

What is accessibility and why I should care?

Some common pitfalls and a few tips for success

Sandra Eriksson

Certified Accessibility Expert

sandra.m.eriksson@gmail.com

@sandrability

Usability, accessibility - or both?

Usability

*“The effectiveness, efficiency and satisfaction
with which specified users achieve specified goals in
particular environments.”*

ISO 9241

Accessibility

*“Aspects related to user experience
for people with disabilities.”*

W3C

Related but not to be mixed up. Neither should be excluded.

Whereas usability considers how easy a site is to use and understand, accessibility is concerned with whether you can get there at all.

*“Solutions designed for users
with the biggest difficulties,
will work for everyone.”*

People with disabilities - who are they?

15% with a disability

The world's biggest minority

Permanent or temporary

Over a billion of the world's population.

Temporarily disabilities: accidents, health issues, age, everyday life episodes



Who do you know with a disability?

Disabilities

Vision

Hearing

Physical

Cognitive & Neurologic

Vision: Blindness, low vision and colour blindness

Accessibility always been present

Born with an eye condition

Accessibility & Assistive technology always been present

Hearing: deaf and the hard of hearing, consider themselves as a minority, not as group of people with a disability.

Sign language is for deaf people their first language.

Physical: Impaired mobility and speech

Cognitive & Neurological: Dyslexia, attention or memory difficulties, epilepsy, autism, intellectual limitations

What is a screen reader?

Assistive software

Reads content on screen

Combines different features

Sophisticated or primitive

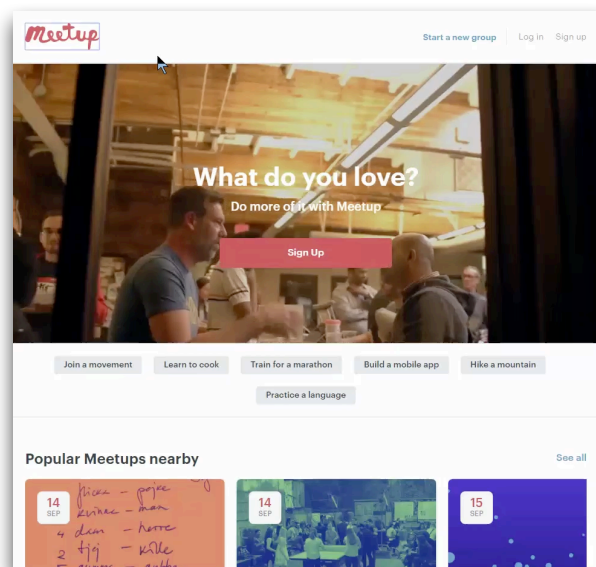
Assistive software: Visual impairment and dyslexia

Combines different features: Speech, Braille Magnifying

Sophisticated or primitive: controlled solely with keyboard or click-and-read features

Screen reader examples: NVDA (open source, Jaws 40 minutes demo, Voice Over for Apple Products and Google Talkback for Android)

A screen reader in action!



Buttons with no labels

English

—

```
<li class="lp-has-children">
  <a href="/english/demo-environment-
programme.html">
    Demo Environment Programme
  </a>
  <button class="lp-toggler" aria-expanded="false">
  </button>
</li>
```

