

# **Web Accessibility**

**Interface Pessoa-Máquina - 25/26 - LEI / UM**

**Hugo Pacheco**

**hpacheco@di.uminho.pt**

# The Web

- A Internet é um lugar de igualdade sem precedentes (para o bem e para o mal)

*“The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect”* (Tim Berners-Lee)

- Acesso universal para toda a gente, independentemente de:
  - Hardware, software
  - Language, location
  - Abilities, disabilities
  - ...

# Disability?

- World Health Organization
  - **Disability** = “The outcome of the interaction between a person [...] and the **environment and attitudinal barriers** they may face”
    - ~16% da população mundial é afetada por uma incapacidade significativa
  - Algumas categorias de incapacidades
    - Learning and applying knowledge
    - General tasks and demands
    - Communication
    - Basic physical mobility, Domestic life, and Self-care
    - Interpersonal interactions and relationships
    - Community, social and civic life, including employment

# Acessibilidade é usabilidade para mais gente

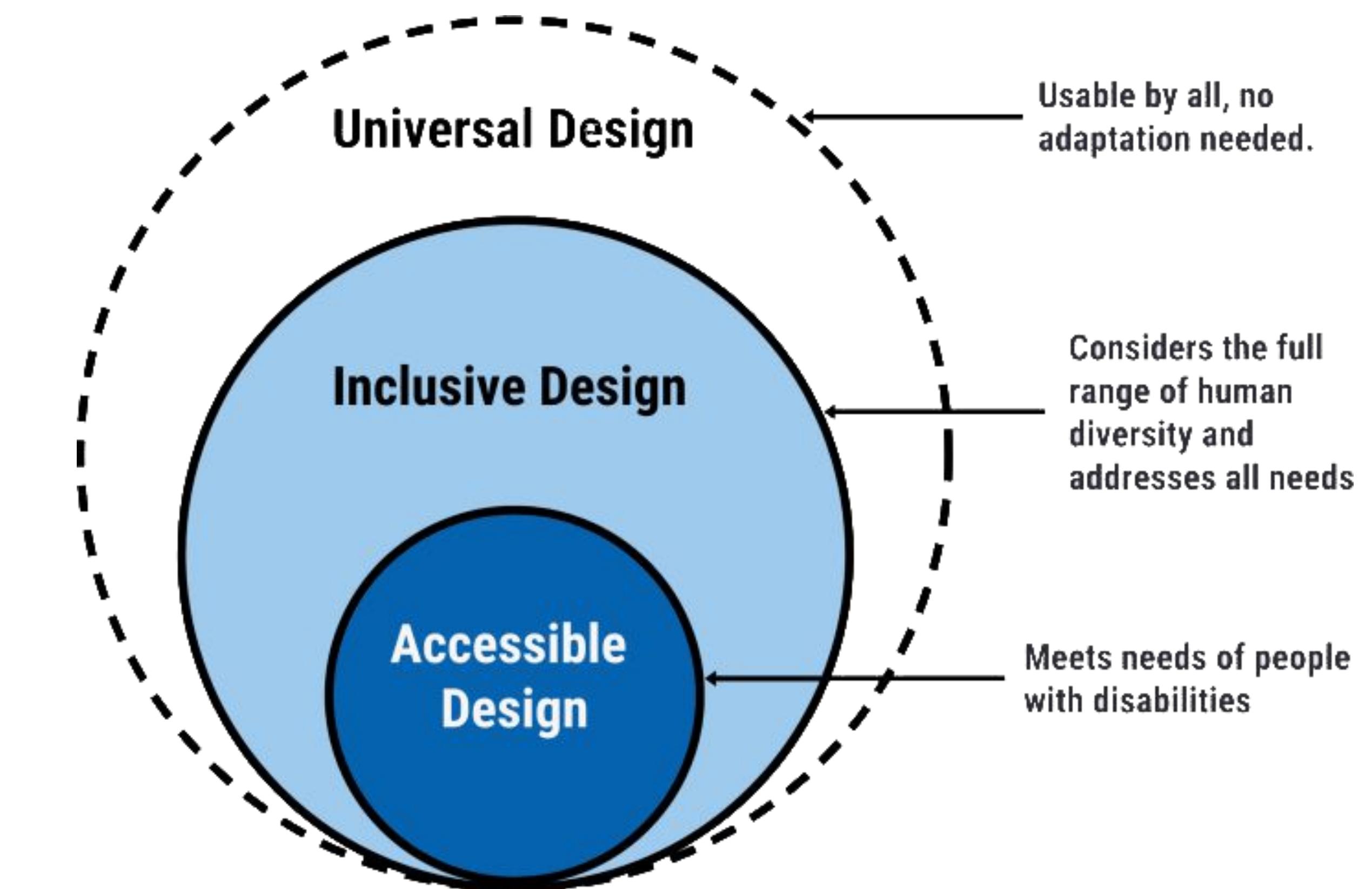
- ISO 9241 - Ergonomics of human-system interaction
  - Part 11: Usability: Definitions and concepts
    - **Usability** = “The effectiveness, efficiency and satisfaction with which a **specified set of users** can achieve a specified set of tasks in a particular environment”
  - Part 20: An ergonomic approach to accessibility within the ISO 9241 series
    - **Accessibility** = “The usability of a product, service, environment or facility by people with the **widest range of capabilities**”

# Web Accessibility

- Designing and developing websites, applications, and digital content in a way that ensures they can be used by **everyone**, including people with **disabilities**
- More relevant types of **disabilities** in this context
  - ⌚ Visual ⇒ Screen readers, colour contrast
  - 👂 Auditory ⇒ Captions, transcripts
  - ♿ Motor ⇒ Keyboard accessibility, voice commands
  - 🧠 Cognitive ⇒ Clear language, consistent navigation

# Quem beneficia?

- Toda a gente
- Quem está num espaço público
- Quem deixou os óculos em casa
- Quem usa dispositivos móveis
- À medida que se envelhece
- Com incapacidades

