Web Accessibility Training

Jared Smith, Jon Whiting, and John Northup

WebAIM.org

webaim.org/presentations/2019/pitt/

Minimizing Risk

- Ensure your content is accessible
- Have a policy and implementation plan in place
  - Samples at ncdae.org/goals
- Address automatically detectable errors

Auditory Disabilities

WCAG Requirements

Level A

- Captions
- Transcript for audio-only or video-only content
- Audio description OR transcript (if needed)

Level AA

- Audio description (if needed)
- Live audio – Captions

Level AAA

- Transcript
- Sign language
- Live audio – Transcript

Our recommendation

Level A

- Captions
- Transcript for audio-only or video-only content
- Audio description OR transcript (if needed)

Level AA

- **Transcript**
- Live audio – Captions

Level AAA

- **Audio description** (if needed)
- Sign language
- Live audio – Transcript

What are additional benefits of captions and transcripts?

Deaf-blind

- Content accessible to users who are blind **and** to users who are deaf will generally be accessible to users who are deaf-blind.
- Descriptive transcripts provide media accessibility.

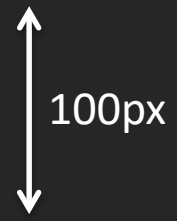
Low Vision

Browser Zoom

- Browser controls
  - Larger: Ctrl/command & +
  - Smaller: Ctrl/command & -
  - Reset : Ctrl/command & 0
- Responsive design supports users with low vision who zoom page content
- Zoomed content will trigger responsive breakpoints
- Some users may adjust only text sizes
  - Test in Firefox

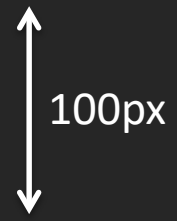
Avoid CSS height

I am some text in a div that has a pixel height



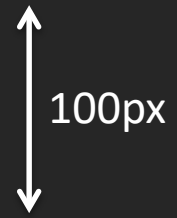
Avoid CSS height

I am some text in a div that has a pixel height



CSS min-height

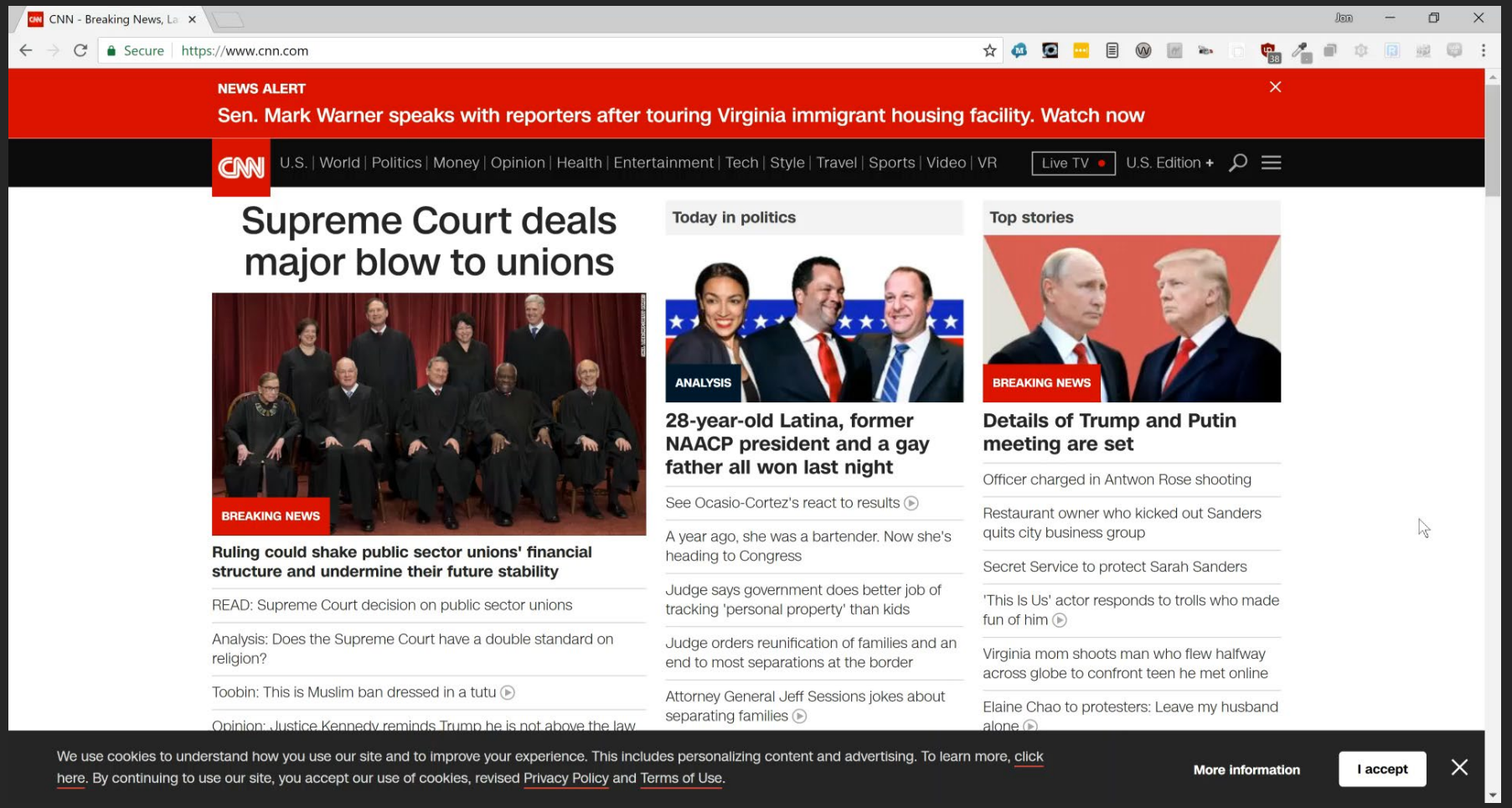
I am some text in a div that has a pixel min-height



WCAG Zoom Requirements

- 1.4.4 Resize text (Level AA)
  - “Except for captions and images of text, text can be resized without assistive technology up to 200% without loss of content or functionality.”
  - Zoom to 200% is supported.
- 1.4.10 Reflow (Level AA - WCAG 2.1)
  - “Content can be presented without loss of information or functionality…” and without horizontal scrolling (except when necessary) at 400% zoom with viewport width at 1280 pixels.
- Consider text sizing
  - Perhaps to 130%-150%

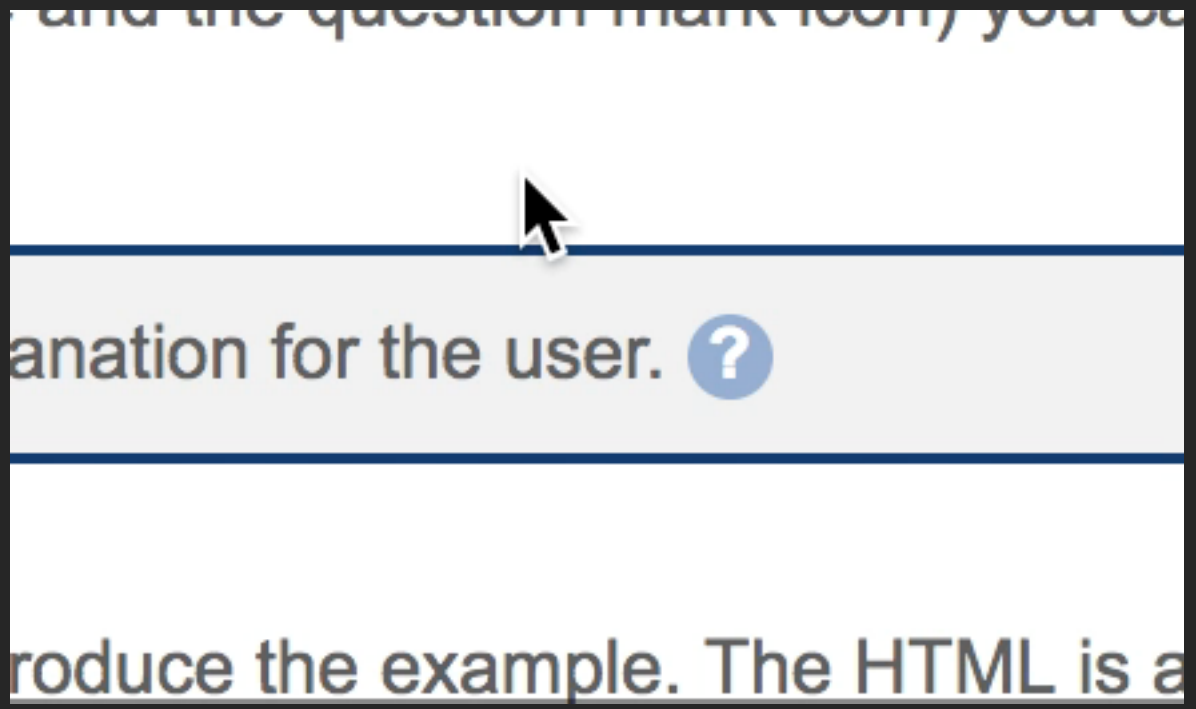
200% Zoom on CNN.com



Practice Browser Zoom

www.pitt.edu

Use click to trigger hover interactions



WCAG 2.1 – Content on Hover or Focus (Level AA)

Content that appears on hover and focus must be:

- “Dismissible…without moving pointer hover or keyboard focus”: Esc key dismisses
- “Hoverable”: Doesn’t disappear when moving the pointer to the new content
- “Persistent”: Visible until you move mouse away or dismiss it

Present messaging close to form controls

WCAG 2.1: Non-text Contrast

- Level AA
- 3:1 contrast of:
  - User Interface Components
    - Including states (focus indicators, hover states, etc.)
  - Graphical Objects

UI “Boundaries”

☐ I have read and agree to the Terms of Use

UI “Boundaries”

… except when “determined by user agent”

UI “States”