Cookies and personal data

+

Live Areas

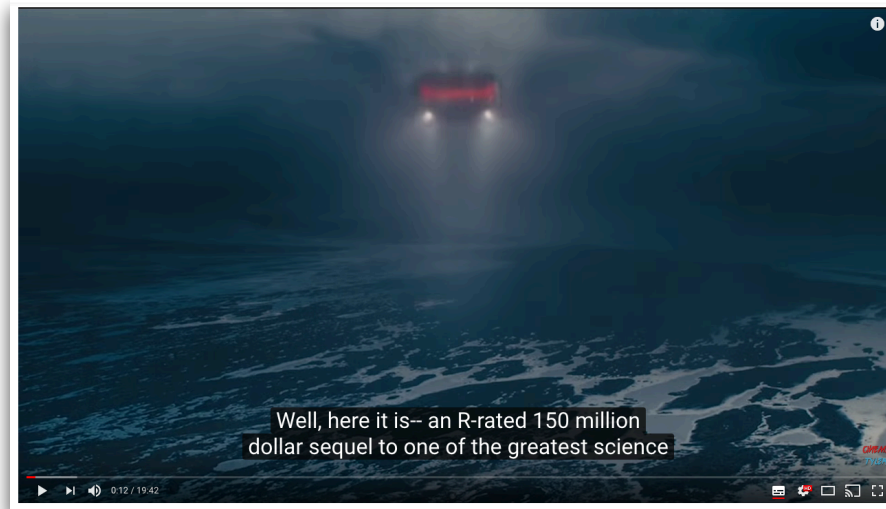
nothing yet.

Start

Stop

If not coded correctly, impossible to read.

Audio content with no subtitles





Video with no subtitles > problems for users with impaired hearing.
Severe when speaker isn't visible.

Unreachable objects



Departing From?
Enter a stop number, address, intersection
or landmark.

or
Select from Popular Locations ▼

Going To?
Enter a stop number, address, intersection,
or landmark.

or
Select from Popular Locations ▼

```
<td style="padding-left: 6px" valign="middle">
...
<div id="OriginHelp"></div>

</td>
```

Unreachable buttons > users navigating with keyboard.

Distractive content

February 2017 theguardian

MY BEST SELLING CAR LEASING DEALS!

JAZZ IT UP! **CZECH GOLF!** **MEGA-HATCH!**

Honda Jazz
1.3 i-VTEC SE 16v (102bhp)
Hatchback Sdr 1318cc Petrol
Manual (6 Speed)
RENT NEW £167/mth
including VAT
3 year cheap car leasing
Petrol engine
Manual gearbox
Flat paint
Drive 10,000 miles per year
(Other mileages available)

Skoda Octavia
1.0 TSI SE Technology 12v
(115bhp) (5 seats) Hatchback
Sdr 999cc Petrol Manual (6 Speed)
RENT NEW £185/mth
including VAT
3 year cheap car leasing
Petrol engine
Manual gearbox
Flat paint
Drive 10,000 miles per year
(Other mileages available)

Citroen C3 Aircross
1.2 PureTech Feel 12v
(82bhp) Hatchback Sdr
1199cc Petrol Manual (5 Speed)
RENT NEW £193/mth
including VAT
1 year cheap car leasing
Petrol engine
Manual gearbox
Metallic paint
Drive 10,000 miles per year
(Other mileages available)

PLAY QUIZ!
WIN A WAHI MUG
[CLICK HERE!](#)

ABARTH
124 Spider
Convertible
595
595 Convertible
695
695 Convertible

ALFA ROMEO
Giulia Saloon
Giulietta
Stelvio Estate

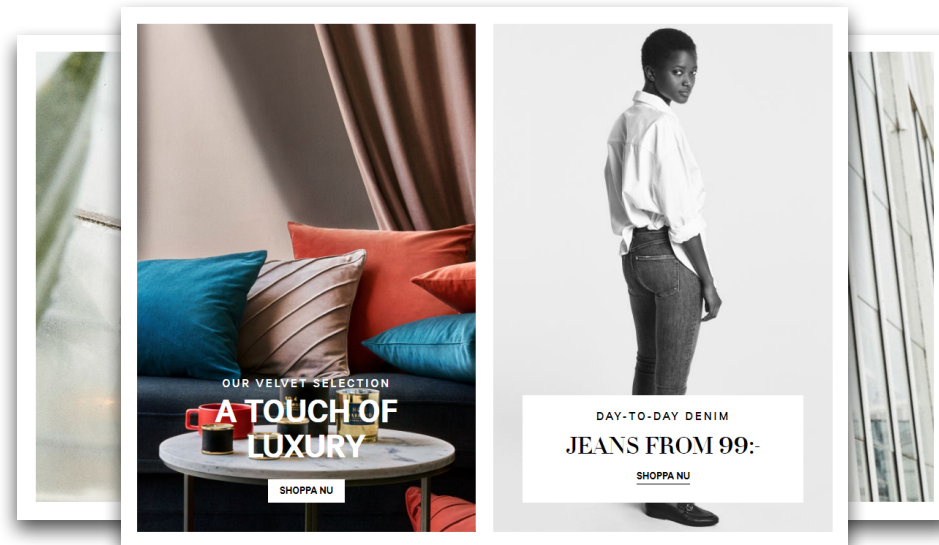
AUDI
A1 3dr Hatch
A1 Sportback
A3
A3 Saloon
A3 Sportback
A4 Allroad Estate
A4 Estate
A4 Saloon
A5 Convertible
A5 Coupe
A5 Sportback
A6 Allroad Estate
A6 Estate
A6 Saloon