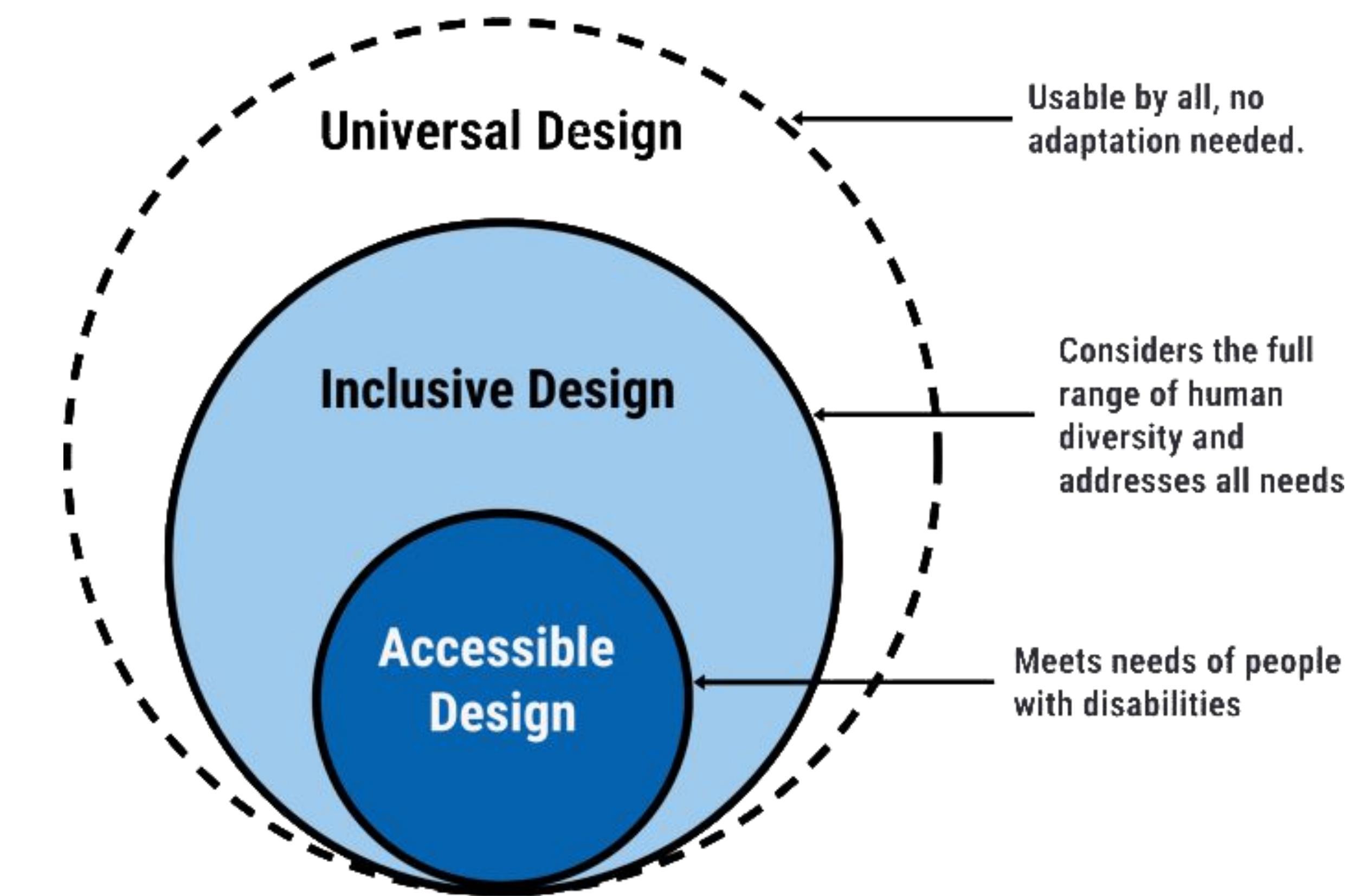


# Accessible < Inclusive < Universal Design

- “*The goal is **universal design** that's integrated and equal: Don't make disabled people use a different Web structure but make it so they can use it too.*”

(Dan Fruchterman)

- Não apenas ajustar tecnologia às várias incapacidades ou necessidades de vários utilizadores
- A própria tecnologia deve ser naturalmente adaptativa, e.g.:
  - HTML: Markup Semântico
  - CSS: Responsive Web Design



# Web Accessibility: importante?

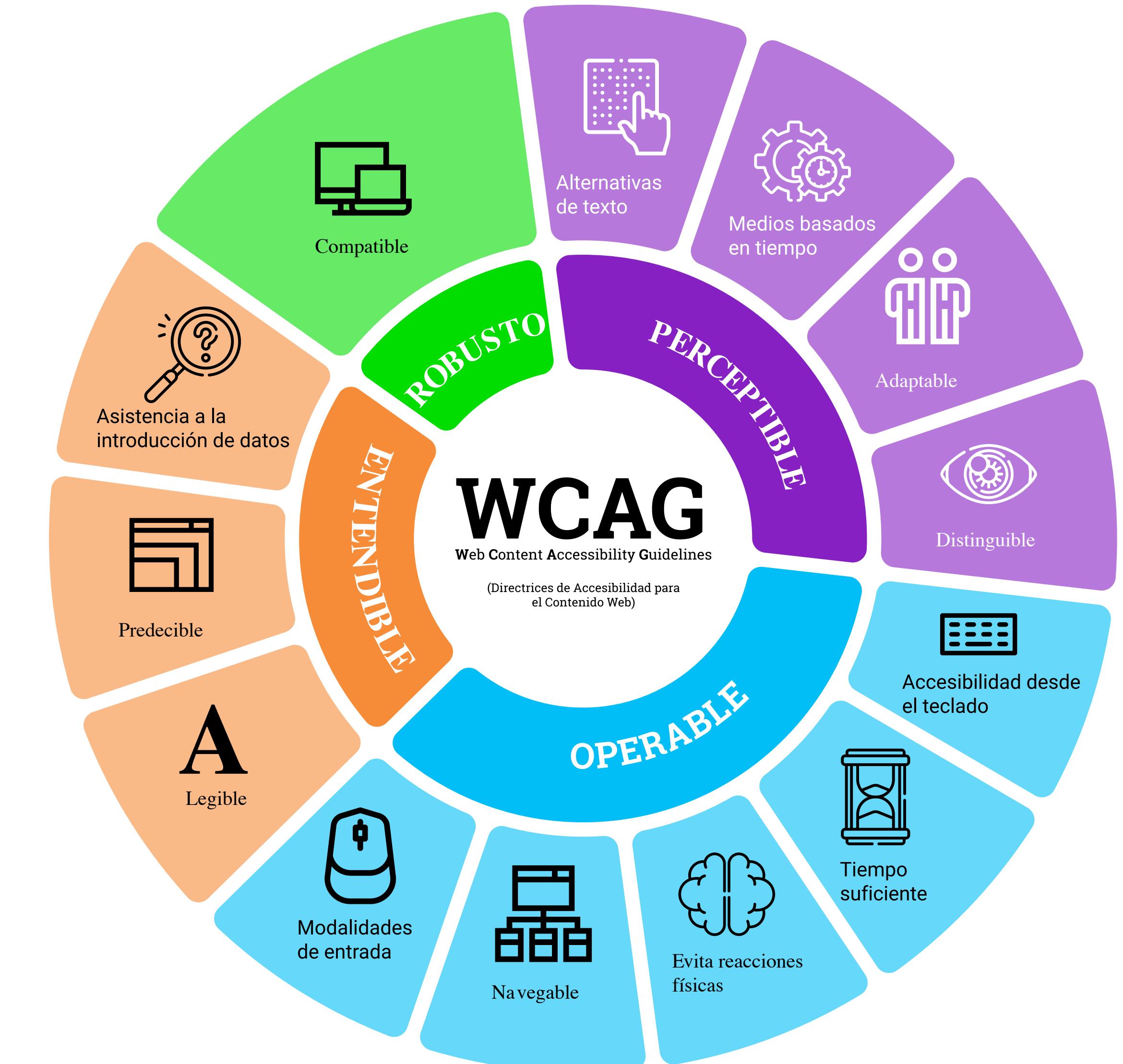
- Um dever ético
  - Garantir acesso equitativo a informação e serviços
- Vantagens de mercado
  - Alcançar audiências mais alargadas
  - Melhorar user experience e usabilidade geral
  - Aumentar o ranking do site em motores de busca  
Search Engine Optimization (SEO)
  - Aumentar reputação da marca

# Web Accessibility: uma questão de lei

- US Americans with Disabilities Act (1990, update 2024)
  - Proíbe discriminação com base em incapacidades
- US Rehabilitation Act (1973, update 2017)
  - Obriga agências federais a tornar a sua informação e tecnologias acessíveis
- EU European Accessibility Act (2019)
  - Obriga um vasto leque de produtos e serviços a serem acessíveis
- EU Web Accessibility Directive (2016)
  - Obriga todos os sites e aplicações mobile de entidades públicas a serem acessíveis

