

ACCESSIBLE CODE PATTERNS

francesco bedussi

inspired by

Inclusive design pattern by Heydon Pickering

*unfortunately, we have a habit of
overengineering and overcomplicating
things*

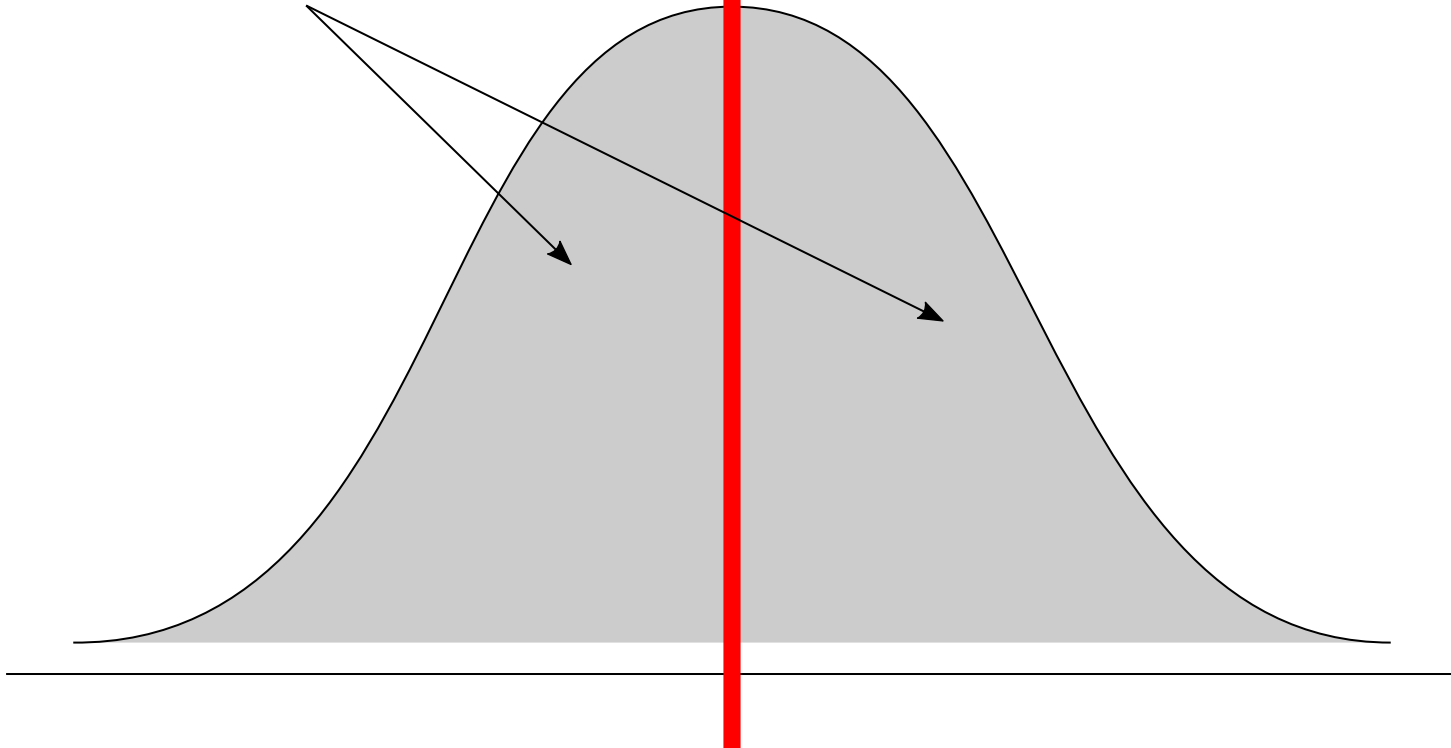
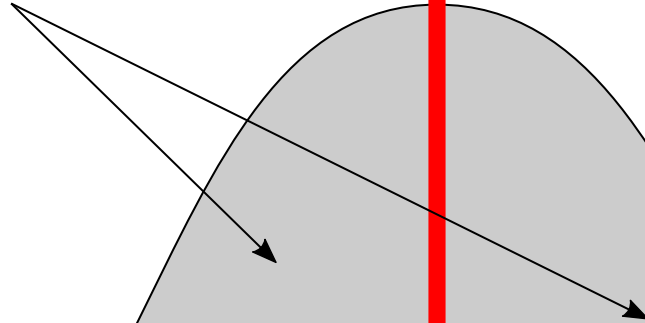
*targeting a so-called average user is a
disastrous interface design strategy,
because average users do not exist*

Heydon Pickering

average user



all the rest



ACCESSIBILITY, WHY?