I have read and agree to the Terms of Use

common

sense

is

vital

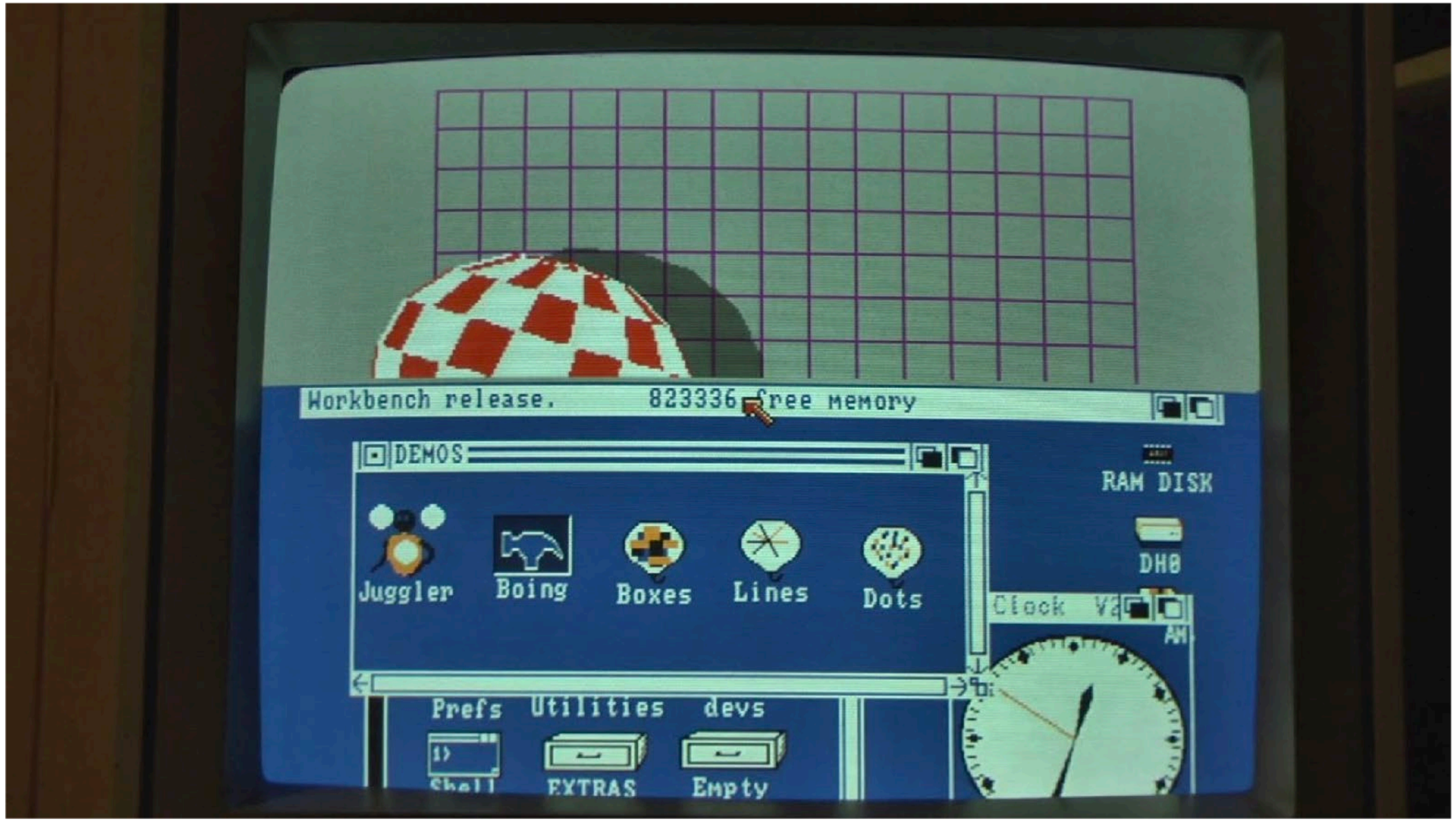
when

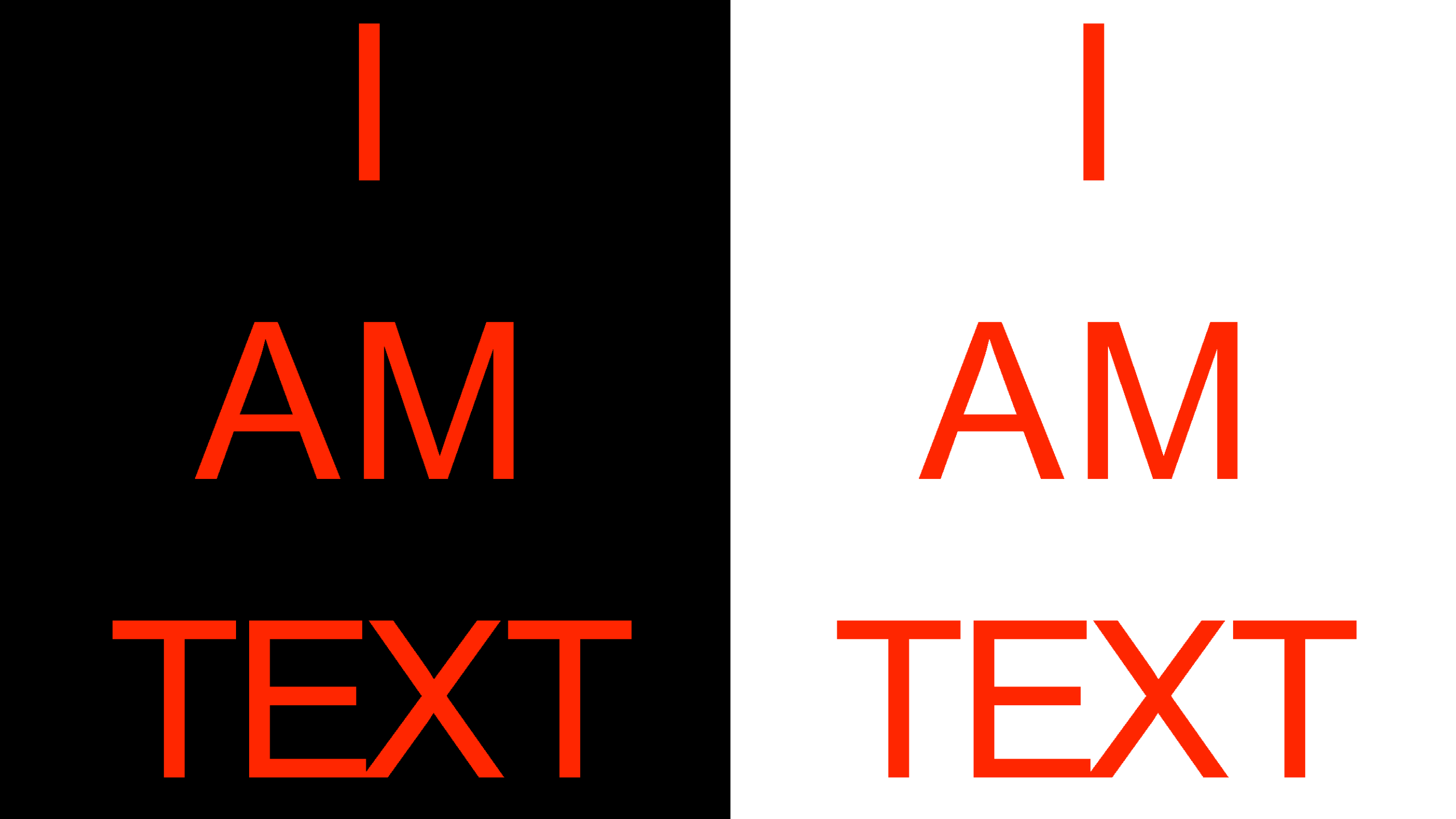
considering

color

contrast

pass
pass
pass
pass
fail
fail
fail
fail


Workbench release.
823336 free memory
DEMOS
Juggler
Boing
Boxes
Lines
Dots
RAM DISK
DH0
Clock
AM
Prefs
Utilities
devs
Shell
EXTRAS
Empty


I
AM
TEXT
I
AM
TEXT

5.25:1
4:1

I AM TEXT

I AM TEXT

4.5:1

4.5:1

I AM TEXT

I AM TEXT

2.79:1

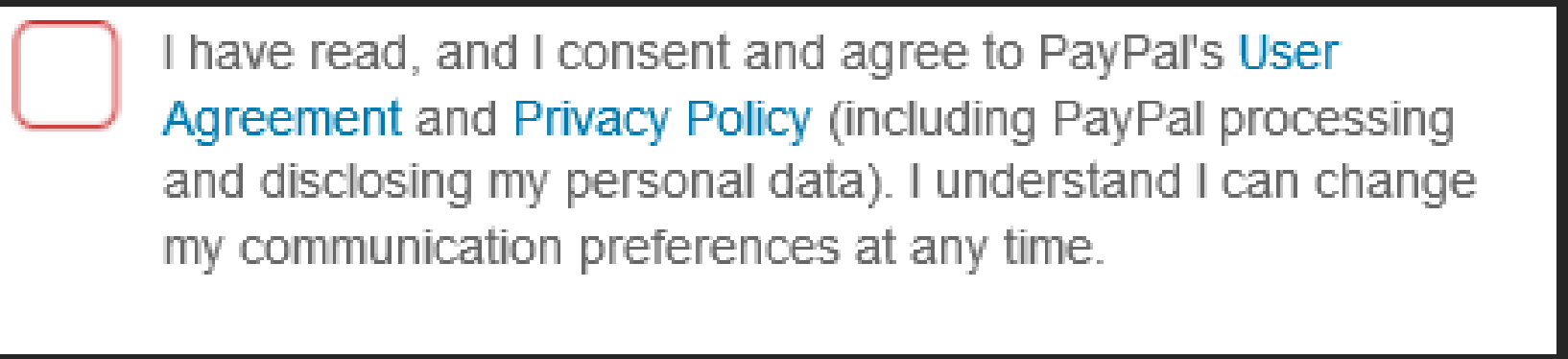
4.65:1

Color Blindness

1. Ensure sufficient contrast
2. Do not rely on color alone

Requirements for non-underlined links

- A 3:1 contrast ratio between link text and non-link text
- Link must present a non-color cue (typically underline) on mouse hover AND keyboard focus


I have read, and I consent and agree to PayPal's User Agreement and Privacy Policy (including PayPal processing and disclosing my personal data). I understand I can change my communication preferences at any time.

I have read, and I consent and agree to PayPal's User Agreement and Privacy Policy (including PayPal processing and disclosing my personal data). I understand I can change my communication preferences at any time.

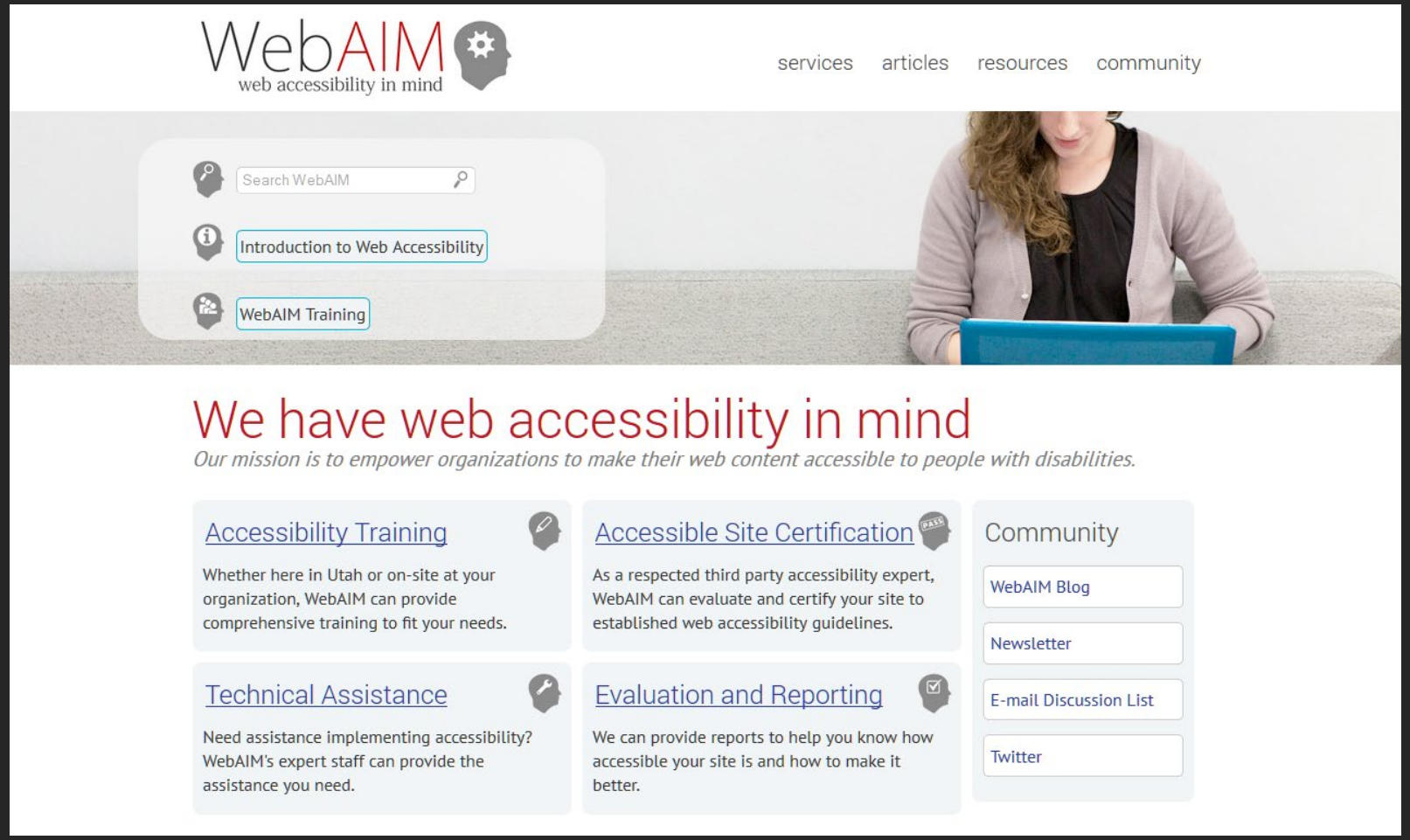
I have read and agree to PayPal's **User Agreement**, **Privacy Policy** and **Electronic Communications Delivery Policy**. If you provide a mobile phone number, you agree that PayPal may contact you to service your account, investigate fraud, or collect a debt, using autodialed or prerecorded calls or texts. You may decline to receive autodialed or prerecorded calls or texts and still use PayPal by contacting **customer support**.

I have read and agree to PayPal's **User Agreement**, **Privacy Policy** and **Electronic Communications Delivery Policy**. If you provide a mobile phone number, you agree that PayPal may contact you to service your account, investigate fraud, or collect a debt, using autodialed or prerecorded calls or texts. You may decline to receive autodialed or prerecorded calls or texts and still use PayPal by contacting **customer support**.

WebAIM Link Contrast Checker

webaim.org/resources/linkcontrastchecker/

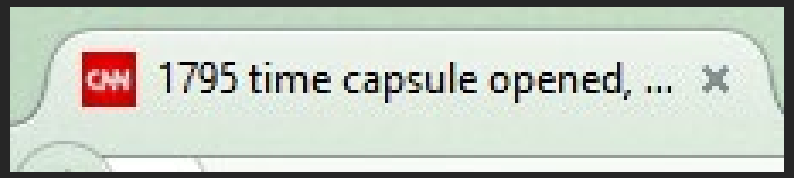
Blindness


WebAIM
web accessibility in mind
services
articles
resources
community
Search WebAIM
Introduction to Web Accessibility
WebAIM Training
We have web accessibility in mind
Our mission is to empower organizations to make their web content accessible to people with disabilities.
Accessibility Training
Whether here in Utah or on-site at your organization, WebAIM can provide comprehensive training to fit your needs.
Accessible Site Certification
As a respected third party accessibility expert, WebAIM can evaluate and certify your site to established web accessibility guidelines.
Technical Assistance
Need assistance implementing accessibility? WebAIM's expert staff can provide the assistance you need.
Evaluation and Reporting
We can provide reports to help you know how accessible your site is and how to make it better.
Community
WebAIM Blog
Newsletter
E-mail Discussion List
Twitter

Headings

- The most common method for finding information on a page
  - 67% of WebAIM survey respondents
- Headings should describe a section of content.
  - Should never be empty
- Usually one `<h1>` per page.
- In isolation, document headings serve as a table of contents.
- Do not skip heading levels (e.g., `<h2>` to `<h4>`).
  - Can skip backward (e.g., `<h4>` to `<h2>`).