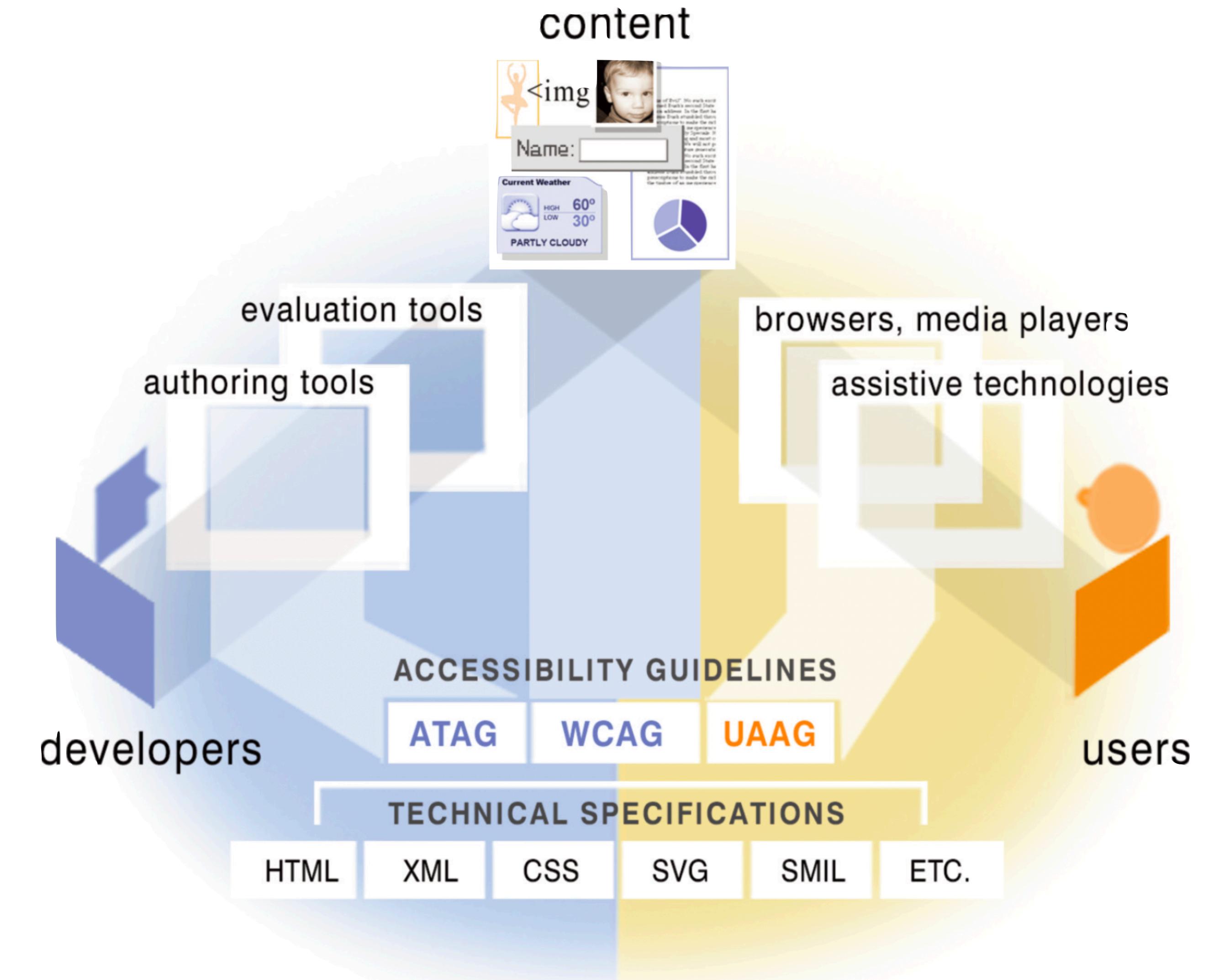
# Web Content Accessibility Guidelines

- Standard W3C
  - Versão atual [WCAG 2.2](#)
  - Draft mais recente [WCAG 3.0](#)
- Um conjunto de 11 guidelines para web design mais acessível
  - Organizadas em 4 principles
  - Avaliadas com 86 success criteria
- 💡 Todas as leis anteriores se alinham e exigem conformidade com WCAG



# WCAG+

- WCAG para conteúdo (websites, apps)
- Outros standards W3C associados
  - ATAG (Authoring Tools Accessibility Guidelines) para ferramentas de criação de conteúdo
  - UAAG (User Agent Accessibility Guidelines) para ferramentas de leitura de conteúdo (browsers, screen readers)

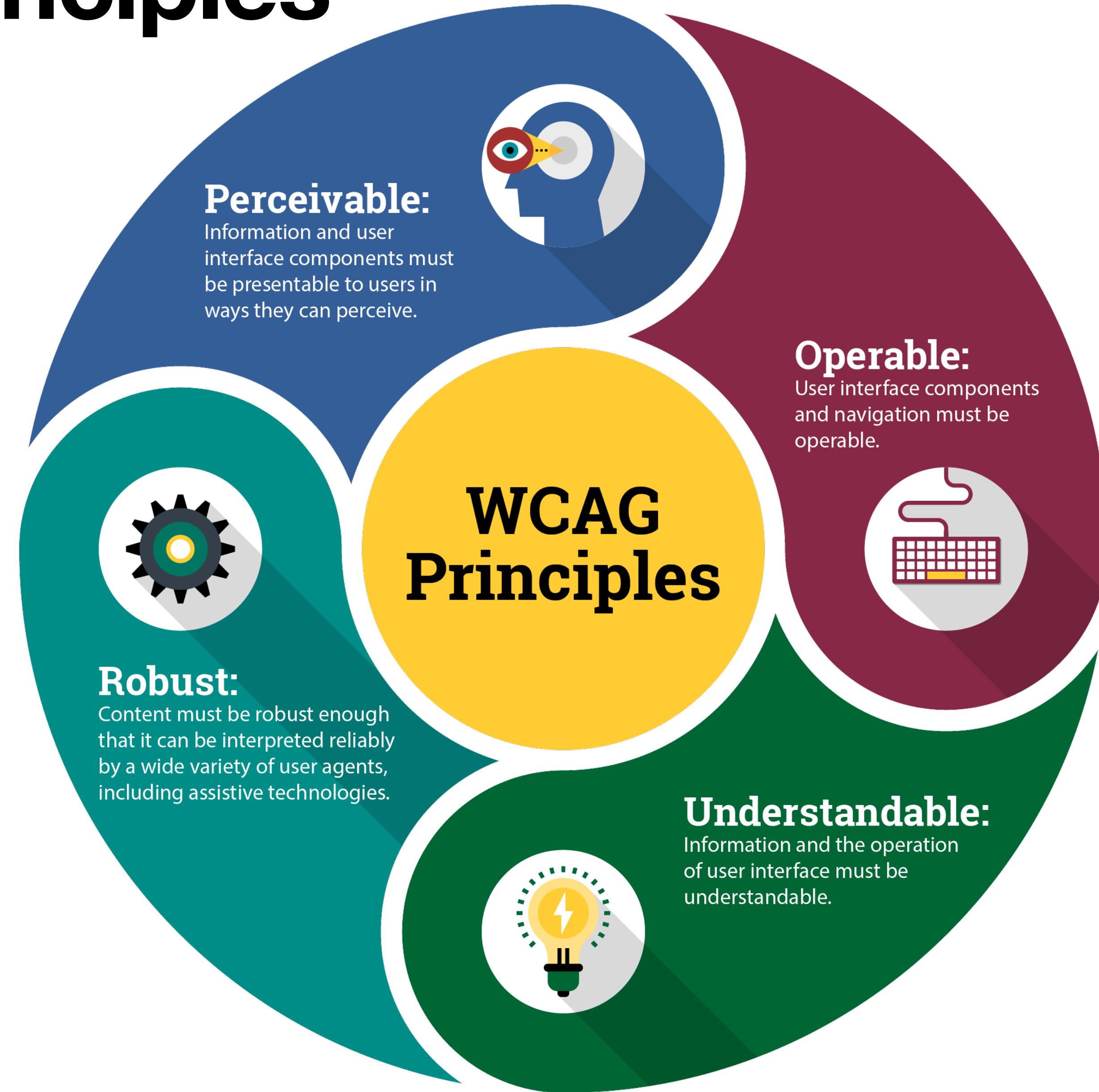


Rodríguez Vázquez, Silvia, and Jesús Torres del Rey. "A communicative approach to evaluate web accessibility localisation using a controlled language checker: the case of text alternatives for images." (2012).