- It involves **a lot of people**: Though estimates vary, most studies find that about one fifth (20%) of the population has some kind of disability
- Search engine spiders are severely "impaired", so an accessible site is also a **search engine friendly site**
- **We have to**. Governments (especially USA) are enforcing accessibility by legislation

ACCESSIBILITY, FOR WHOM?

- Vision impairments
- Limited fine motor control
- Cognitive disabilities (rings a bell? :-P)
- Hearing impairments
- Older people (ourselves tomorrow)

CHECKLIST

- People who don't use a mouse should be able to use a site
- People who don't look at a screen should be able to use a site
- A site's content should be visually legible
- People should have control over automatic changes to the page

PATTERNS VS PRINCIPLES

INDEX

- document
- navigation
- (menu) button
- page
- paragraph
- list of products
- filter widget
- registration form

DOCUMENT

THE LANG ATTRIBUTE

- on `html` element, on any inner element
- `:lang(en)` pseudo-class

ALLOW PINCH-TO-ZOOM

DON'T

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0, minimum-scale=1.0,  
maximum-scale=1.0, user-scalable=no">
```

DO

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0">
```

FONT SIZES

do

```
html {  
  font-size: 100%;  
}
```

don't

```
html {  
  font-size: 16px;  
}
```

responsive font-size

```
html { font-size: calc(1em + 1vw); }
```

Font loading

- prefer FOUT over FOIT
- load web fonts asynchronously

- use proper title element
- use the main element

BONUS: PRINT ONLY THE MAIN ELEMENT

```
@media print {  
  body > *:not(main) {  
    display: none;  
  }  
}
```

SKIP TO MAIN CONTENT

- for sighted keyboard users

```
[href="#main"] {  
  position: absolute;  
  top: 0;  
  right: 100%; /* moves off screen */  
}  
  
[href="#main"]:focus {  
  right: auto;  
}
```

NAVIGATION

NAVIGATION

- a menu should look like a menu



- a menu is a nav containing an ul
- identify the current page link...
- ...but not with color alone

A NAVIGATION EXAMPLE

```
<header role="banner">
  <a href="#main">
    
  </a>
  <nav>
    <ul>
      <li><a href="#main">home</a></li>
      <li><a href="/about">about</a></li>
      <li><a href="/products">products</a></li>
    </ul>
  </nav>
</header>
```

TABLE OF CONTENTS

```
<nav class="toc" aria-labelledby="contents-heading">
  <h2 id="contents-heading">Contents</h2>
  <ul>
    <li><a href="#history">Our history</a></li>
    <li><a href="#services">The services we offer</a></li>
    <li><a href="#office">Visit our office</a></li>
  </ul>
</nav>
```

if you hijack the link to implement smooth scroll don't
forget to manage the focus

(MENU) BUTTON

(MENU) BUTTON

- (hamburger) icon + label + button appearance
- min touch target size: 48x48px

```
<nav aria-label="site">
  <button aria-expanded="false">
    <svg><use xlink:href="#navicon"></use></svg>
    menu
  </button>
  <ul hidden>
    <li><a href="#main">home</a></li>
    <li><a href="/about">about</a></li>
    <li><a href="/products">products</a></li>
  </ul>
</nav>
```


critical JS (vanilla, to be included at the end of the page)

```
(function() {  
    var button = document.querySelector('[aria-label="site"] b  
    var menu = button.nextElementSibling;  
  
    button.setAttribute('aria-expanded', 'false');  
    button.hidden = false;  
    menu.hidden = true;  
  
    button.addEventListener('click', function() {  
        var expanded = this.getAttribute('aria-expanded') ===  
        this.setAttribute('aria-expanded', String(!expanded));  
        menu.hidden = expanded;  
    });  
})();
```

PAGE

HEADINGS

- describe nested sections
- one h1 per page
- never skip a level
- never use a heading for a subtitle

VIDEO

- use captions: non-native speakers, for people watching video with audio turned off (e.g. public places), for deaf and hard of hearing
- use keyboard and screen reader accessible players (e.g. YouTube, it provides captions too)
- provide a transcript (linearized version of captions)

