających spam oparte na DNS (DNSBL)
Bazy danych np. NiX Spam i SORBS.
Jeśli formularz zostanie wypełniony przez bot znajdujący się w bazie danych (np. NiX Spam, SORBS), przesyłanie formularza zostanie odrzucone.

📌 Ekran potwierdzenia wykonania akcji – tylko, kiedy logicznie ma sens

📌 Weryfikacja SMS, email

📌 Filtracja danych w formularzu – najlepiej na serwerze

tabindex

Testuj kolejność elementów na stronie. Koryguj je semantyką lub tagiem **tabindex**

Atrybut **tabindex** służy do określania kolejności w nawigowaniu po elementach, które mogą złapać *focus*, czyli mogą stać się aktywne / gotowe do interakcji na danej stronie.

tabindex="-1" element wykluczony z nawigacji - czyli nie będzie można aktywować go za pomocą klawisza Tab na klawiaturze
W dalszym ciągu będzie można ustawić *focus* na elemencie za pomocą kodu JS lub przy pomocy odpowiedniego linka na stronie.
tabindex="-1" ma takie samo działanie jak **tabindex="-100"**

Przykład użycia:

wyskakujące okienka (toast, alert, snackbar...), należy jednak pamiętać o umożliwieniu szybkiego zamknięcia takiego okna za pomocą klawisza Esc.

tabindex="0" element dołączony do listy elementów dostępnych za pomocą klawisza Tab
Nie jest wymagany dla elementów, które domyślnie przyjmują stan *focus* (input, button, select, textarea, a, area)

Przykład użycia:

Customowe elementy, np. ``

tabindex="1" Nadanie kolejności na liście elementów dostępnych za pomocą klawisza Tab w zależności od wartości atrybutu
Korzystajmy tylko w bardzo specyficznych przypadkach!

Przykład użycia:

Modal – kolejność dostępnych przycisków – ustawienie domyślnego stanu *focus* na przycisk zatwierdzający

Bonus

Checklisty dostępnej strony

- ★ Nie tylko dla developerów
- ★ Na podstawie dokumentacji
- ★ Z dokumentacji
- ★ Od wuhcag.com od druku
- ★ Życiowa checklista
- ★ Krócej się nie da...
- ★ ... ale da się bardziej szczegółowo
- ★ Na dwoje babka wróżyła
- ★ Pigułka on-line
- ★ Dla humanistów
- ★ APLIKACJA (Google Play / App Store)
- ★ Po prostu – checklista
- ★ PLAKATY



Źródła

- ✓ **Specyfikacja WCAG 2.1**
<https://www.w3.org/TR/WCAG21/>
- ✓ **Materiały fundacji Widzialni.org**
<http://widzialni.org/>
- ✓ **Blog Ach te Internety**
<https://ach-te-internety.pl/>
- ✓ **Blog Bitsrc**
<https://blog.bitsrc.io/>
- ✓ **Auditus.io**
<https://www.auditus.io/>
- ✓ **WebAIM**
<https://webaim.org/articles/>
- ✓ **Dev.to**
<https://dev.to/lkopacz/3-simple-tips-to-improve-keyboard-accessibility-5hc8>
- ✓ **Piotr Nalepa**
<https://blog.piotrnalepa.pl>
- ✓ **Internet bez barier**
<http://internet-bez-barier.com>