# Assistive Technologies

- **Screen Readers:** Convert text into speech or braille, allowing visually impaired users to navigate websites (e.g., JAWS, NVDA, VoiceOver)
- **Screen Magnifiers:** Enlarge text and images on the screen for users with low vision (e.g., ZoomText).
- **Speech Recognition Software:** Allow users to control their computer and input text using voice commands (e.g. Google Voice Typing, Microsoft Dictate).
- **Alternative Input Devices:** Devices such as eye trackers and motion trackers enable users with motor impairments to interact with their computer.
- ...

# WCAG Principles



# WCAG Principle #1: Perceivable

- Information and user interface components must be presented to users in ways they can perceive
- **1.1 Text Alternatives**
  - For non-text content, e.g. images  
``
- **1.2 Time-based Media**
  - E.g., offering captions for audio content

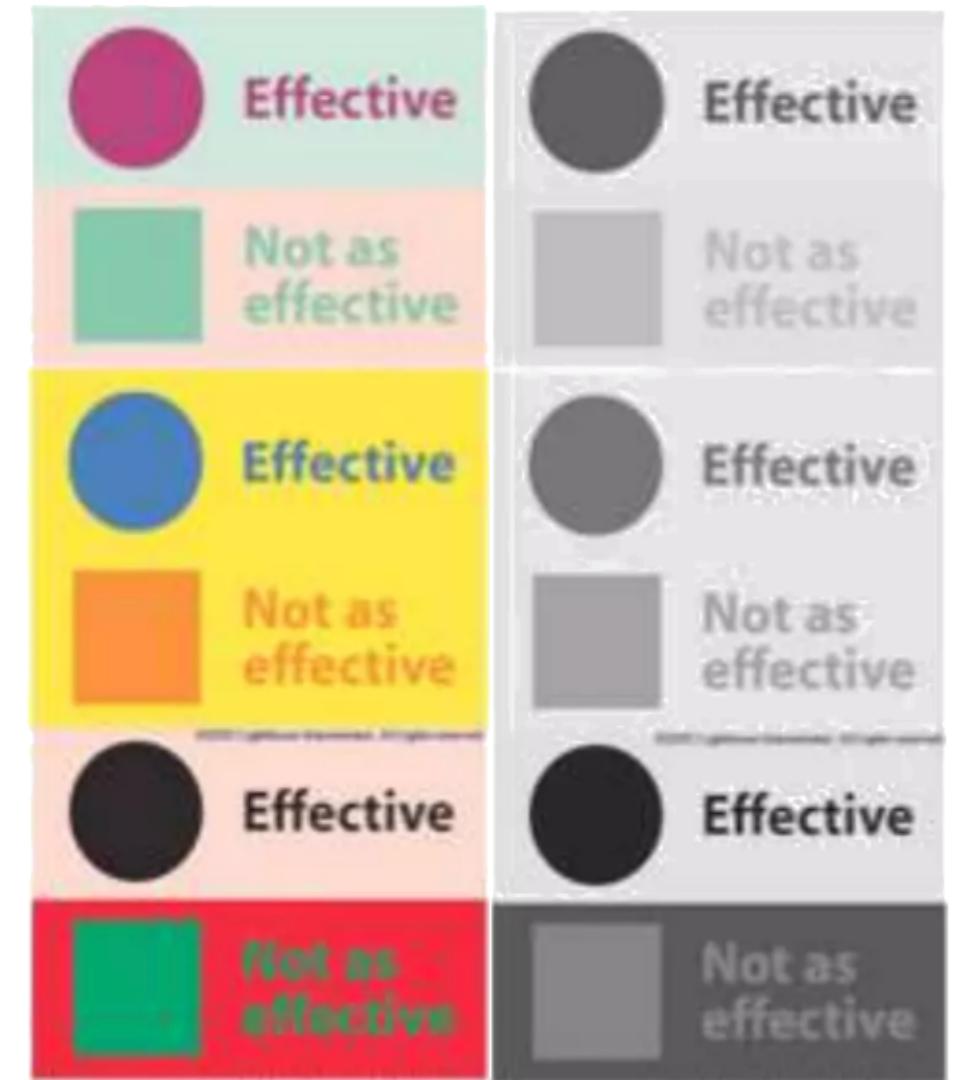
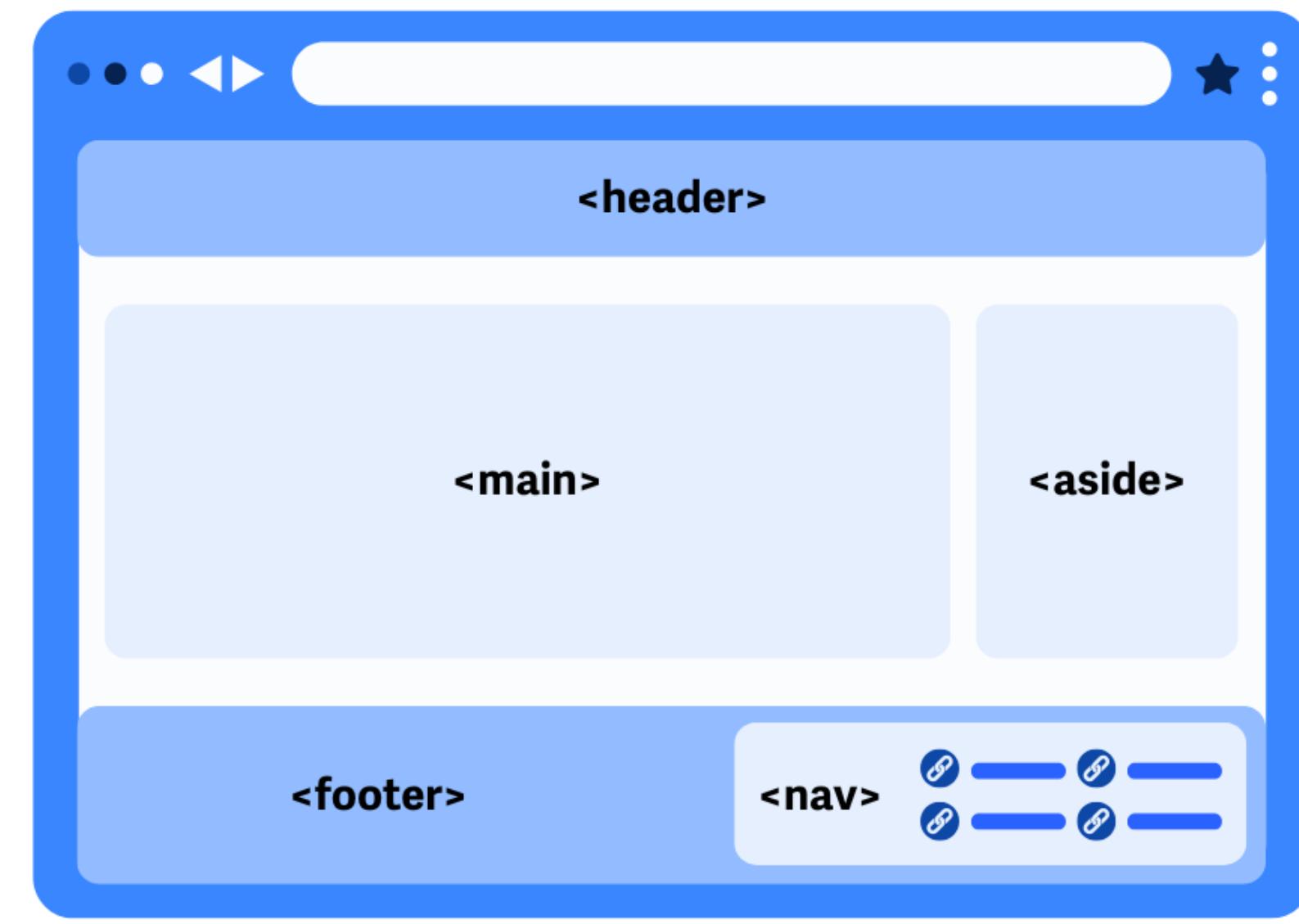


