

# Accessible Rich Interface Applications (ARIA)

felipe@ift3225 Hiver 2020




## Using VoiceOver to Evaluate Web Accessibility

[Home](#) > [Articles](#) > Using VoiceOver to Evaluate Web Accessibility

### Article Contents

- [Introduction](#)
- [Getting Started](#)
- [Reading](#)
- [Navigation](#)
  - [Other useful shortcut keys](#)
- [Images](#)
- [Data Tables](#)
- [Forms](#)
- [Practice](#)
- [VoiceOver on the iPhone, iPod Touch, and iPad](#)

### Introduction

[VoiceOver](#)  is a screen reader program that comes on new Mac computers, iPhones, iPads, and iPod touches. This article is designed to help users who are new to VoiceOver learn the basic controls for testing web content, and to serve as a reference for the occasional VoiceOver user.

Last updated: Jun 19, 2017

#### Translations

-  [German](#) by Anatoli Bauer
-  [Spanish](#) by José Manuel Cámara Iglesias

#### Related Resources

- [Three things you should know before using VoiceOver for testing](#)
- [Using JAWS to Evaluate Web Accessibility](#)
- [Using NVDA to Evaluate Web Accessibility](#)
- [Testing with Screen Readers: Questions and Answers](#)
- [Quick Reference: Testing Web Content for Accessibility](#)

# Faits

## Errors and Error Density

Errors are accessibility issues that are automatically detectable via WAVE, have notable end user impact, and are likely WCAG 2 conformance failures. **59,653,607 distinct accessibility errors were detected across the 1 million home pages—an average of 59.6 errors per page.**

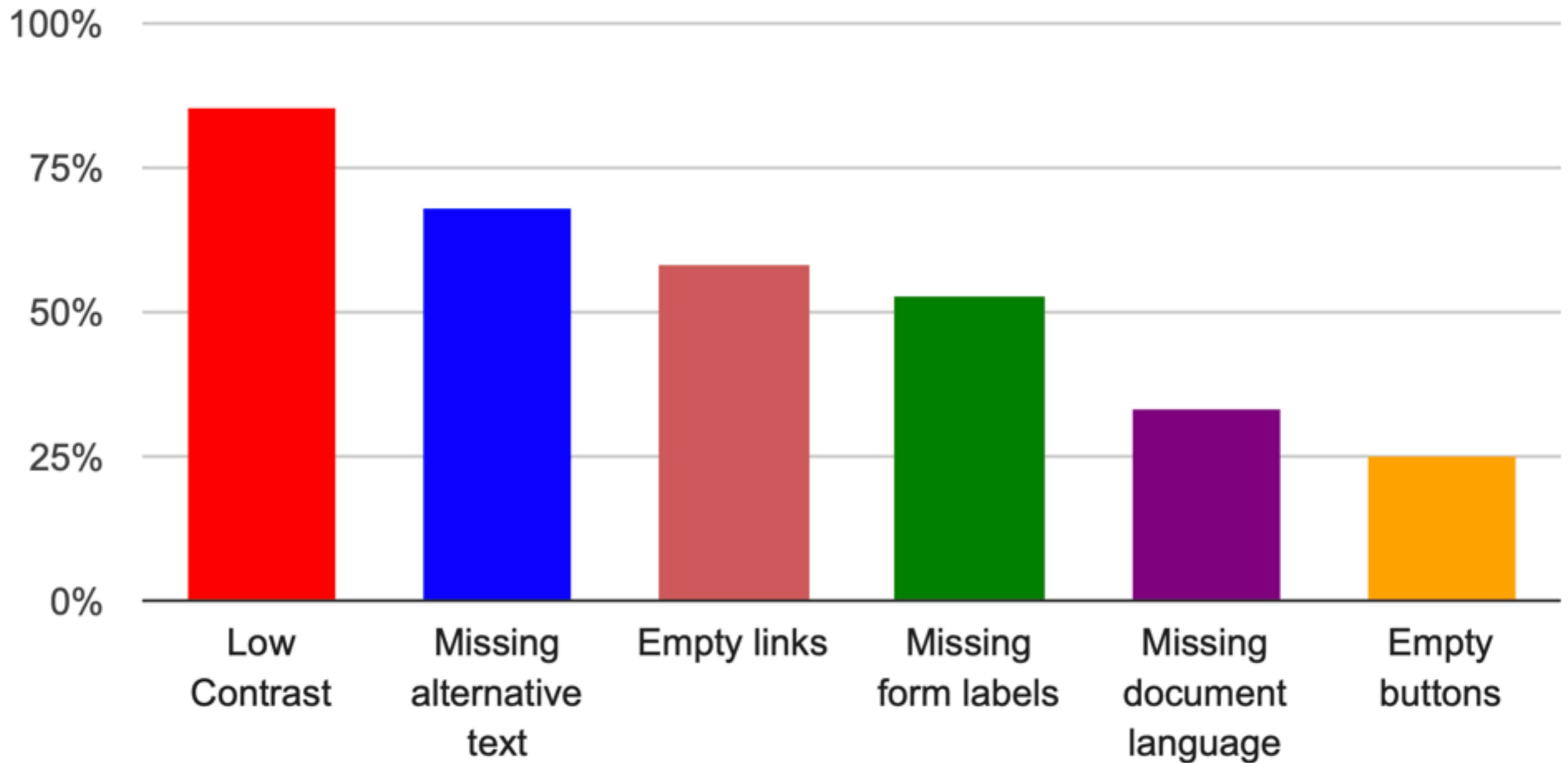
Error density (number of errors divided by number of page elements) for all home pages was collected. 782,481,056 distinct HTML elements were analyzed, meaning there was an average of 782 elements per home page. This results in approximately 7.6% of all home page elements having a detectable accessibility error. **Users with disabilities would expect to encounter detectable errors on 1 in every 13 elements with which they engage.**