Page Title



- `<title>1795 time capsule opened, …</title>`
- Often the first thing read
- Should be succinct and descriptive
- Put the most useful information first
- Should usually match or be similar to the `<h1>`

ARIA
Accessible Rich Internet Applications

First Rule of ARIA Use

If you can use a native HTML element or attribute with the semantics and behaviour you require **already built in**, instead of re-purposing an element and adding an ARIA role, state or property to make it accessible, **then do so**.

Notes on Using ARIA in HTML (w3.org/TR/aria-in-html/)

First Rule, Simplified

If you can use HTML, then do so.

HTML Regions

- <header>
- <nav>
- <main>
- <aside>
- <footer>

Region – ARIA Landmark Alignment

- <header> - role="banner"
- <nav> - role="navigation"
- <main> - role="main”
- <aside> - role="complementary"
- <footer> - role="contentinfo”
- ??? – role="search"

Regions/Landmarks

- “Region” / “Landmark” are generally synonymous
- Screen reader users can navigate by region
- All content in the page should be within a region.
- More doesn’t always mean better.
- We recommend HTML regions (Rule #1)
  - `<header role="banner">` is unnecessary

Alternative Text

Can be presented two ways…

- In the **alt attribute** of the img element.
  `<img alt="I am alt text">`
- In the **context** or surroundings of the image itself.

```
<a href="minivans.html">
      <img src="minivans.jpg" alt="???"><br>
      Minivans
</a>
```

```
<a href="minivans.html">
      <img src="minivans.jpg" alt=""><br>
      Minivans
</a>
```

Images that are the only thing within a link must **ALWAYS** have alternative text

... and image map hot-spots and image buttons too.

Image Link

Text Link

```
<a href="minivans.html">
<img src="minivans.jpg" alt="Minivans"></a>
<br>
<a href="minivans.html">Minivans</a>
```

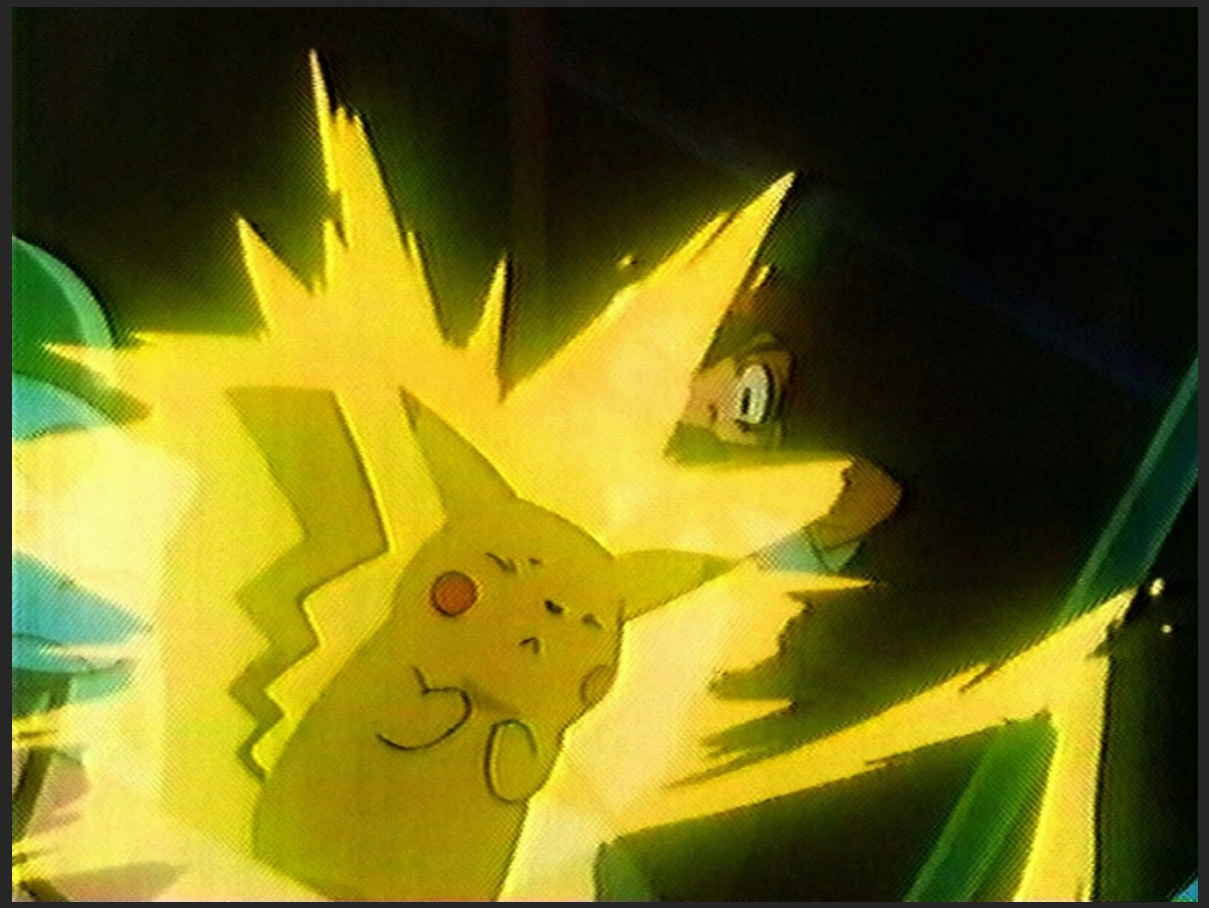
Download the Employment Application

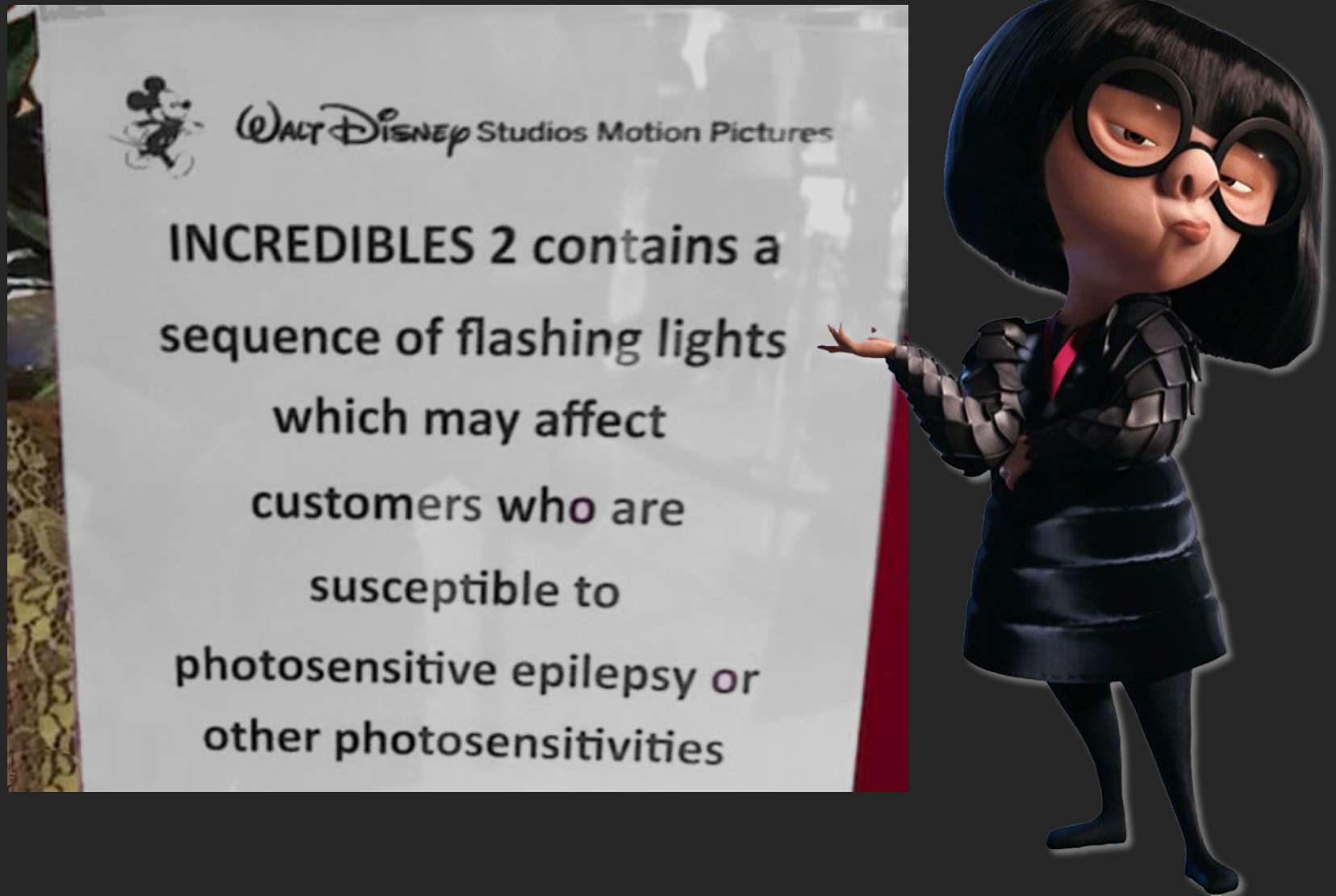
```
<a href="app.pdf">Download the
Employment Application
<img src="acrobat.png" alt="???"></a>
```

Download the Employment Application

```
<a href="app.pdf">Download the
Employment Application
<img src="acrobat.png" alt="PDF"></a>
```

Photosensitive Epilepsy




Walt Disney Studios Motion Pictures
INCREDIBLES 2 contains a
sequence of flashing lights
which may affect
customers who are
susceptible to
photosensitive epilepsy or
other photosensitivities

Be careful with flashing/strobing content

- 3 times per second or greater
- Size, brightness, and red threshold
- Annoying rule

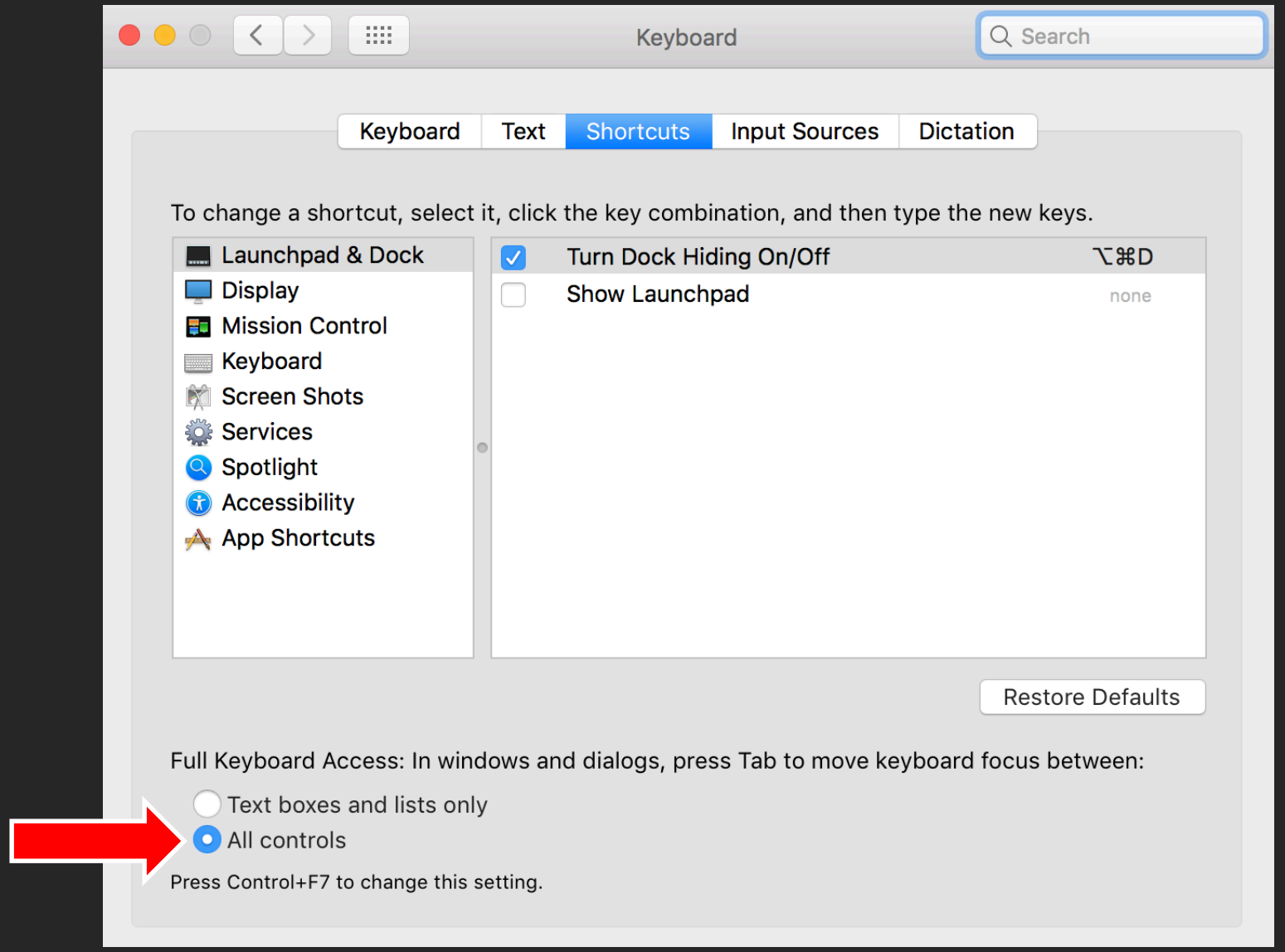
Motor disabilities

- Content and functionality should be accessible to mouse users and keyboard users
- Lack of fine motor control
- Repetition and fatigue
- Control over timing or moving elements

Standard keystrokes

- Navigate links, form controls, etc.: Tab, Shift + Tab
- Link: Enter
- Button: Enter **or** Spacebar
- Checkbox: Spacebar
- Radio buttons: ↑/↓ and ←/→
- Select menu: ↑/↓, Letters, Spacebar to expand
- Other widgets: use standard keyboard interaction patterns

Some tweaking required on Mac



2.5.3 – Label in Name (A)

“For user interface components with labels that include text or images of text, the name contains the text presented.”