Red Fox



# WCAG Principle #1: Perceivable

- Information and user interface components must be presented to users in ways they can perceive
- **1.3 Adaptable**
  - Clear structure (semantic HTML markup, responsive CSS layout). E.g., no (exclusive) reliance on shape, size, location, color or sound to navigate
- **1.4 Distinguishable**
  - E.g., ensuring sufficient color contrast



# WCAG Principle #2: Operable

- Users must be able to operate the user interface components and navigation

- **2.1 Keyboard Accessible**

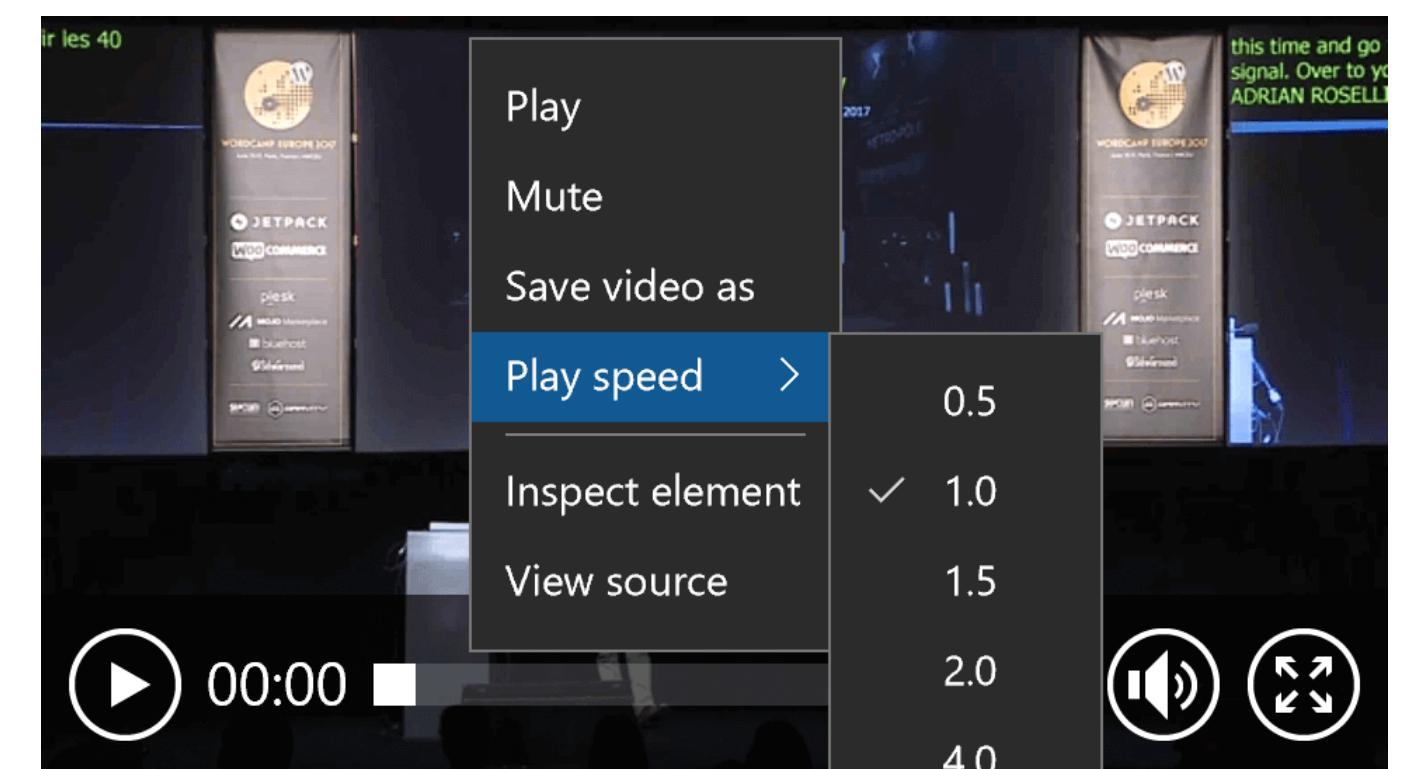
```
<nav>
  <a href="#home" tabindex="1">Home</a>
  <a href="#about" tabindex="2">About</a>
</nav>
<input tabindex="3">...</input>
```

The tabindex attribute controls the order in which elements receive focus when navigating with the Tab key