VERTICAL FLOW

Use paragraph line-height as basis for vertical spacing:
e.g if the line-height is 1.5, then one unit of vertical
whitespace should be 1.5rem:

```
main * + * {  
  margin-top: 1.5rem;  
}
```

note the use of the owl selector

.VISUALLYHIDDEN

```
.visuallyHidden {  
  position: absolute;  
  width: 1px;  
  height: 1px;  
  overflow: hidden;  
  clip: rect(1px, 1px, 1px, 1px);  
}
```

TRICKS

DEFENSIVE CODING

```
main :empty {  
  display: none;  
}
```

FORCEFEED.JS

allows you to feed the layout with random content of different length

PARAGRAPH

PARAGRAPH

- the ideal length of a line is around 60 characters

```
main {max-width: 60rem;}
```

- no justification
- line height ~1.5

```
p { line-height: 1.5; /*unitless!*/ }
```

- avoid extremely high or extremely low contrast
- keep the standard link style (blue, underlined)
whenever possible, eventually improve it
- never ever just remove focus style, eventually replace it

LIST OF PRODUCTS

LIST OF PRODUCTS

- code products list as ``
- use self-governing grid

```
.grid {display: flex; flex-wrap: wrap;}  
.grid li {flex-grow: 1; flex-shrink: 1; flex-basis: 10em}
```


KEY INFORMATION

group key pieces of information in a definition list:

```
<dl>  
  <dt>Size:</dt>  
  <dd>90cm x 30xm</dd>  
  <dt>Price:</dt>  
  <dd>€35.95</dd>  
</dl>
```

IMAGES

- provide alt text only if it adds meaning
- optimize & lazy load images

CTA

- if the cta is a link do not style it as a button...
- ...but use some visual clue (e.g. color) common to links, buttons, ctas to indicate their interactivity
- if cta is a link style it using tag + class selector, e.g. `a.call-to-action`
- buttons do not have an hand cursor
- include the product name, eventually visually hidden, in the cta label

AVOID LINK WRAPPING PRODUCT BLOCK

- it doesn't have a dedicated label
- produces unexpected behavior in some screen readers
- on touch device it can be pressed accidentally

USE MICRODATA TO ENHANCE SERP

```
<main id="main" itemscope itemtype="http://schema.org/Product">
  <h1>
    <span itemprop="name">Naked man in garage forecourt</span>
    <a href="/artist/kenny-mulbarton">by Kenny Mulbarton</a>
  </h1>
  
  <dl>
    <dt>Size: </dt>
    <dd>90cm &times; 30cm</dd>
    <dt>Price: </dt>
    <dd>
      <span itemprop="offers" itemscope itemtype="http://schema.org/Offer">
        <meta itemprop="priceCurrency" content="EUR" />
        €<span itemprop="price">35.95</span>
      </span>
    </dd>
  </dl>
</main>
```

LOADING MORE RESULTS

- avoid infinite scroll
- prefer a "load more" button
- remember to manage focus
- and to communicate that the loading is in progress

1. click on "load more"
2. the button is disabled and its label changed in "loading"
3. a hidden live region announces "loading more products"
4. the request is handled
5. on success the content is rendered
6. the live region announces "products loaded"
7. focus is moved to the first of the new products
8. the "load more" button reverts to its original status

A FILTER WIDGET

A FILTER WIDGET

- leverage HTML behavior
- visually hide radio buttons and style labels based on radio buttons status
- use live regions to communicate that the content is being fetched
- do not remove the submit button

A REGISTRATION FORM

LOGIN/REGISTER TOOLBAR

```
<h1>Welcome</h1>
  <div role="toolbar" aria-label="login or register">
    <button aria-pressed="true">Login</button>
    <button aria-pressed="false">Register</button>
  </div>
  <div id="forms">
    <div id="login">
      <form>
        <!-- login form -->
      </form>
    </div>

    <div id="register">
      <form>
        <!-- registration form -->
      </form>
    </div>
  </div>
</div>
```


- USE LABELS!!!
- placeholders are not labels, they are hints on how to fill in the field

- `<fieldset>` are pointless without `<legend>`
- use `aria-required="true"` and prefer it over `required` html attribute
- give the possibility to show password
- mark error message containers as `aria-live="assertive"`
- use `aria-invalid="true"` on field that do not pass the validation (it can be used as a CSS selector as well)
- provide a hint on why the field is invalid