## WCAG Conformance

**97.8% of home pages had detectable [WCAG 2](#) failures!** These are only automatically detectable errors that align with WCAG conformance failures with a high level of reliability. Because automatically detectable errors constitute a small portion of all possible WCAG failures, this means that the actual WCAG 2 A/AA conformance level for the home pages for the most commonly accessed web sites is very low, perhaps below 1%.

**Étude menée par WebAIM en février 2019 sur 1M de home pages**

## Most common WCAG failures (% of home pages)



### ARIA

**60.1% of the 1 million home pages had ARIA present.**

## Fun Facts

---

Here are several other fun facts regarding this research:

- The WebAIM Million database has 168,000,000 data points.
- It took 66.2 days of cumulative computer processing time to download and process all 1,000,000 home pages in the sample. This was shared among 5 AWS instances that ran continuously for 5 days.
- Despite being 2019, 11,200 home pages had `<marquee>` and 570 home pages had blinking content (`<blink>` or CSS `text-decoration:blink`).
- 2,099,665 layout tables were detected compared to only 113,737 data tables.
- The most errors detected on a single home page was 26,680!

**Cette étude contient pleins d'autres statistiques intéressantes**

# WAI-ARIA

**Web Accessibility Initiative - Accessible Rich Internet Applications** est une spécification W3C qui définit des **attributs** améliorant la sémantique et l'accessibilité d'une page. Il y a trois types de *features*:

- Rôles  
Indique ce qu'un élément fait (ex: role=« navigation »)
- Propriétés  
Indique une propriété d'un élément (ex: aria-required=true)
- État  
Indique une condition particulière d'un élément (ex: aria-disable=true)

# Sémantique

```
<header>
  <h1>...</h1>
  <nav role="navigation">
    <ul>...</ul>
    <form role="search">
      <!-- search form -->
    </form>
  </nav>
</header>
```

Particulièrement utile pour des éléments (ex: <div>) dont la sémantique n'est pas portée par le nom du tag

crédits

```
<main>
  <article role="article">...</article>
  <aside role="complementary">...</aside>
</main>
```

```
<footer>...</footer>
```

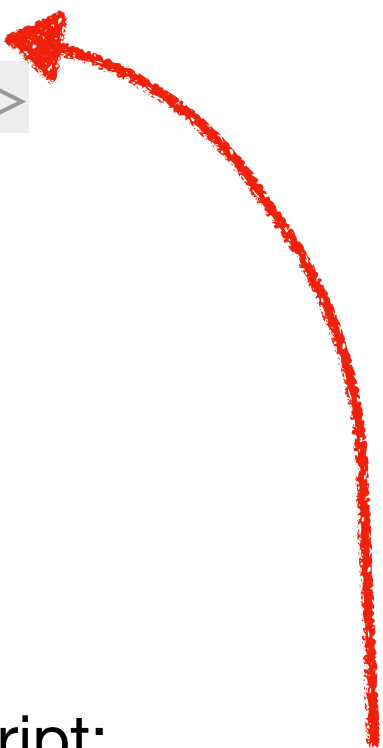
```
<input type="search" name="q" placeholder="Search query"
aria-label="Search through site content">
```

lu lorsque l'élément est sélectionné

crédits

# Changements dynamiques

```
<section arial-live"assertive" aria-atomic="true">  
  <h1>Random quote</h1>  
  <blockquote>  
    <p></p>  
  </blockquote>  
</section>
```



- **arial-live: off** défaut (ne rien annoncer)
- **arial-live: polite** annonce seulement si l'utilisateur ne fait rien
- **arial-live: assertive** (annonce dès que possible)
- **aria-atomic="true"**: indique que tout le contenu doit être annoncé (pas juste la partie modifiée)

avec en javascript:

```
window.setInterval(showQuote, 10000);
```



**Quelque chose ne devrait pas être lu ?**

```
aria-hidden="true"
```

**+d'ARIA**

**Voir aussi les bonnes règles d'un codage HTML sémantique**