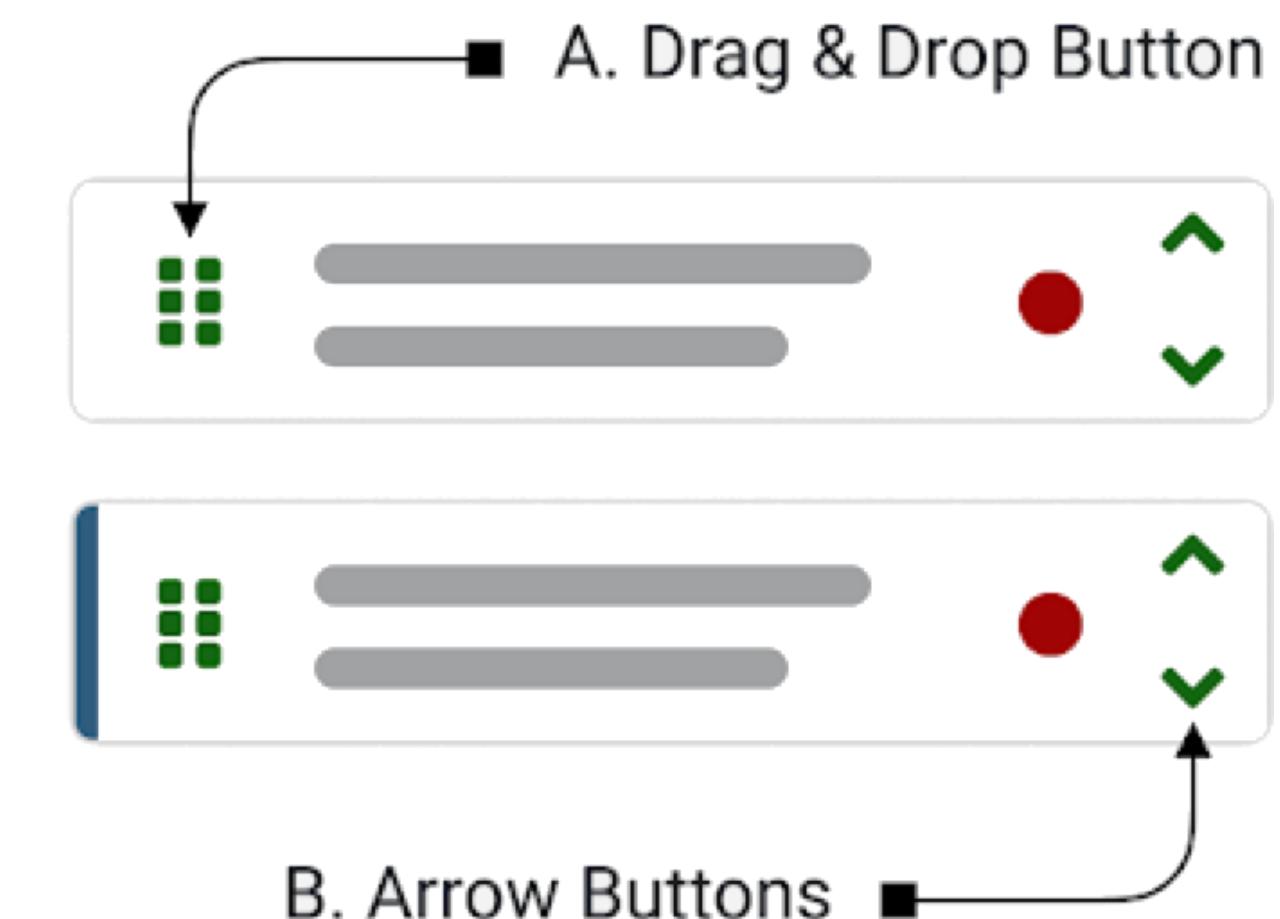
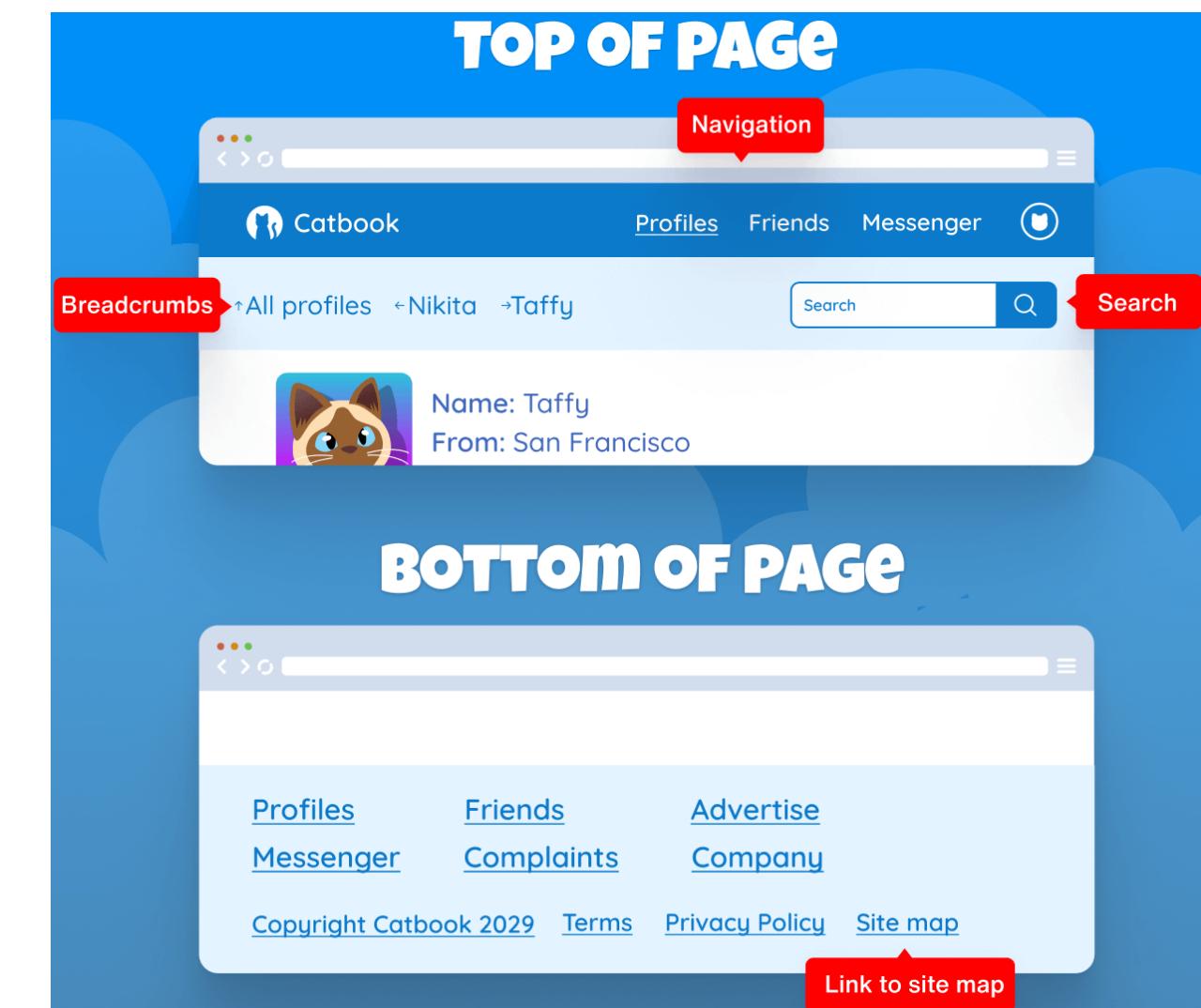
- **2.2 Enough Time**

- E.g., no automatic redirects, user-adjustable time to read and use content



# WCAG Principle #2: Operable

- Users must be able to operate the user interface components and navigation
- **2.3 Seizures and Physical Reactions**
  - E.g., eliminate blinking screen features
- **2.4 Navigable**
  - E.g., alternative ways of finding other site pages (table of contents, site map, etc)
- **2.5 Input Modalities**
  - E.g., support alternative inputs beyond keyboard and mouse movement



# WCAG Principle #3: Understandable

Language complexity heatmap

- Users must be able to understand the information and operation of the user interface

- **3.1 Readable**

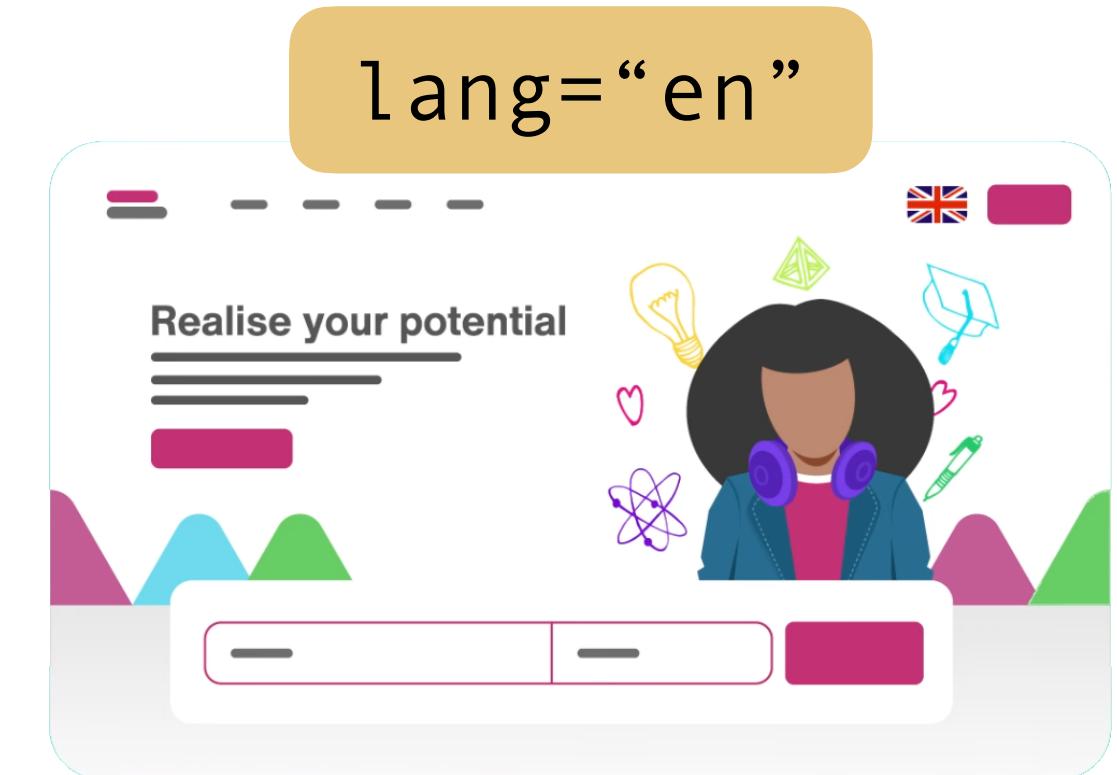
- E.g., identification of multi-lingual sections, glossaries of acronyms and unusual terms, plain language