SEOL SOUL! **SUNKISSED!** **SMOKIN'!**

MEET MY STAFF

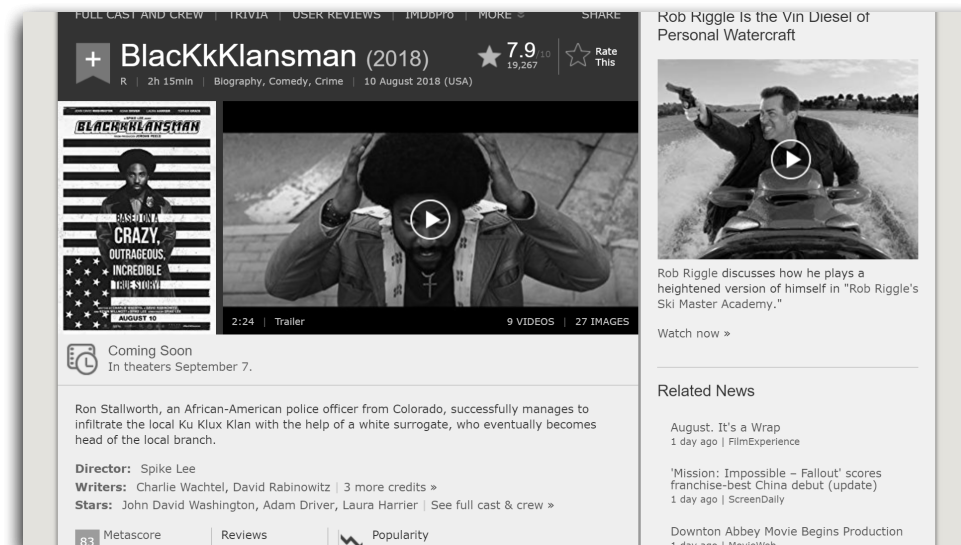
Video cannot be stopped. Blinking and jumping objects. Lots and lots of bright colors!

Low contrasts



Sometimes easy to fix contrasts in a fancy way!

Use of color



Links are highlighted with only color, nothing else. No colors > rare. Difficult for users with color blindness.

WCAG

Web Content Accessibility Guidelines

International standard

Developed by World Wide Consortium

23 september, the Web Accessibility Directive (websites, apps provided by public sector), based upon EN 301549 -> WCAG 2.0

Technical and complex

WCAG

4 Principles

13 Guidelines

78 Success Criteria

Level A, AA, and AAA

Wcag 2.0 is old, from 2008

Wcag 2.1 > Wcag 2.0 updated with new success criteria in 2018

To conform legislation -> A and AA levels are mandatory

Hopefully Web directive and EN-standard, embrace Wcag 2.1

Principle 1: Perceivable

Success criterion 1.1.1 (A): Non-text content

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose.

Exceptions:

- Decoration
- Captcha
- Time-based media

Images described with alt-texts, videos are described shortly in plain text.