2.5.3 Label in Name (A)

WCAG requires an accessible “label” (2.4.6, 3.3.2) and “name” (1.1.1)

- The **Label** is visually presented
- The **Name** is presented to assistive technology
  - Also called “accessible name”
  - May be visually hidden

What is the WCAG “Label”?

What is the WCAG “Name”?

```
<a href="page2.php"…>
    <img alt="Continue"…>
</a>
```

“Label” is not in “Name” – Fails 2.5.3

```
<a href="page2.php"…>
    <img alt="Continue"…>
</a>
```

“Label” is in “Name” – Passes 2.5.3

```
<a href="page2.php"…>
    <img alt="Next Page"…>
</a>
```

Better solution

```
<a href="page2.php"…>
   <img alt="Next"…>
</a>
```

Other principles

- Ensure adequate spacing between clickable elements
- Make sure clickable items look clickable
  (and vice versa)
- Don’t use the `accesskey` attribute
- Be careful with inactive/disabled controls

Hiding Content

- Hide from everyone:
  - `display:none`
  - `visibility:hidden`
  - HTML `hidden` attribute
- Ensure hidden elements are not navigable
  - Hide them with display:none or ensure they become visible when they receive focus
- Avoid 0 pixels, same color as background, etc.

Visually Hiding Content

- Position off-screen left with CSS for screen readers

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

Can also use CSS clip and/or clip-path

Do not remove the keyboard focus indicators from links

Avoid

```
a {outline:0;}
```

or

```
a {outline:none;}
```

WCAG AA Failure

Enhance the default focus indicators

```
a:focus {
      outline:1px solid #f068bb;
      background-color:#ff0;
      text-decoration:underline;
      }
```

Non-underlined links must become underlined on hover **and** focus.
Ensure 3:1 contrast.

```
a:hover {…}
```

```
a:hover, a:focus {…}
```

:focus-within

Hover and :focus-within row highlighting

Heading 1	Heading 2	Heading 3	Heading 4
Content 1 a	Content 2 a	Content 3 a	Content 4 a
Content 1 b	Content 2 b	Content 3 b	Content 4 b
Content 1 c	Content 2 c	Content 3 c	Content 4 c

```
tr:focus-within,
tr:hover {
  background: rgba(lightBlue, .4);
}
```

http://www.scottohara.me/blog/2017/05/14/focus-within.html

"Skip" links

```
<a href="#maincontent">Skip to
main content</a>

…

<main id="maincontent">...
```

WCAG 2.4.1 (Level A): A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.

Hidden "Skip" links

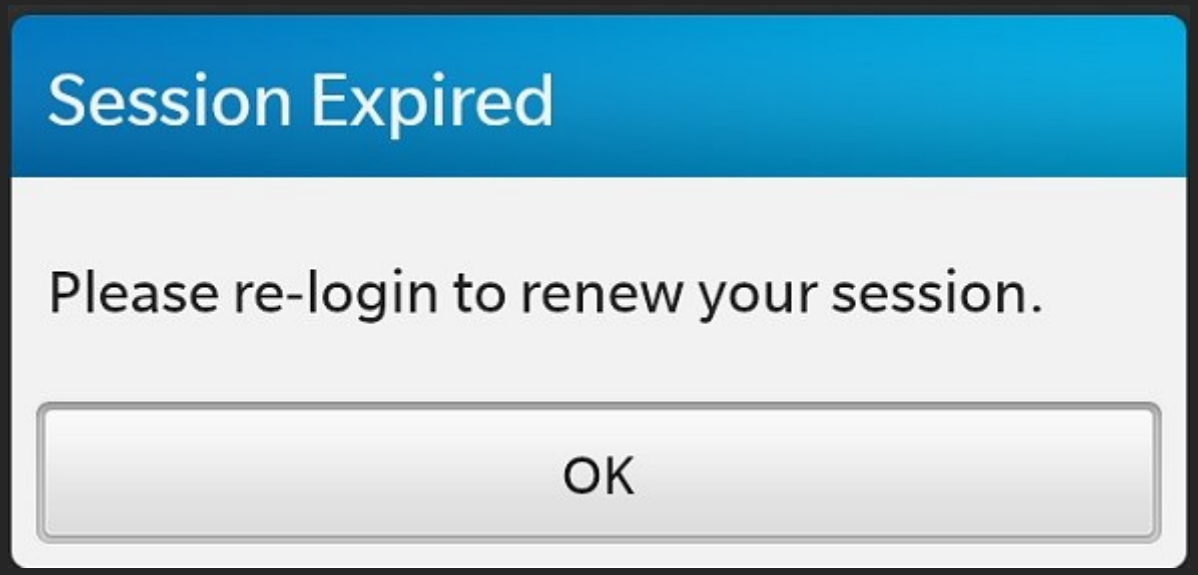
```
a#skip {
  position:absolute;
  left:-10000px;
  top:auto;
  width:1px;
  height:1px;
  overflow:hidden;
}

a#skip:focus {
  position:static;
  width:auto;
  height:auto;
}
```

Don't use `display:none` on the link

The Secret to Everlasting Happiness

Please finish reading this text – it will give you the secret to everlasting happiness. The secret is simple, all you need to do is to stop worrying about the key to everlasting happiness and enjoy the moment.


Session Expired
Please re-login to renew your session.
OK

WCAG Pause, Stop, Hide (Level A)

- For any moving, blinking or scrolling information that (1) starts automatically, (2) lasts more than five seconds, and (3) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it...; and
- For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update.

Use lists appropriately

- `<ol>` for ordered, hierarchical lists
- `<ul>` for unordered, bulleted lists
- `<dl>` for definition/description list (name/value pairs in HTML5)
  - FAQ

Frame Title

```
<frame title="navigation">
<frame title="main content">
```

```
<iframe title="advertisement">
```

```
<iframe src="http://youtube.com/abcd"
  title="Intro to Web Accessibility
                video">
```

iframe title is optional

Fonts & Text

- Absolute (px, pt, etc.) or relative sizes (%, em, etc.)
- Font faces
  - Web fonts and embedded fonts are OK
  - Choose good, legible fonts
- Be careful with ALL CAPS
  - Use CSS `text-transform:uppercase`
- Consider special characters and verbosity
  - $.99 vs $99
  - Minus symbol for negative numbers
  - Aural CSS – great idea, but no current support

WCAG 2.1 - Text Spacing (Level AA)

“No loss of content or functionality occurs” when the user increases spacing between:

- Paragraphs: 2 × font size
- Lines: 1.5 × font size
- Words: 0.16 × font size
- Letters: 0.12 × font size

Text Spacing Applied

Earlier guidelines [edit]

The first web accessibility guideline was compiled by Gregg Vanderheiden and released in January 1995, just after the 1994 Second International Conference on the World-Wide Web (WWW II) in Chicago (where Tim Berners-Lee first mentioned disability access in a keynote speech after seeing a pre-conference workshop on accessibility led by Mike Paciello).[2]

Over 38 different Web access guidelines followed from various authors and organizations over the next few years.[3] These were brought together in the *Unified Web Site Accessibility Guidelines* compiled at the University of Wisconsin-Madison.[4] Version 8 of the *Unified Web Site Accessibility Guidelines*, published in 1998, served as the starting point for the W3C's WCAG 1.0.[5]

WCAG 1.0 [edit]

The WCAG 1.0 was published and became a W3C recommendation on 5 May 1999. They have since been superseded by WCAG 2.0.

WCAG 1.0 consist of 14 guidelines—each of which describes a general principle of accessible design. Each guideline covers a basic theme of web accessibility and is associated with one or more checkpoints that describes how to apply that guideline to particular webpage features.

- Guideline 1: Provide equivalent alternatives to auditory and visual content
- Guideline 2: Don't rely on colour alone
- Guideline 3: Use markup and style sheets, and do so properly
- Guideline 4: Clarify natural language usage
- Guideline 5: Create tables that transform gracefully
- Guideline 6: Ensure that pages featuring new technologies transform gracefully
- Guideline 7: Ensure user control of time sensitive content changes
- Guideline 8: Ensure direct accessibility of embedded user interfaces
- Guideline 9: Design for device independence

Earlier guidelines [edit]

The first web accessibility guideline was compiled by Gregg Vanderheiden and released in January 1995, just after the 1994 Second International Conference on the World-Wide Web (WWW II) in Chicago (where Tim Berners-Lee first mentioned disability access in a keynote speech after seeing a pre-conference workshop on accessibility led by Mike Paciello).[2]

Over 38 different Web access guidelines followed from various authors and organizations over the next few years.[3] These were brought together in the *Unified Web Site Accessibility Guidelines* compiled at the University of Wisconsin-Madison.[4] Version 8 of the *Unified Web Site Accessibility Guidelines*, published in 1998, served as the starting point for the W3C's WCAG 1.0.[5]

WCAG 1.0 [edit]

The WCAG 1.0 was published and became a W3C recommendation on 5 May 1999. They have since been superseded by WCAG 2.0.

WCAG 1.0 consist of 14 guidelines—each of which describes a general principle of accessible design. Each guideline covers a basic theme of web accessibility and is associated with one or more checkpoints that describes how to apply that guideline to particular webpage features.

- Guideline 1: Provide equivalent alternatives to auditory and visual content
- Guideline 2: Don't rely on colour alone
- Guideline 3: Use markup and style sheets, and do so properly
- Guideline 4: Clarify natural language usage

[Text spacing bookmarklet]

Language of Page

```
<html lang="en">
```

Language of Page

Most people today can hardly conceive of life without the internet. Some have argued that no other single invention has been more revolutionary since Gutenberg's printing press in the 1400s. Now, at the click of a mouse, the world can be “at your fingertips”—that is, if you can use a mouse... and see the screen... and hear the audio—in other words, if you don't have a disability of any kind.

Language of Page

Most people today can hardly conceive of life without the internet. Some have argued that no other single invention has been more revolutionary since Gutenberg's printing press in the 1400s. Now, at the click of a mouse, the world can be “at your fingertips”—that is, if you can use a mouse... and see the screen... and hear the audio—in other words, if you don't have a disability of any kind.