- **3.2 Predictable**

- E.g., consistent content across pages, users can easily find help

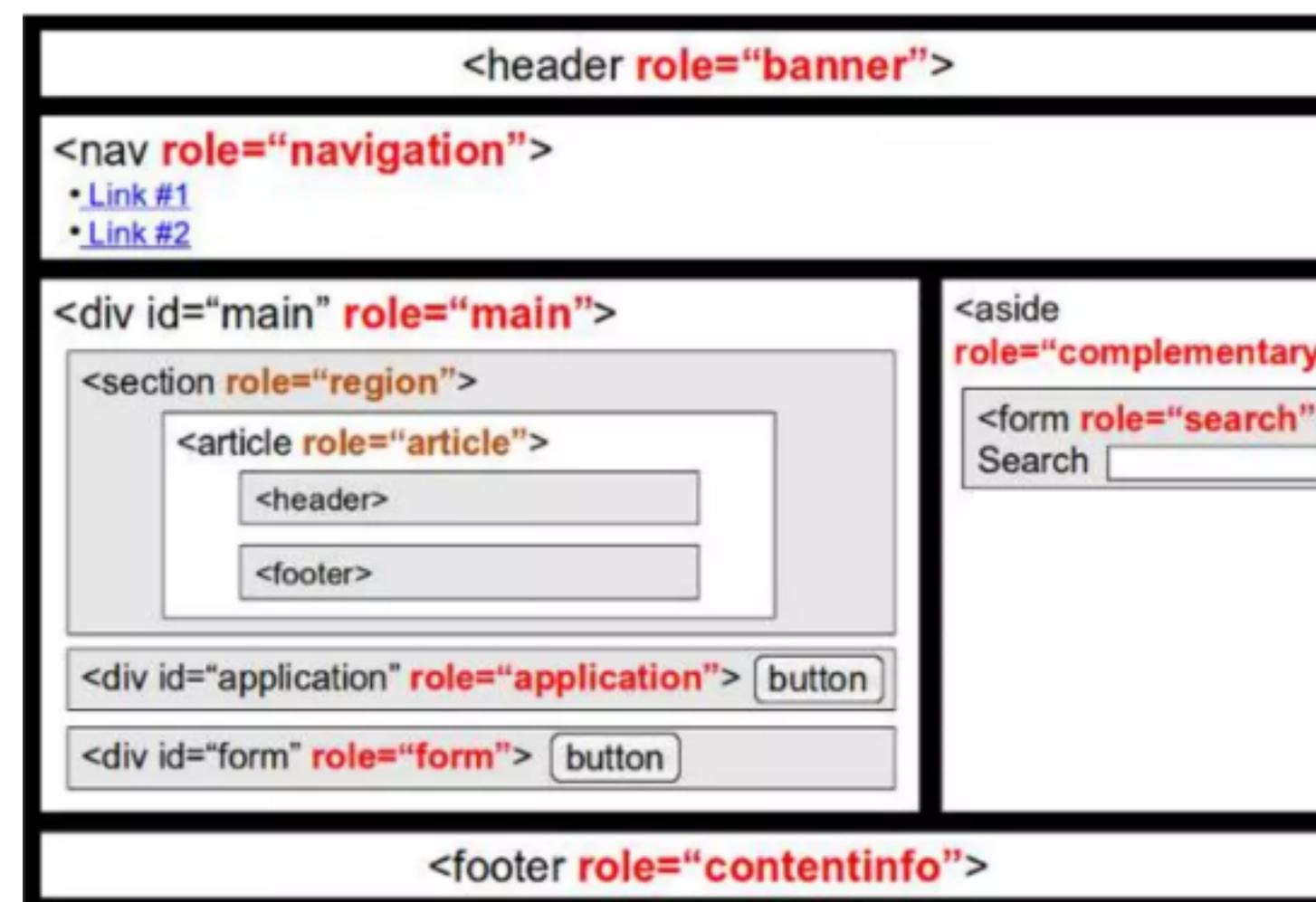
- **3.3 Input Assistance**

- E.g., helping users avoid and correct mistakes, intuitive error messages



# WCAG Principle #4: Robust

- Content must be robust enough for reliable interpretation by various user agents
- **4.1 Compatible**
  - E.g., semantic HTML = accessibility
  - E.g., using clean, standards-compliant HTML and CSS
  - E.g., ensuring compatibility with assistive technologies (e.g. ARIA)



HTML5 Element	ARIA Role
<article>	"article"
<aside>	"complementary"
<footer>	"contentinfo"
<header>	"banner"
<nav>	"navigation"
<section>	"region"
<main>	"main"
	"search"

# WAI-ARIA

- 💡 WCAG is technology-agnostic
- WAI-ARIA = Web Accessibility Initiative - Accessible Rich Internet Applications
  - Standard W3C, Latest Version [WAI-ARIA 1.2](#)
  - Provides additional context for assistive technologies as HTML attributes

```
<div id="progress-bar" role="progressbar"
      aria-valuenow="50" aria-valuemin="0"
      aria-valuemax="100" style="width: 50%;">
  50%
</div>
```

# WAI-ARIA

- **Roles**
  - Describe the **structure** of the Web page (headings, regions, ...)
  - Describe the type of **widget** presented (menu, treeitem, slider, progressbar)
  - More roles beyond structure: abstract, landmark, live region, window
- **States** (`aria-* attributes`): describe **characteristics** that may change in response to user action or automated processes (e.g., `aria-checked` for elements beyond checkboxes)
- **Properties** (`aria-* attributes`): describe the **nature** of a given object, or a data value associated with the object.(e.g., `aria-name`)
- Support for **keyboard navigation** for Web objects and events

# WAI-ARIA

- ARIA roles and attributes define, for each element:
  - **Role:** what type of "thing" is this? (e.g. button, main content)
  - **State:** in which situation is it? (e.g. disabled, checked, expanded)
  - **Properties:** e.g., under which name/label?
  - **Relationships:** is it tied to any other elements in the page somehow?
- This information is mapped by the browser to the operating system's accessibility API and exposed to assistive technologies

```
<div role="menu" aria-labelledby="menu-label">
  <h2 id="menu-label">Options</h2>
  <button role="menuitem" aria-disabled="true">Save</button>
  <button role="menuitemcheckbox" aria-checked="false" aria-labelledby="sort-label">
    <span id="sort-label">Sort by Date</span>
  </button>
  <button role="menuitem" aria-haspopup="true" aria-expanded="false" aria-controls="submenu">More Options</button>
</div>
<ul id="submenu" role="menu" hidden>
  <li role="menuitem">Option 1</li>
  <li role="menuitem">Option 2</li>
</ul>
```

# The 5 rules of ARIA

- W3C WAI-ARIA Authoring Practices Guide
- According to W3C Using ARIA

## 1. Don't use ARIA, use native HTML instead

- HTML5 semantic markup (e.g., `<header>`, `<nav>`, `<main>`) inherently provides accessibility support
  - ✓ `<button>Place Order</button>`
  - ✗ `<div role="button">Place Order</div>`
- Exceptions for new custom elements or new semantics

# The 5 rules of ARIA

- According to [W3C Using ARIA](#)