Alt-text



Also describe the how the alt-attribute is used for images.
Empty alt-attributes are interpreted as decorative.

Principle 2: Operable

Success criterion 2.1.1 (A): Keyboard

All functionality of the content operable through a keyboard interface without requiring specific timings for individual keystrokes.

Exceptions:

- Functions requiring input depending on the path of the user's movement

Users navigating with a keyboard should be able to reach all object on a website

Exceptions:

Drawing, gaming...

Keyboard navigation



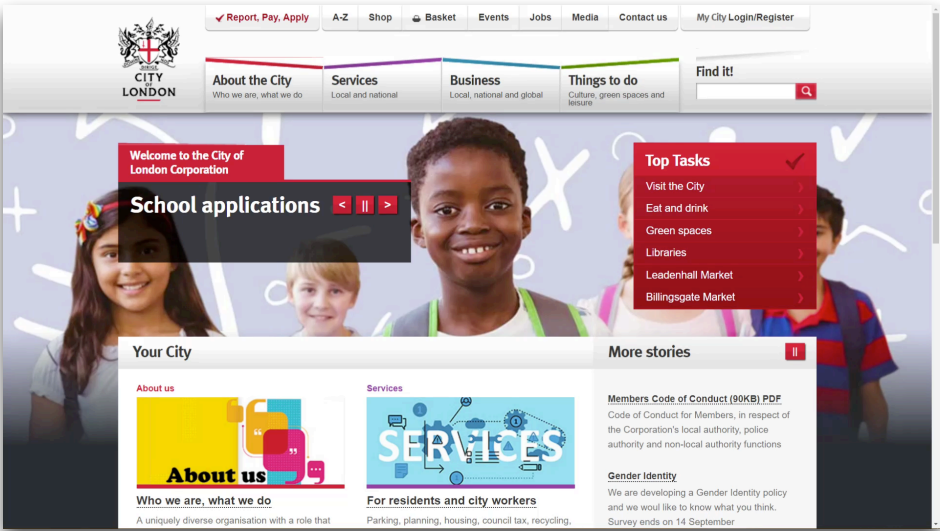
Principle 3: Understandable

Success criterion 3.2.1 (AA): Consistent Navigation

Navigational mechanisms that are repeated on multiple web pages with a set of web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user.

Areas that are repeated appear in the same order on a website.

Consistent Navigation



Principle 4: Robust

Success criterion 4.1.3 (A): Status Messages

In content implemented using markup languages, status messages can be programmatically determined through role or properties such that they can be presented to the user by assistive technology without receiving focus.

HTML controls meets this criterion when used according to documentation.

Status Messages

There's no internets. Go to the pub!

```
<div id="status" role="alert" aria-  
live="assertive" class="online">  
  <p>You are online.</p>  
</div>
```

Credit to Haydon Pickering, Practical ARIA examples

Add video clip showing how a screen reader announces the name, role and value of a native HTML control.

Accessible Coding™

Think “accessible” from the start

Pick a code standard and stick to it

WAI-ARIA?

Keep it simple!

Use HTML native objects, WAI-ARIA when needed

Test your code

Errors in code > assistive software misinterpret document structure

WAI-ARIA > dynamic content, roles, states and properties, varies among screen readers

Keep it simple > complex interfaces -> hard to understand and hard to code

Testing Accessibility

Test with users

Let go of the mouse

Try a screen reader - they don't bite!

Turn off your monitor

Funkify – Google Chrome addon

Automatic tools

Got stuck? Talk to an expert!

Screen reader: navigate with keyboard, listen to the output. Reads label, type, status, sometimes in a different ord. Try list tool for links, headings, controls.