Language of Parts

```
<div lang="fr">
```

Tables

Data tables

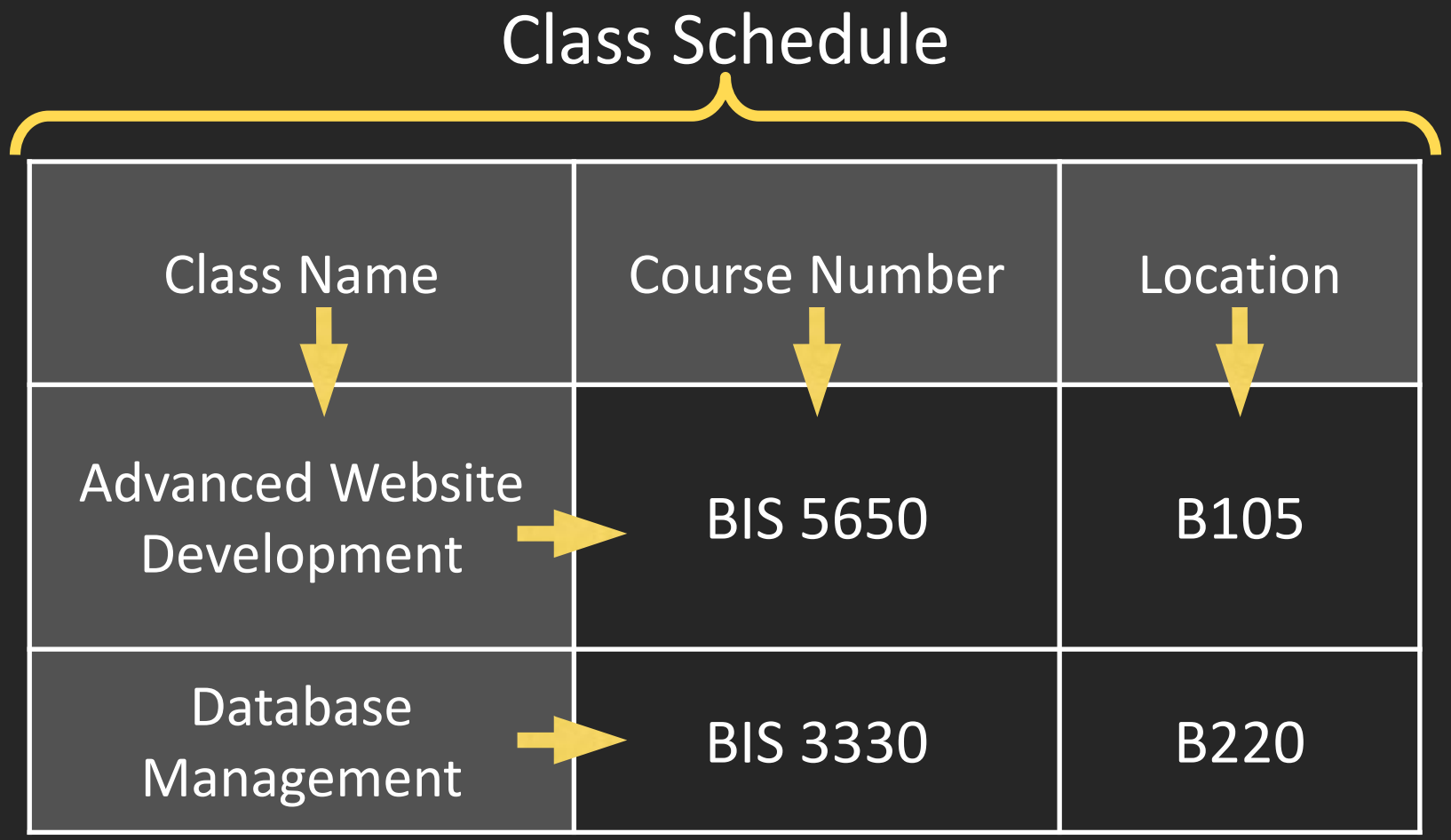
Class Schedule

Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

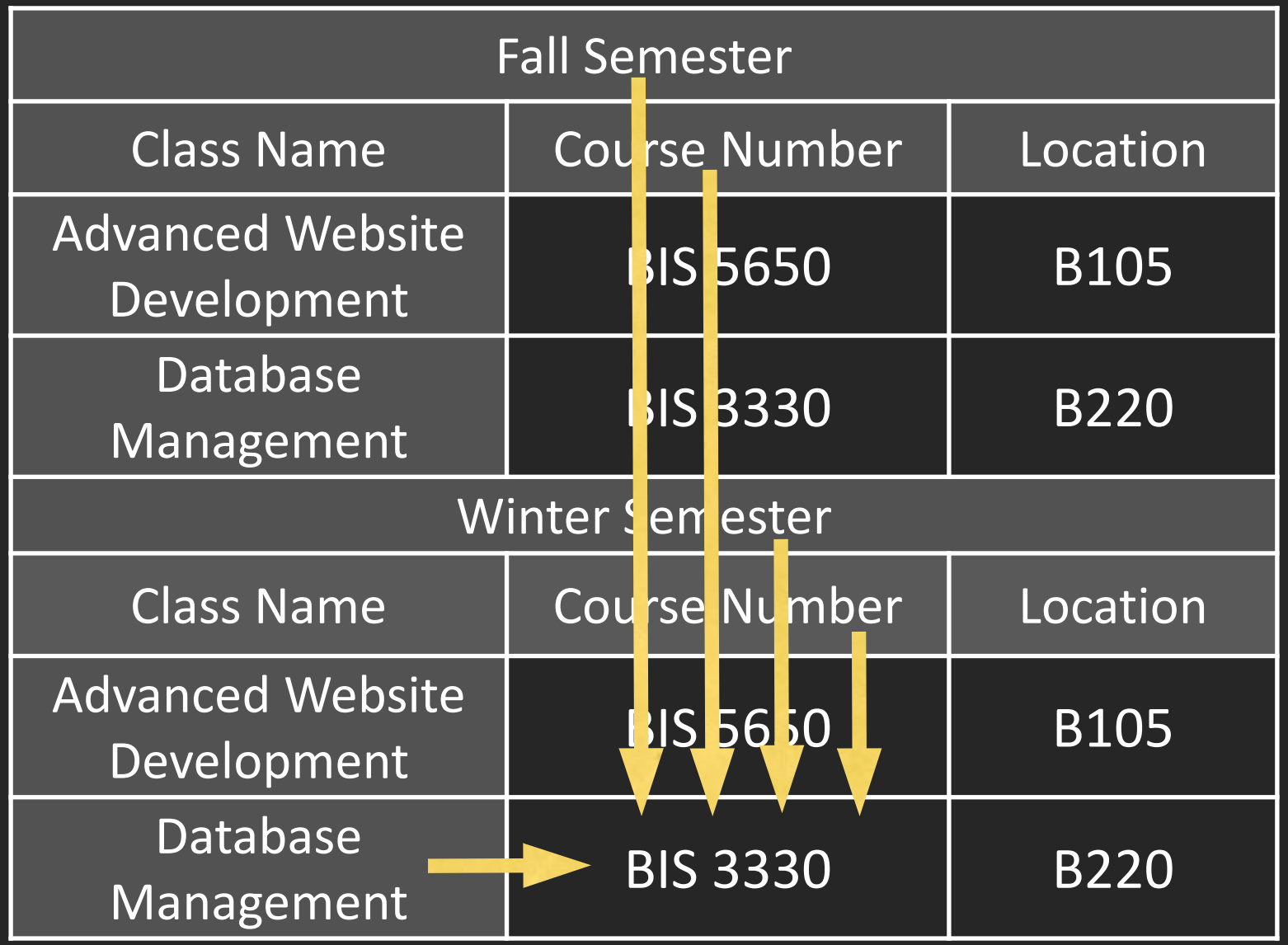
Data tables

```
<table>
<caption>Class Schedule</caption>
<tr>
  <th scope="col">Class Name</th>
  <th scope="col">Course Number</th>
  <th scope="col">Location</th>
</tr>
<tr>
  <th scope="row">Advanced Website Development</th>
  <td>BIS 5650</td>
  <td>B105</td>
</tr>
...
```

Data Tables


Class Schedule
Class Name
Course Number
Location
Advanced Website Development
BIS 5650
B105
Database Management
BIS 3330
B220

Fall Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220
Winter Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220


Fall Semester
Class Name
Course Number
Location
Advanced Website Development
BIS 5650
B105
Database Management
BIS 3330
B220
Winter Semester
Class Name
Course Number
Location
Advanced Website Development
BIS 5650
B105
Database Management
BIS 3330
B220

Fall Semester

Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Winter Semester

Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

headers and id

```
<th colspan="3" id="fall">Fall Semester</th>
...
<th id="cnum1" headers="fall">Course
Number</th>
...
<th id="advweb1" headers="fall cname1">Advanced
Website Development</th>
...
<td headers="fall cnum1 advweb1">BIS 5650</td>
```

Fall Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220
Winter Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Avoid headers and id if possible

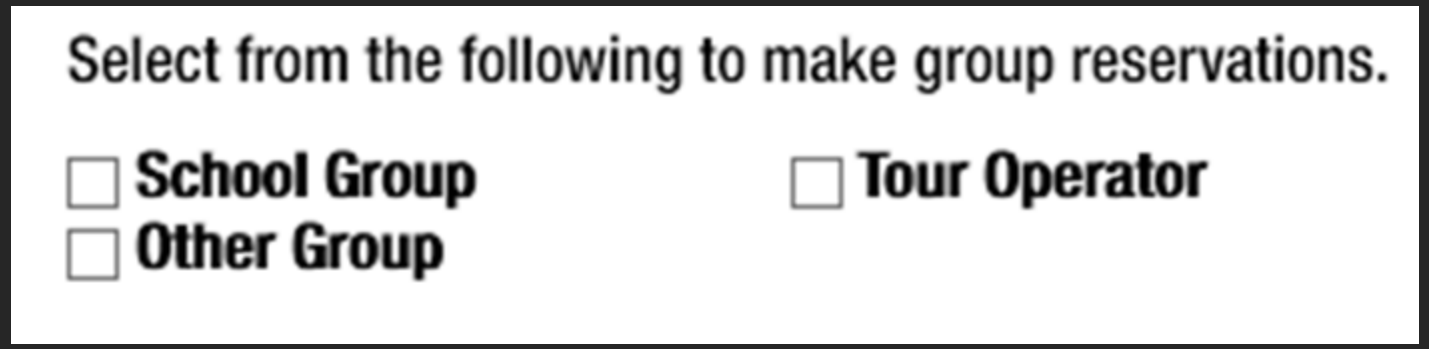
- It is a lot of work
- It can almost always be avoided
- It is often implemented incorrectly
- Screen reader support is not as strong
- Headers are often repeated

Tables

- Use data table markup only for data tables
- Use `<table role="presentation">` on layout tables
- Avoid empty `<th>`
- `<thead>`, `<tfoot>`, and `<tbody>` can be used, but don't provide any accessibility benefits.
- Abbreviated table header
  `<th abbr="First Name">First Name <button>Sort ascending</button></th>`
- Do not use table `summary`
- Use `<table>`, not `<div>`, for data tables.

Forms

Use the Correct Input Type



HTML5 Input Types

- <input type="date | time | search | url | number | range | color | email | tel">
- The browser can (or, more accurately, hopefully will) provide a natively accessible control/interface
- New form attributes - required, pattern, autocomplete, placeholder, autofocus, etc.

Form Labels

First Name:

```
<label for="fname">First Name:</label>
<input type="text" id="fname"…>
```

Text boxes, text areas, select menus, checkboxes, radio buttons, and HTML5 input types.

Implicit Form Labels

```
<label>First Name:
<input type="text"></label>
```

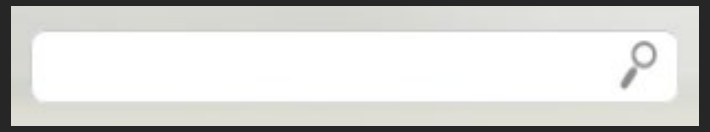
NEVER combine explicit and implicit

```
<label for="fname">First Name :
<input type="text" id="fname"></label>
```

<fieldset> and <legend>

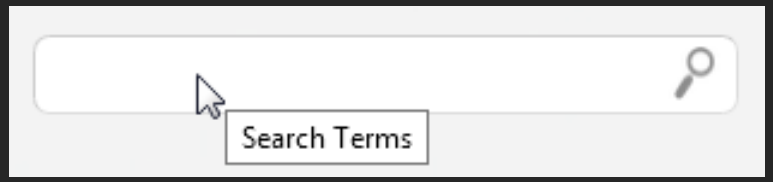
```
<fieldset>
  <legend>Choose a shipping method:</legend>
  <input id="overnight" type="radio"…>
  <label for="overnight">Overnight</label>
  ...
</fieldset>
```

Hidden <label>



```
<label class="sr-only" for="s">
    Search Terms</label>
<input type="text" id="s">
```

Title Attribute

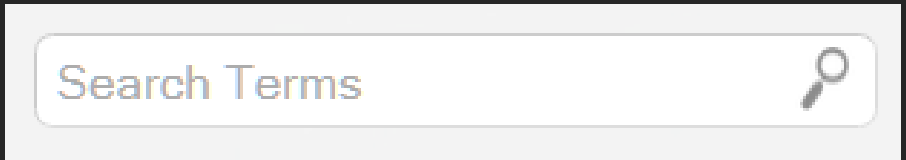



```
<input title="Search Terms"…>
```

Title Attribute

- Advisory information only
- Should be read for:
  - Form elements missing labels
  - Frames and iframes
  - `<abbr>`/`<acronym>`...usually
    - Typically best to avoid `<abbr>` and `<acronym>`
- May or may not be read in other places (links, images, etc.)
- Not accessible to keyboard users, touch screen users, etc.