## 2. Do not change native semantics, unless you really, really have to

- Do not override the default semantics of HTML elements with ARIA roles, states, or properties



```
<div role=tab><h2>heading tab</h2></div>
```



```
<h2 role=tab>heading tab</h2>
```

- Bad ARIA is worse than no ARIA

# The 5 rules of ARIA

- According to [W3C Using ARIA](#)

## 3. All interactive ARIA controls must be usable with a keyboard

- ARIA provides semantics, but does not define keyboard control
- Developer must ensure that all ARIA elements are controllable with the keyboard
- E.g., Custom button shall work like a native button



```
<div role="button" tabindex="0">Place Order</div>
```

- Change focus with Tab key
- Activate the functionality by using both Enter and Space keys

# Keyboard Navigation

- Ensure all interactive elements are focusable
  - Remember: some HTML elements are inherently focusable (<a>, <button>, <input>, ...)
  - Remember: we can make non-focusable elements focusable

```
<div tabindex="0">This div is now focusable</div>
```

- Provide visible focus indicators

```
:focus { outline: 2px solid blue; outline-offset: 4px; }  
:focus-visible { outline: 2px solid blue; outline-offset:  
4px; }
```

- Avoid keyboard traps: ensure navigation into and out of components

# The 5 rules of ARIA

- According to [W3C Using ARIA](#)
- 4. **Do not use role="presentation" or aria-hidden="true" on a focusable element**

- That is, don't "neutralise" focusable elements



<button role=presentation>press me</button>



<button aria-hidden="true">press me</button>



<button role="presentation" tabindex="-1">Don't Click Me</button>



<button style="display: none;">Don't Click Me</button>

# The 5 rules of ARIA

- According to [W3C Using ARIA](#)

## 5. All interactive elements must have an accessible name

- Interactive elements, such as buttons and links, must have a name that can be programmatically determined and spoken by screen readers



First name<input type="text">



<input type="text" aria-label="First Name">



<label for="fname">First name</label> <input type="text" id="fname">

- Helps users understand the purpose of the element

# WCAG Conformance Levels

- WCAG: 86 success criteria at multiple levels (A, AA, AAA)
- How to Meet WCAG (Quick Reference)

**A**

Must have

This is the most basic level of compliance in WCAG.

**AA**

Should have

This is the most desired level of guidelines having good readable text along with good contrast ratio and support for interface zoom.

**AAA**

Good to have

This is the highest level of accessibility requires sign language for audio content which is a bit difficult in all cases to put in place.

# Some WCAG Best Practices

- **Text Alternatives:** Helps screen readers convey content
  - Provide descriptive text for images using the alt attribute
    - Decorative images  
`<img alt="" role="presentation" ... />`
    - Informative images  
`<img alt="A chart showing sales growth over time" ... />`
  - Provide descriptive text for input elements  
`<button aria-label="Close dialog">X</button>`
- **Colour Contrast:** Ensure text is readable for users with visual impairments
  - Check contrast ratios (e.g., WCAG Contrast Checker)
  - Aim for a minimum contrast ratio of 4.5:1 for normal text

WCAG2 Level A

WCAG2 Level AA

# Some WCAG Best Practices

- **Form Accessibility**

- Use <label> elements to associate text with form controls
- Provide clear instructions and error messages
- Use ARIA attributes to announce errors dynamically
  - `aria-live="assertive"` announces the error immediately, interrupting the current screen reader output
  - `aria-live="polite"` waits until the current screen reader output is finished before announcing the error

WCAG2 Level A/AA

```
<form id="exampleForm">
  <label
    for="username">Username:</
  label>
  <input type="text"
    id="username"
    name="username" />
  <button
    type="submit">Submit</
  button>
</form>
<!-- Error message
  container --&gt;
&lt;div id="error-message"
  class="error" aria-
  live="assertive"&gt;&lt;/div&gt;</pre>
```

# Some WCAG Best Practices

- **Responsive Design:** Ensure accessibility across different devices and screen sizes
  - Use flexible layouts and responsive units (e.g., percentages, ems)
  - Test on various devices and screen readers
- **Multimedia Accessibility**
  - Captions and Transcripts:
    - Provide captions for video content
    - Offer transcripts for audio content
  - Audio Descriptions:
    - Describe visual information for users with visual impairments

WCAG2 Level AA

WCAG2 Level A/AA

# WCAG Compliance

1. Check HTML through W3C validators
2. Use a tool for automated accessibility testing

-  WAVE browser plugin
-  Lighthouse browser plugin
-  Pa11y command-line tool

3. User testing?
  - E.g. HTML tables for layout make narration difficult to understand



The following apply to the entire page:

Styles: OFF  ON

Current URL: /panel

Details

Errors: 0 | Contrast Errors: 1 | Alerts: 1

Features: 1 | Structure: 4 | ARIA: 1

AIM Score: 9.1 out of 10

1 Contrast Errors  
1 Very low contrast

1 Alerts  
 1 Missing first level heading

Adicionar Apagar

## h2 Registrar Eventos

Assistências: 0  Incrementar  Focar

Remates: 0  Incrementar  Focar

Golos: 0  Incrementar  Focar

Recuperações: 0  Incrementar  Focar

Total: 0  Reset



# Lighthouse

Current URL: /panel

Painel

Adicionar

Apagar

## Registrar Eventos

Assistências: 0

Incrementar

Focar

Remates: 0

Incrementar

Focar

Golos: 0

Incrementar

Focar

Recuperações: 0

Incrementar

Focar

Total: 0

The screenshot shows the Lighthouse panel interface. At the top, there are four buttons: 'Painel' (highlighted in green), 'Adicionar', and 'Apagar'. Below this is a section titled 'Registrar Eventos' with four rows for tracking assists, shots, goals, and recoveries, each with 'Incrementar' and 'Focar' buttons. A 'Total: 0' summary is shown. The main area is a dark-themed browser window with the following details:

- Toolbar: Elements, Console, Recorder, Sources, Network, Performance, Memory, Application, **Lighthouse** (selected), Adblock Plus.
- Address bar: http://localhost:5173/panel
- Performance scores: 68, 94, 100, 82.
- Accessibility audit:
  - A red warning icon with the text: ▲ Background and foreground colors do not have a sufficient contrast ratio.
  - A note: These are opportunities to improve the legibility of your content.
  - An ARIA section with the note: ○ ARIA IDs are unique.



# Pally

```
● ● ● vue-counters-pinia-2 — -zsh — 116x32
[hpacheco@hpacheco vue-counters-pinia-2 % pally http://localhost:5173/
Welcome to Pally
> Running Pally on URL http://localhost:5173/
Results for URL: http://localhost:5173/panel

• Error: This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 2.78:1. Recommendation: change background to #258829.
└── WCAG2AA.Principle1.Guideline1_4.1_4_3.G18.Fail
    ├── #top > nav > a:nth-child(1)
    └── <a href="/panel" class="router-link-active router-link-exact-active" aria-current="page">Panel</a>

• Error: Duplicate id attribute value "abutton" found on the web page.
└── WCAG2AA.Principle4.Guideline4_1.4_1_1.F77
    ├── #abutton
    └── <button data-v-d5fae8ca="" id="abutton">Incrementar</button>

• Error: Duplicate id attribute value "abutton" found on the web page.
└── WCAG2AA.Principle4.Guideline4_1.4_1_1.F77
    ├── #abutton
    └── <button data-v-d5fae8ca="" id="abutton">Incrementar</button>

• Error: Duplicate id attribute value "abutton" found on the web page.
└── WCAG2AA.Principle4.Guideline4_1.4_1_1.F77
    ├── #abutton
    └── <button data-v-d5fae8ca="" id="abutton">Incrementar</button>

4 Errors
hpacheco@hpacheco vue-counters-pinia-2 %
```