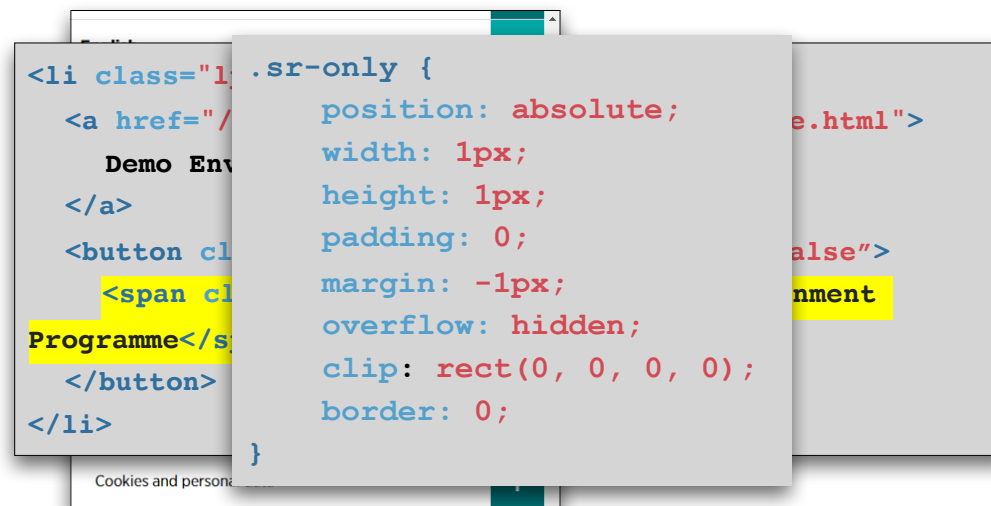
Download NVDA from nvaccess.org, function key 7 for list tool

Automatic tools > SiteImprove, W3C Validator, The Web Accessibility Toolbar



Tips & Tricks

Hidden labels on buttons



Buttons that lack a description. Do not use display: none!

Toggle

```
<p>
  <a class="btn btn-primary" data-toggle="collapse"
href="#collapseExample" role="button" aria-expanded="false" aria-
controls="collapseExample">
    Link with href
  </a>
  <button class="btn btn-primary" type="button" data-toggle="collapse"
data-target="#collapseExample" aria-expanded="false" aria-
controls="collapseExample">
    Button with data-target
  </button>
</p>
<div class="collapse" id="collapseExample">
  <div class="card card-body">
    Anim ...
  </div>
</div>
```

Two kinds of buttons with aria-expanded

Aria-controls are not necessary

Use on interactive objects!

What happens with no aria-expanded?

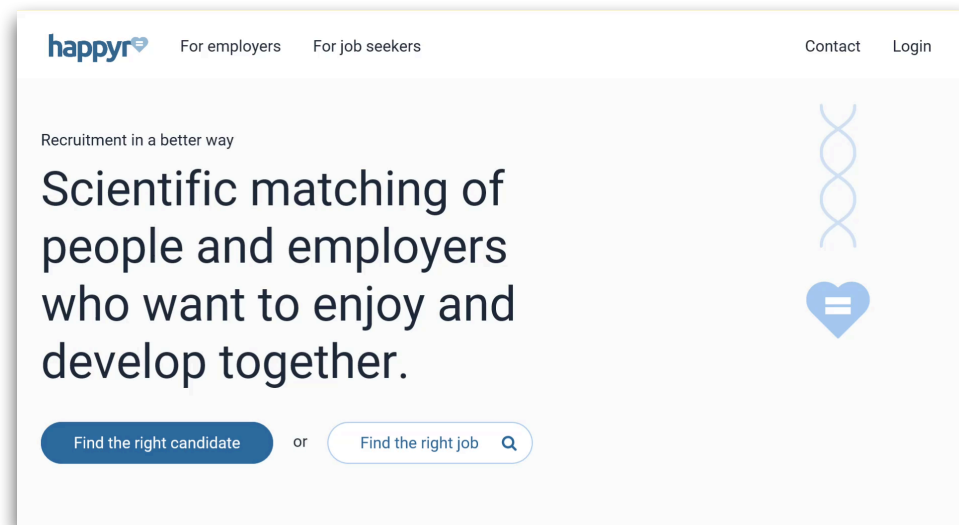
Dialog Boxes

```
<header aria-hidden="true">
...
</header>
<main>
  <div class="other-content" aria-hidden="true">
    ...
  </div>
  <div class="modal" tabindex="-1" role="dialog">
    ...
    <div class="modal-header">
      <h5 class="modal-title">Modal title</h5>
    ...
    <p>Modal body text goes here.</p>
    ...
  </div>
  ...
</div>
</main>
<footer aria-hidden="true">
  ...
</footer>
```