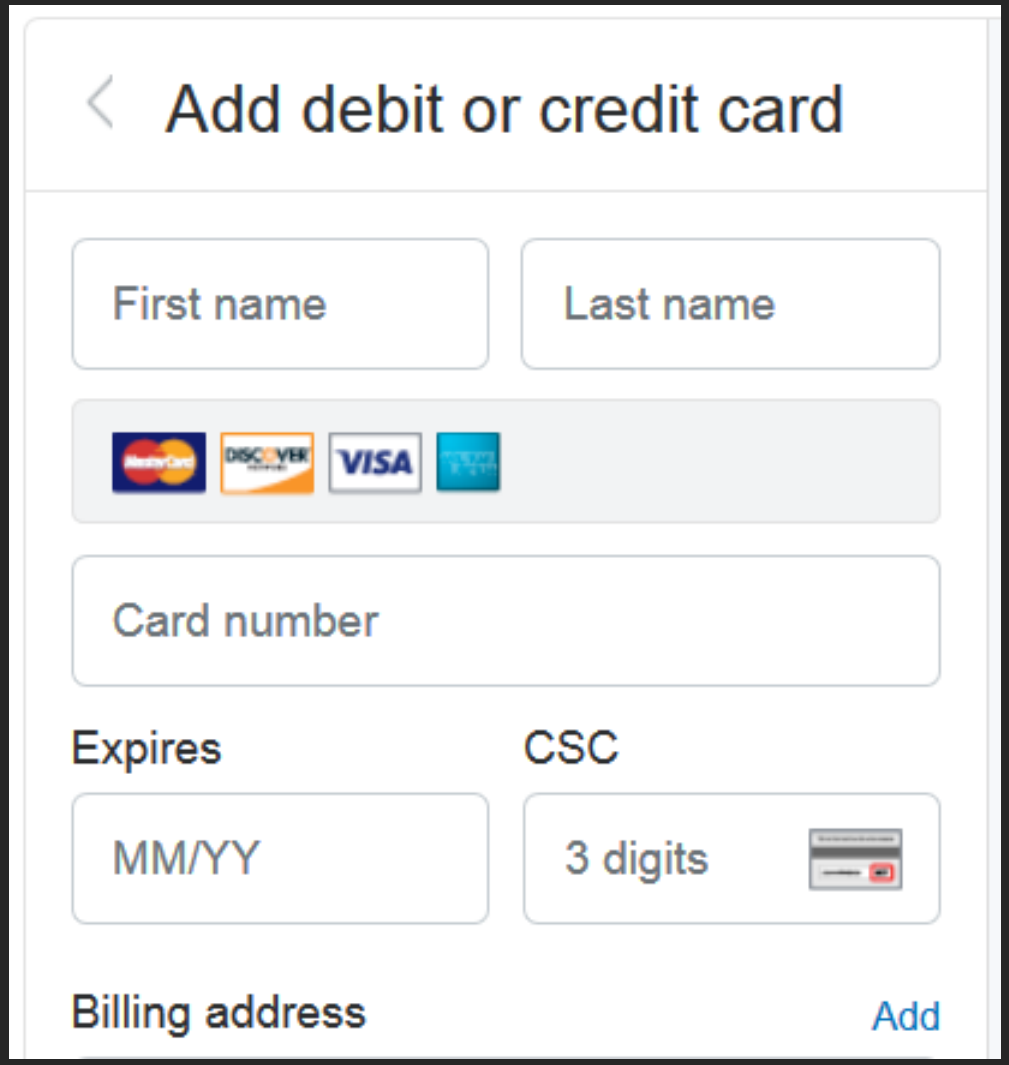
Placeholder

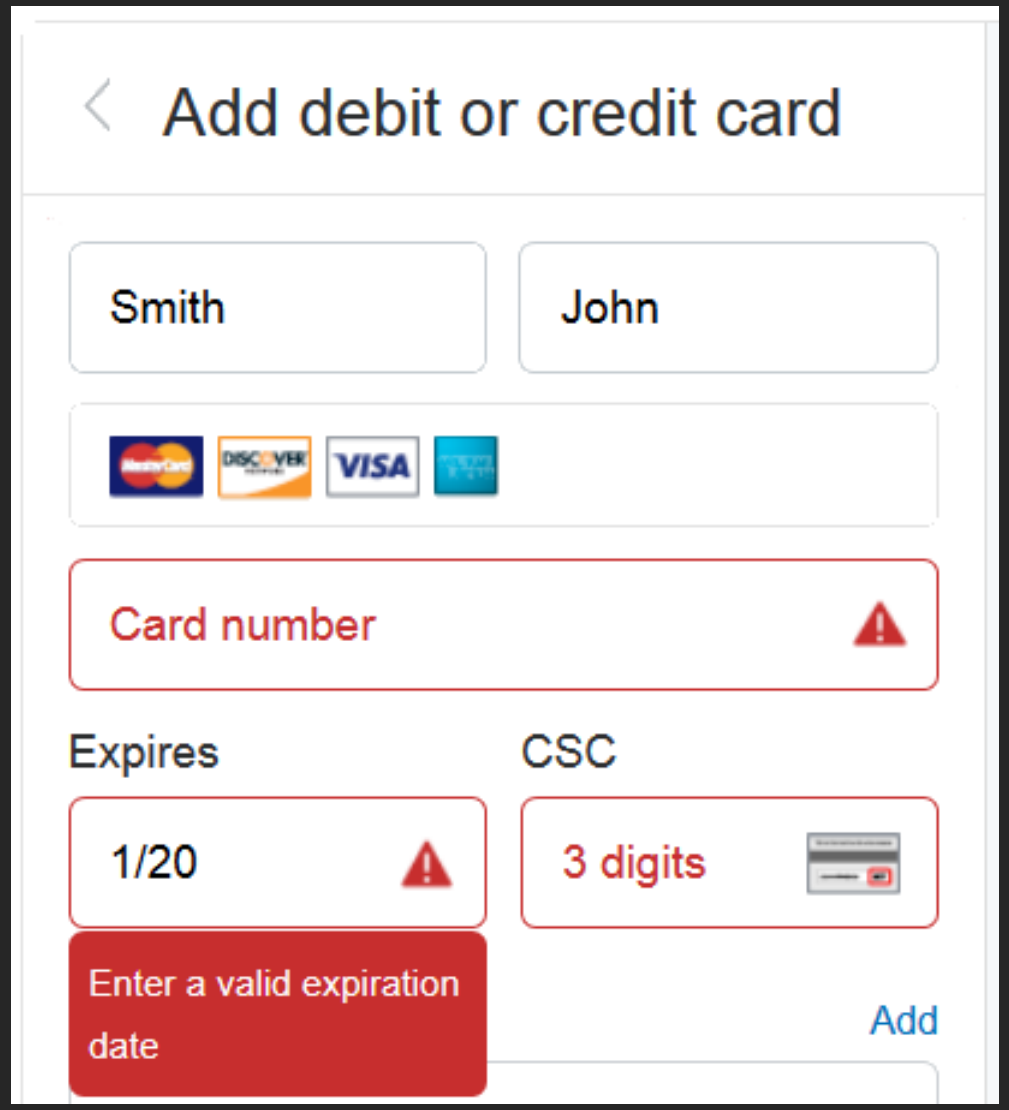



```
<input placeholder="Search Terms"…>
```

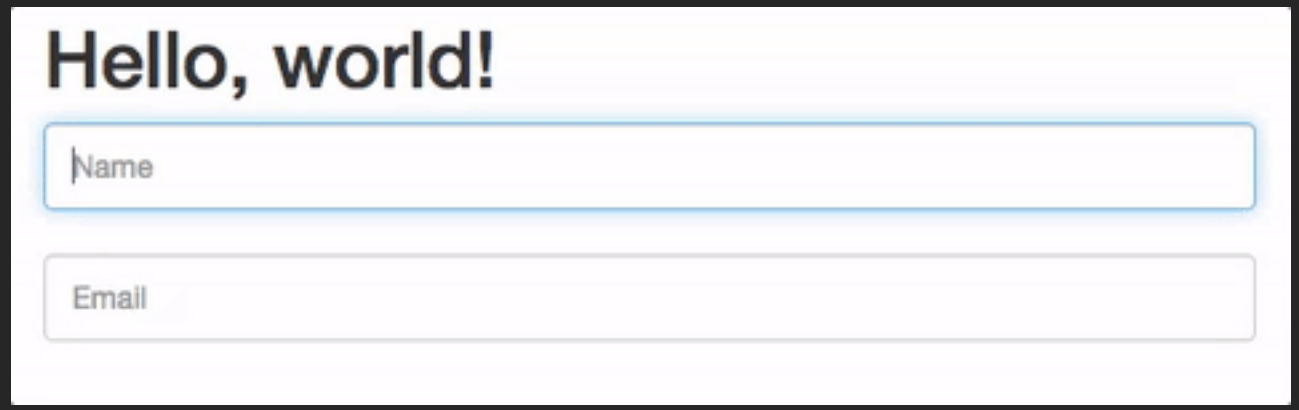
- Placeholder is usually read by a screen reader, but is not a suitable label.
- If `placeholder` duplicates a hidden `<label>`, they must match exactly, or both will be read.

name@example.com
password
confirm password
first name
last name
birthday


Add debit or credit card
First name
Last name
Card number
Expires
CSC
MM/YY
3 digits
Billing address
Add


Add debit or credit card
Smith
John
VISA
Card number
Expires
CSC
1/20
3 digits
Enter a valid expiration date
Add

Floating labels



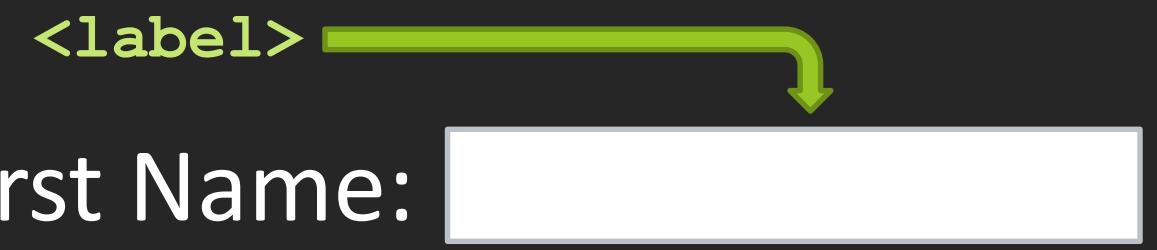
Potential issues:

- Size
- Contrast
- Confusion

Limitations with <label>

1:1 relationship between `<label>` and a form control.

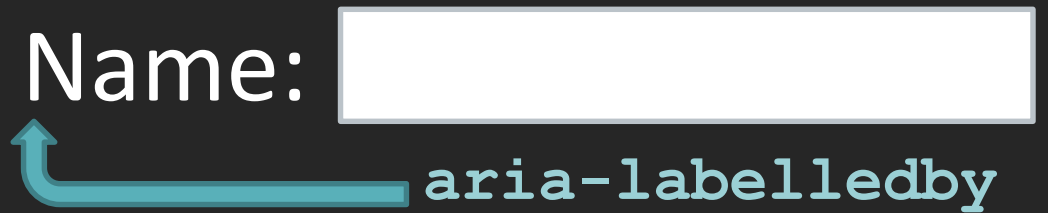
- A `<label>` cannot apply to more than one form control.
- A form control cannot have more than one `<label>`.




```
<label for="fname">First name:</label>

<input type="text" id="fname">
```

First Name:




```
<span id="fnamelabel">First name:</span>

<input type="text"
aria-labelledby="fnamelabel">
```

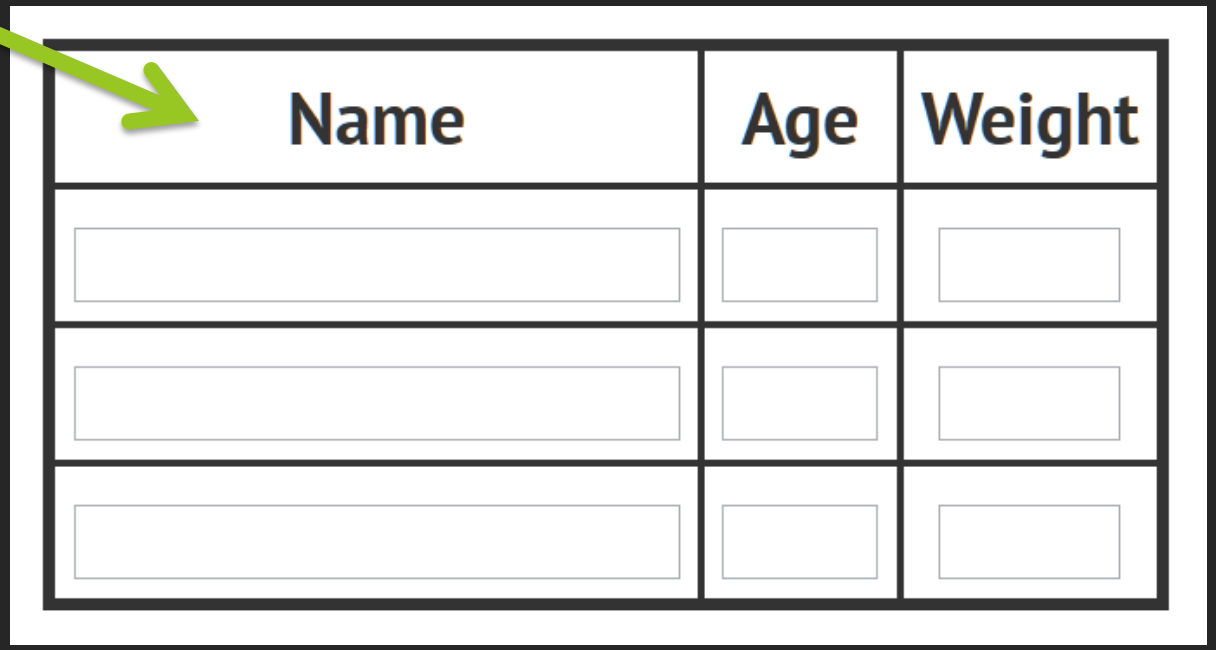
```
<label for="fname" id="fnamelabel">
First name:</label>

<input type="text" id="fname"
aria-labelledby="fnamelabel">
```

Unnecessary markup (Rule #1 of ARIA), but does not cause any problems.

One label for multiple controls

id="namelabel"

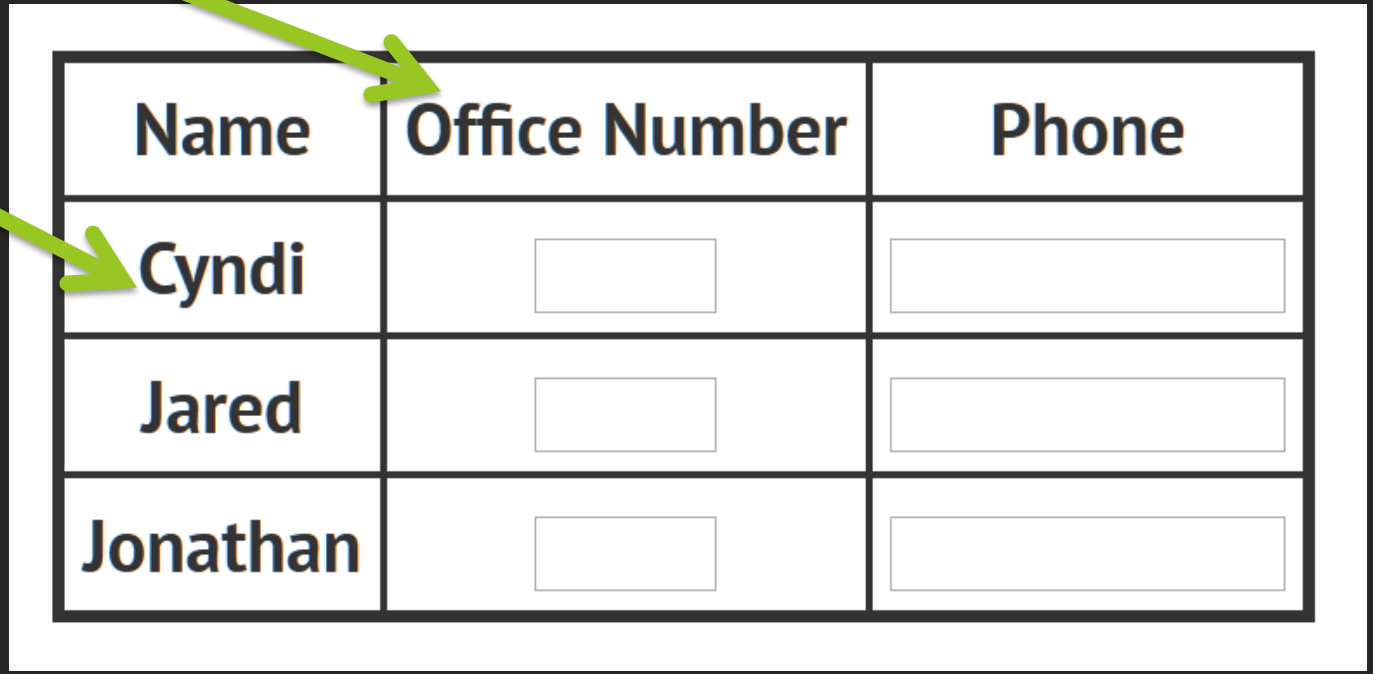



```
<input type="text" name="name1"
aria-labelledby="namelabel">
```

Multiple labels for one control

id="officenum"

id="cyndi"



Name	Office Number	Phone
Cyndi		
Jared		
Jonathan		

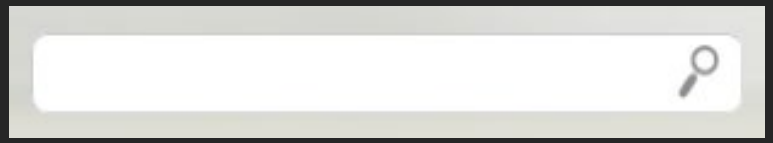
```
<input type="text" name="office1"
aria-labelledby="cyndi officenum">
```

Form values as labels for other inputs

Name	Office Number	Phone
Cyndi		
Jared		
Jonathan		

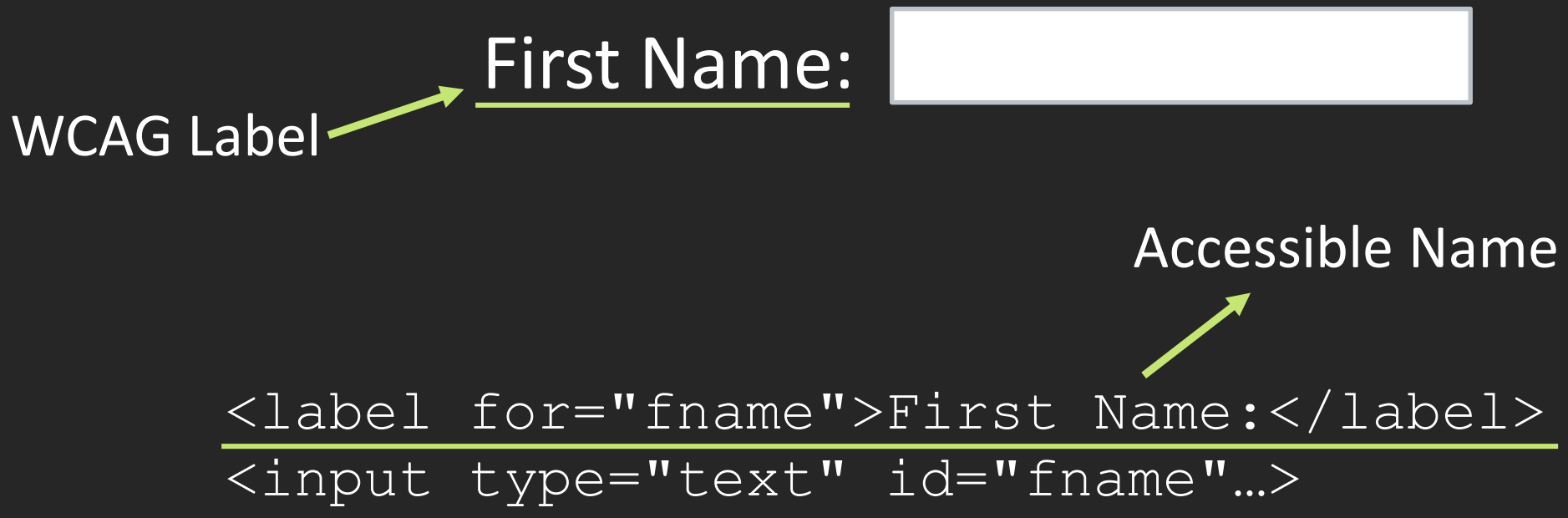
```
<input type="text" name="office1"
aria-labelledby="name1 officenum">
```

aria-label attribute

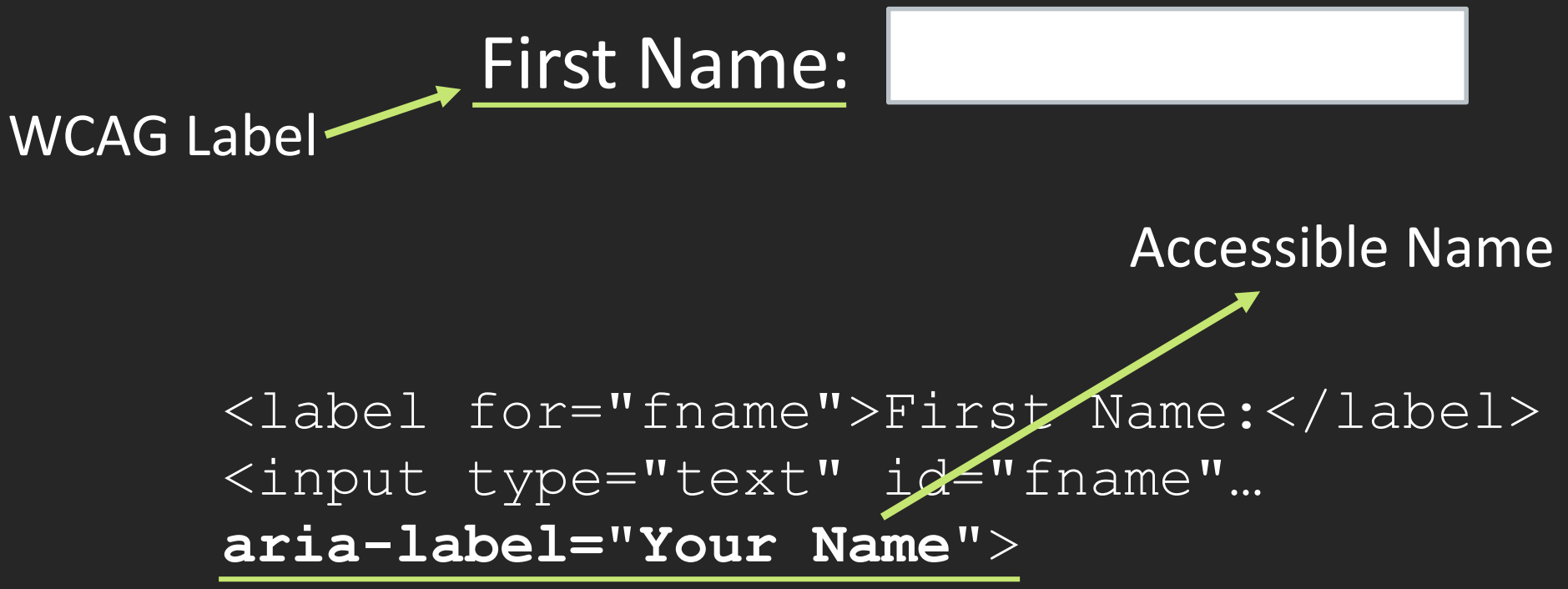


```
<input aria-label="Search Terms"…>
```

ARIA *Labels* override default Accessible *Names*



ARIA *Labels* override default Accessible *Names*



CSS Generated Content

Download the Employment Application

```
<a class="PDFlink">Download the Employment
Application</a>
```

```
.PDFlink::after {
  font-family: FontAwesome;
  content: "\f1c1";
}
```

Generated content may or may not be read.

Icon Fonts or Background Images/Sprites

Download the Employment Application

```
<a class="PDFlink">Download the Employment
Application<span class="sr-only"> (PDF)</span></a>
```

```
<a class="PDFlink" aria-label="Download the
Employment Application (PDF)">Download the
Employment Application</a>
```

aria-describedby

Username:

Must be 8-15 characters

```
<label for="user">Username:</label>
<input type="text" id="user"
aria-describedby="usernamedetails">
<p id="usernamedetails">Must be 8-15 characters</p>
```

ARIA and HTML labels and descriptions

- Elements must be “labelable”
  - Link, form control, table, etc. NOT , , <p>, etc.
  - Or have an appropriate ARIA role
- Are read as a stream of text. Users cannot easily navigate or explore the referenced content.
- Semantics (lists, links, etc.) are removed from referenced content.
- Should be succinct.

Hidden ARIA and HTML labels and descriptions

- `<label>` hidden with CSS `display:none` **will not** be read by screen readers.
- Elements hidden with `display:none`, but referenced by `aria-labelledby` or `aria-describedby` **WILL** be read.

```
<label for="user">Username:</label>
<input type="text" id="user"
aria-describedby="usernameerror">
<p id="usernameerror"
style="display:none">This username is already
taken. Please try again.</p>
```

aria-required

Username:*

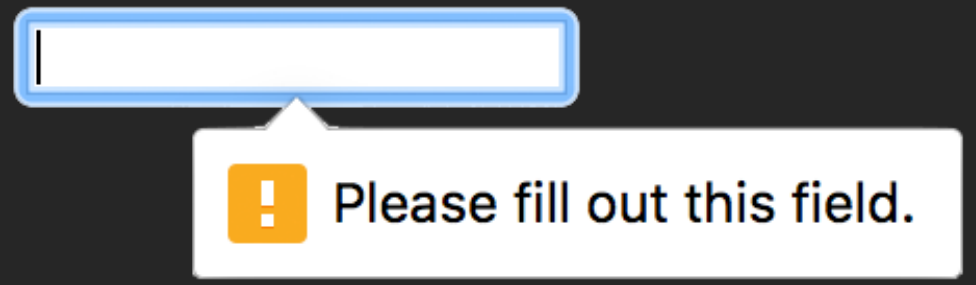
```
<label for="username">Username:</label>*
<input type="text" id="username"
aria-required="true">
```

A screen reader indicates the field is required…

and that's all.

HTML required

Username:*




```
<label for="username">Username:</label>*
<input type="text" id="username" required>
```

Semantics are the same, but validation messaging is provided

We don't share your financial details with the merchant.

United States

Card number

Expires

MM YY

CSC

3 digits

First name

Last name

Email

Mobile

Phone number

Billing address

Street address

Apt., ste., bldg., etc. (optional)

City

State

ZIP

Ship to my billing address

PayPal is the safer, easier way to pay

No matter where you shop, we keep your financial information secure.

aria-invalid

Password:

```
<label for="password">Password:</label>
<input type="password" id="password"
aria-invalid="true"…>
```

A screen reader indicates the field is invalid or broken…

and that's all.

Use ARIA attributes to control styling:

```
[aria-invalid=true] {
  border: 2px solid red;
  background-image: url("error.png");
}
```

Autocomplete attribute

First Name:

```
<label for="fname">First Name:</label>
<input type="text" id="fname"
autocomplete="given-name">
```

WCAG 2.1 – Identify Input Purpose (Level AA)

The purpose of each input field collecting information about the user can be programmatically determined when:

- The input field serves a purpose identified in the Input Purposes for User Interface Components section; and
- The content is implemented using technologies with support for identifying the expected meaning for form input data.

Currently requires appropriate `autocomplete` attribute values for inputs that collect user information.