Semi modal windows, covering a lot of content

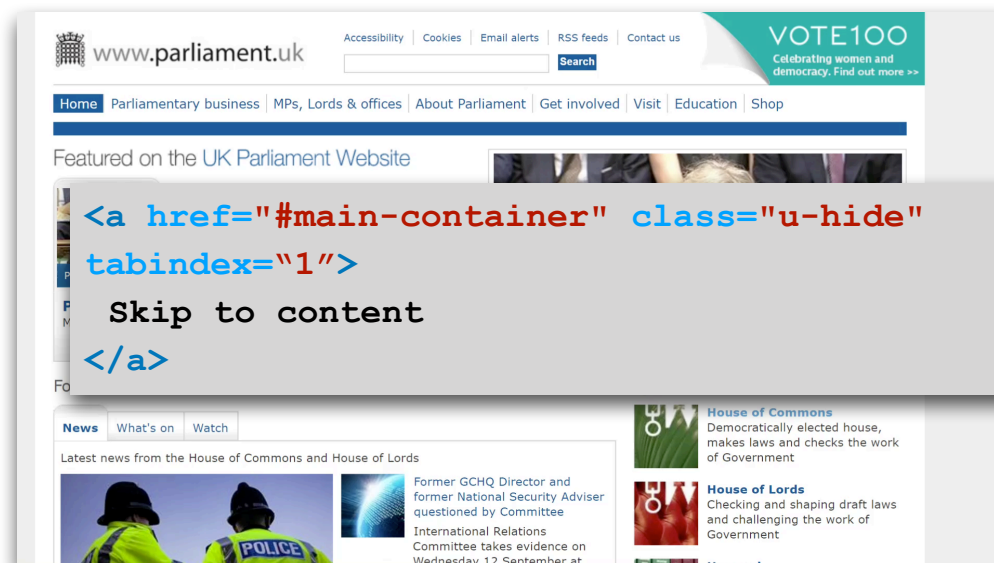
True Bootstrap modal window

Visible Focus



Example of clear and visible focus effects on Happyr
It doesn't have to be ugly!

Skip To Content



Skip to content link
Hidden, visible when in focus
Jumps to main content

The Times and The Sunday Times
<https://www.thetimes.co.uk/magazine/culture/a-wealth-of-talent-gemma-chan-on-crazy-rich-asians-2pmzhx6xr>

Symfony & Accessibility

Symfony helps with the forms, bootstrap with accessibility.

Standard Symfony DIV Theme

```
<form name="user" method="post">
  <div>
    <label>Transport</label>
    <ul>
      <li>This value should not be blank.</li>
    </ul>
    <div id="user_transport">
      <input type="radio" id="user_transport_0" name="user[transport]" value="car"/>
      <label for="user_transport_0">Car</label>
      <input type="radio" id="user_transport_1" name="user[transport]" value="bike"/>
      <label for="user_transport_1">Bike</label>
      <input type="radio" id="user_transport_2" name="user[transport]" value="other"/>
      <label for="user_transport_2">Other</label>
    </div>
  </div>
  ...
</form>
```

This is the output of the normal symfony theme (div) when there is ERROR

Tips: use no validate attribute to disable html5 error messages.

Include Bootstrap

```
{# templates/base.html.twig #}

{% block head_css %}
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/boot
{% endblock %}

{% block head_js %}
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" i
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/
{% endblock %}
```

First you need to include Bootstrap CSS and JS. See the docs!