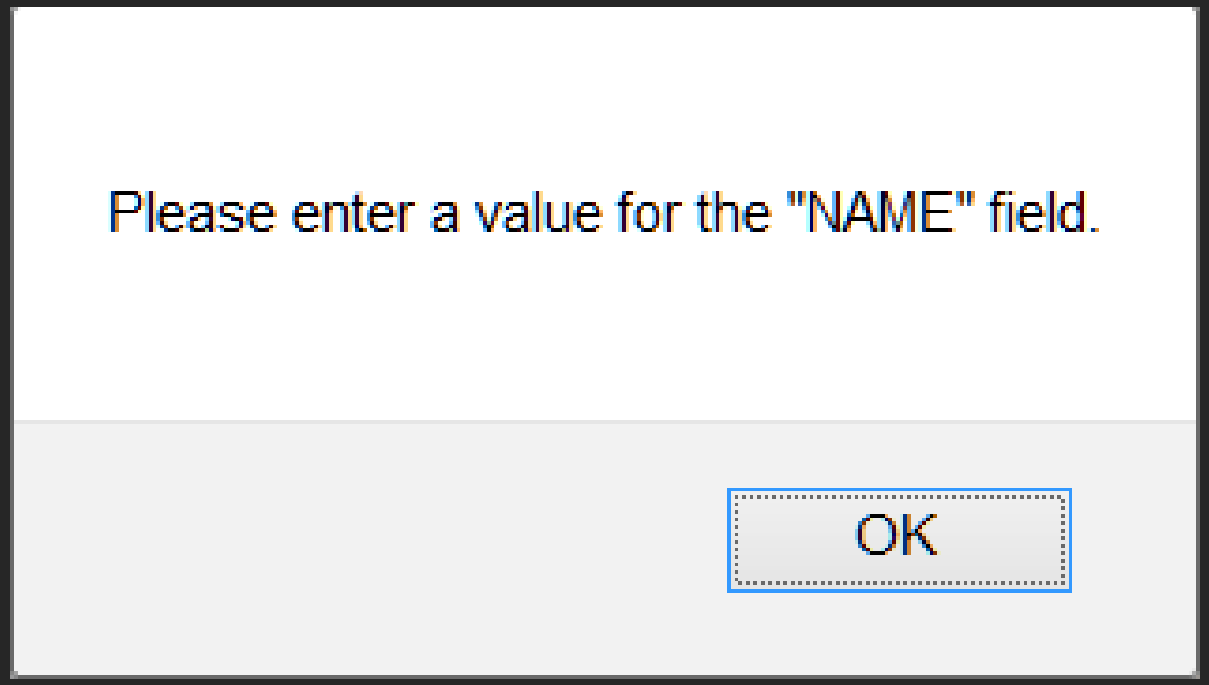
Form Validation and Error Recovery

- Avoid
  - Collecting unnecessary information
  - Forcing users to provide formatting unless necessary
- Identify
  - Let the users know there is an error
  - Direct users to errors
  - Describe the error
- Suggest
  - How to fix it
- Prevent
  - Important changes (e.g., Financial) are reversible/checked/confirmed

Form Validation Types

- Alert, then focus
- Errors at the top
- Inline errors

Alert, then focus

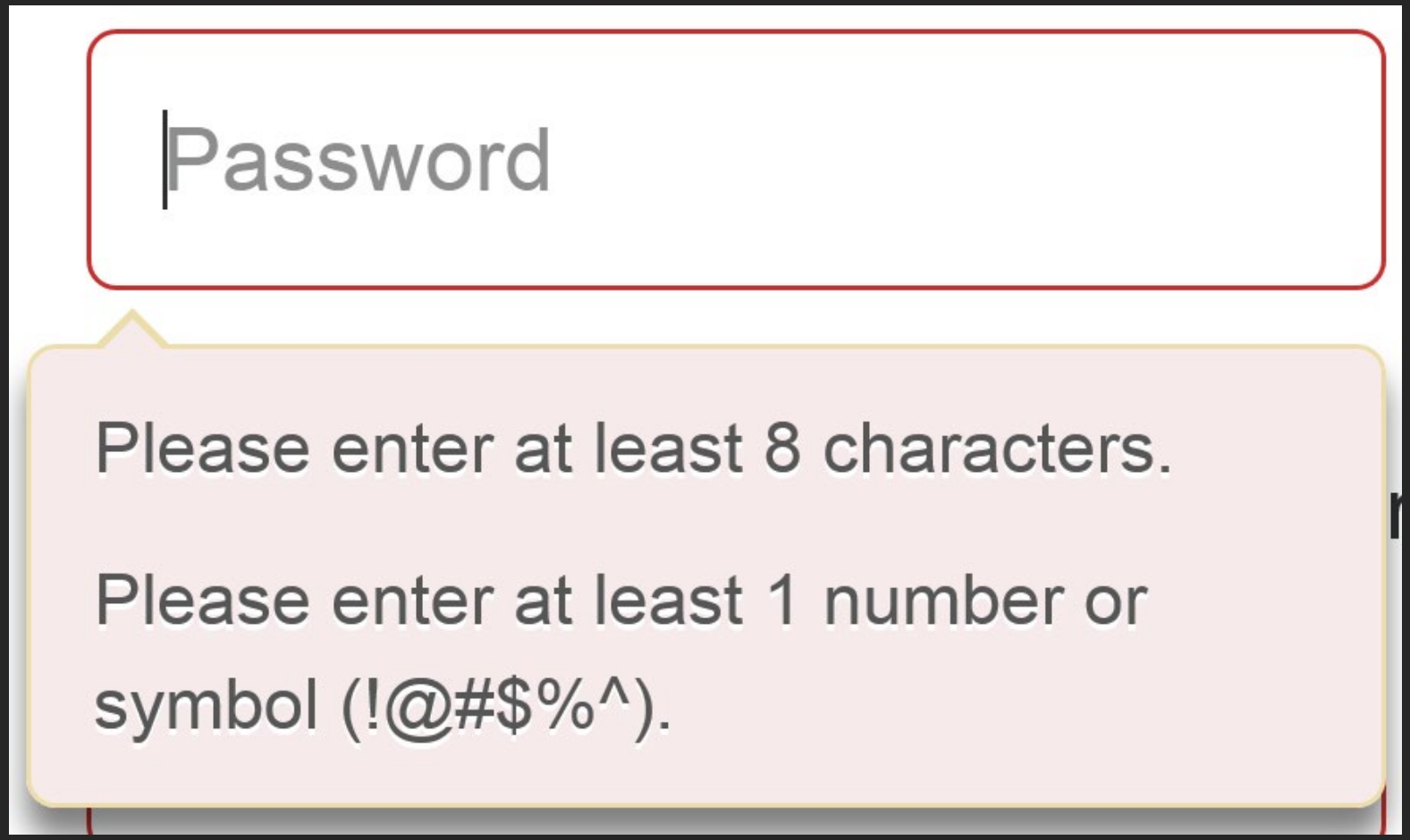


Errors at the top

There was a problem with the form.

- Please enter Your Name.
- Please enter Your E-mail Address.

Inline Errors



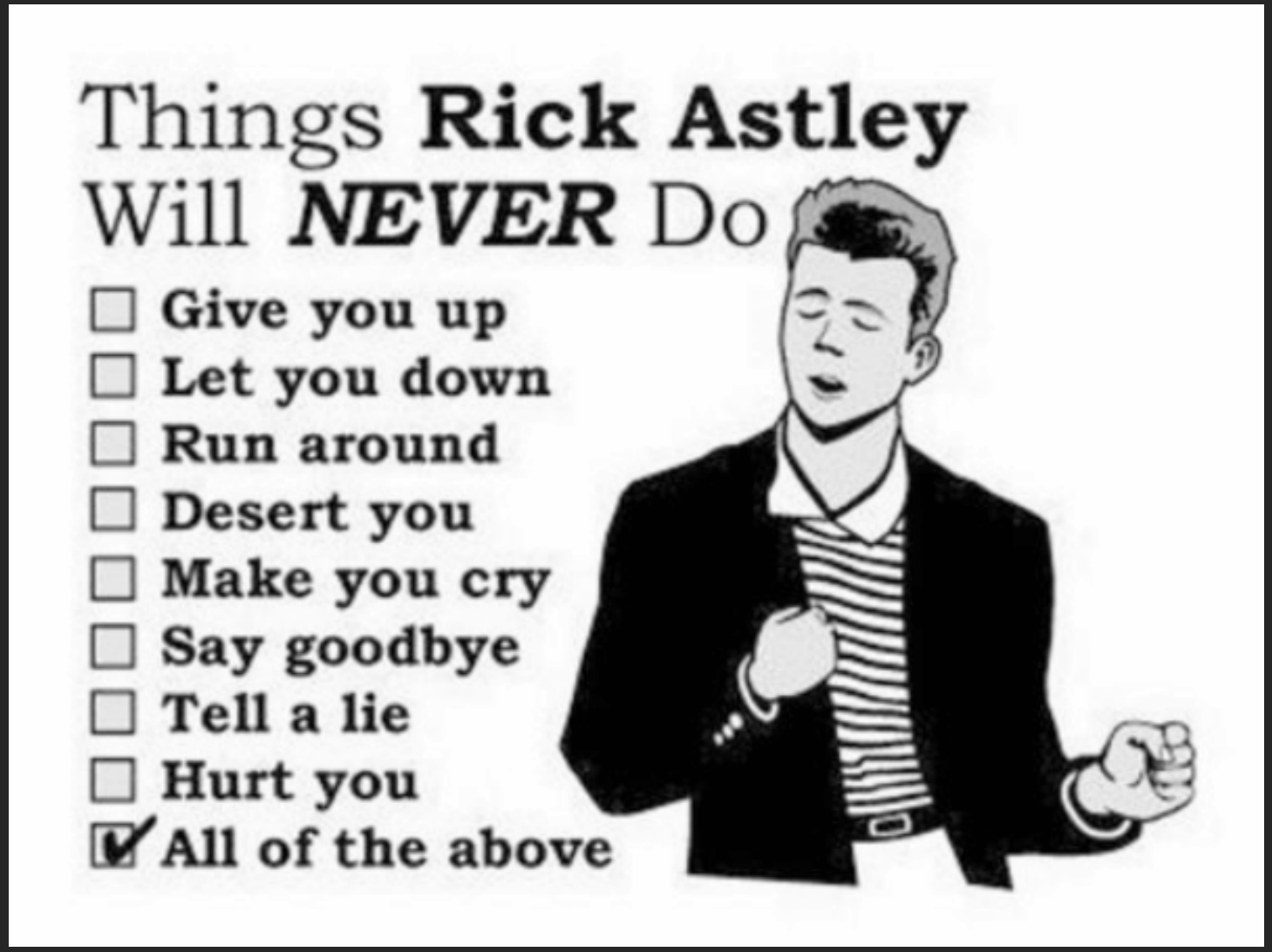
Evaluating accessibility

Accessibility is about people

Only people can evaluate
true accessibility

wave.webaim.org

Use a checklist



Use a checklist

- WCAG 2 -webaim.org/standards/wcag/checklist/
- Customized

WebAIM
web accessibility in mind

WCAG 2 Checklist

Principle 1: Perceivable

Web content is made available to the senses - sight, hearing, and/or touch

Guideline 1.1 Text Alternatives

Provide text alternatives for any non-text content

Success Criteria	Recommendations
1.1.1 Non-text Content (Level A)	☐ All images, form image buttons, and image map hot spots have appropriate, equivalent alternative text. ☐ Images that do not convey content, are decorative, or contain content that is already conveyed in text are given null alt text (alt="") or implemented as CSS backgrounds. All linked images have descriptive alternative text. ☐ Equivalent alternatives to complex images are provided in context or on a separate linked page. ☐ Form buttons have a descriptive value. ☐ Form inputs have associated text labels. ☐ Embedded multimedia is identified via accessible text. ☐ Frames and iframes are appropriately titled.

Guideline 1.2 Time-based Media

Provide alternatives for time-based media

NOTE: If the audio or video is designated as an alternative to web content (e.g., an audio or sign language version of a web page, for example), then the web content itself serves as the alternative.

Success Criteria	Recommendations
1.2.1 Prerecorded Audio-only and Video-only (Level A)	☐ A descriptive text transcript that includes relevant auditory content is provided for non-live audio-only (audio podcasts, MP3 files, etc.). ☐ A descriptive text transcript or audio description is provided for non-live video-only (e.g., video that has no audio track), unless the video is decorative.
1.2.2 Captions (Prerecorded)	☐ Synchronized captions are provided for non-live video (YouTube videos, etc.).

Screen reader testing

Common screen readers

- Windows
  - JAWS
    - JAWS + IE is still common
  - NVDA
  - Narrator: Uncommon but growing
- Mac and iOS – VoiceOver
  - 75% of mobile
- Android
  - Talkback (mostly)
- ChromeVox: Very uncommon

Use a screen reader

- Focus on navigation, forms, document structure, and dynamic content
- WebAIM tutorials for JAWS, NVDA, and VoiceOver
- Try testing without the mouse or monitor
- Don’t worry about pronunciation
- Sometimes it's the screen reader's fault

Screen reader shortcuts

Command	JAWS/NVDA	VoiceOver
“Special” key	Insert	control + option = VO
Read all	Insert + down arrow	VO + A
Read one line	Down arrow	VO + Arrow
Stop reading	Ctrl	Control
Links/Form fields	Tab	tab
Headings	H, 1-6	VO + command + H
Forms	F	VO + command + J
Tables	T	VO + command + T
“Rotor”		VO + U + Arrow keys
Landmarks	R (JAWS), D (NVDA)	Rotor

Do not conduct accessibility testing with users with disabilities

Wait… what?

Conduct user testing
and include users with disabilities

Thank You!

http://webaim.org

- E-mail discussion list
- Monthly newsletter
- Tutorials, articles, and resources
- Blog

WebAIM
web accessibility in mind