Switch to Bootstrap

```
{% extends "base.html.twig" %}

{% form_theme form "bootstrap_4_layout.html.twig" %}

{% block body %}
    <h1>Here is my form:</h1>
    {{ form(form) }}
{% endblock %}
```

Use Bootstrap on any of your forms in Symfony.

Use “form_theme” to define what theme you like to use. Then just render the form as normal.

Using It Everywhere

```
# config/packages/twig.yaml
twig:
  form_themes: ['bootstrap_4_layout.html.twig']

{% extends "base.html.twig" %}
{% block body %}
  <h1>Here is my form:</h1>
  {{ form(form) }}
{% endblock %}
```

In config file

How to use on all forms on website.

Use it on all forms by specifying twig.form_themes in your config.

Perfect Bootstrap Radios

```
<fieldset class="form">
  <legend>Transportation</legend>
  <div class="form-group">
    <input type="text" class="form-control" />
  </div>
  <div class="form-group">
    <input type="radio" /> Car
    <input type="radio" /> Bike
    <input type="radio" /> Other
  </div>
</fieldset>
```

User form

Name

ERROR This value should not be blank.

Transport

ERROR This value should not be blank.

☐ Car

☐ Bike

☐ Other

Submit

This is an example what the Bootstrap theme does for you. It adds all bootstrap required classes and prints perfect HTML

No HTML5 Validation

```
<form name="user" method="post" novalidate="novalidate">  
...  
</form>
```

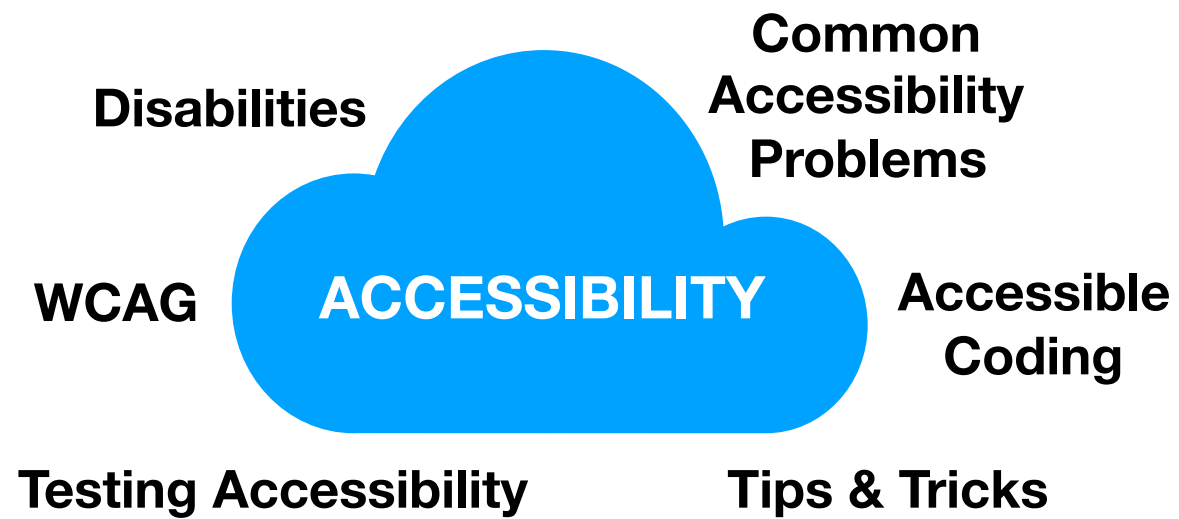
```
{% block body %}  
...  
  {{ form_start(form, {attr{novalidate:"novalidate"}}) }}  
...  
{% endblock %}
```

One step closer to accessible error messages, and accessibility generally.

*“Accessible interfaces
lead to better UX
and to more revenues.”*

The 15% mentioned earlier are your potential customers.

Summary



No more excuses!

*Thank you for listening
and enjoy the rest of the event!*

