

# **Accessibility**

- Types of Disabilities
- Accessibility Approaches
- Implementing Web Accessibility

# (User Interface) Accessibility

*Definition:* Developing content to be as accessible as possible, no matter an individual's physical and cognitive abilities and how they access a user interface.

- Often abbreviated as "a11y", sometimes pronounced "ally"
- Parallel with physical accessibility:
  - It is wrong (even illegal in many cases) to exclude someone from a physical building because they are in a wheelchair.
  - It is wrong (even illegal in many cases) to exclude someone from a website because they have a visual impairment.
- Access to user interfaces is considered a human right

# Abilities

People vary in physical and mental capabilities

- How are your abilities different from other people?
  - How will your abilities change in the future?
  - How do your abilities change in different environments?
- 
- The “average person” is just a statistical construct
    - The people who will use your interface are different than you

Permanent

Temporary

Situational

Touch



One arm



Arm injury



New parent

See



Blind



Cataract



Distracted driver

Hear



Deaf



Ear infection



Bartender

Speak



Non-verbal



Laryngitis



Heavy accent

great a11y  
resource



London street has record cell phone texting injuries

- <https://youtu.be/807vebt-mmQ>

## Brick Lane made Britain's first 'Safe Text' street with padded lampposts to prevent mobile phone injuries

Last updated at 17:00 04 March 2008



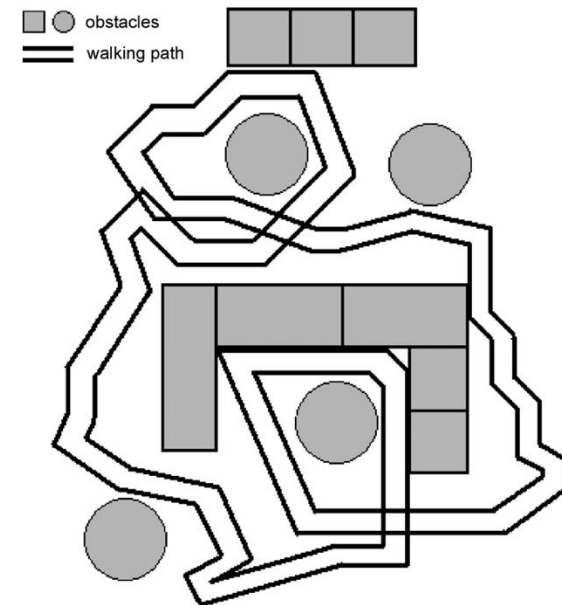
Collision course: Padding around a lamppost in Brick Lane, London. The move is part of the 'safe text' drive to cut the number of phone users injured in street accidents

Actuality, it was a clever  
Guerrilla marketing campaign  
... but it illustrates a point

# Input while Walking

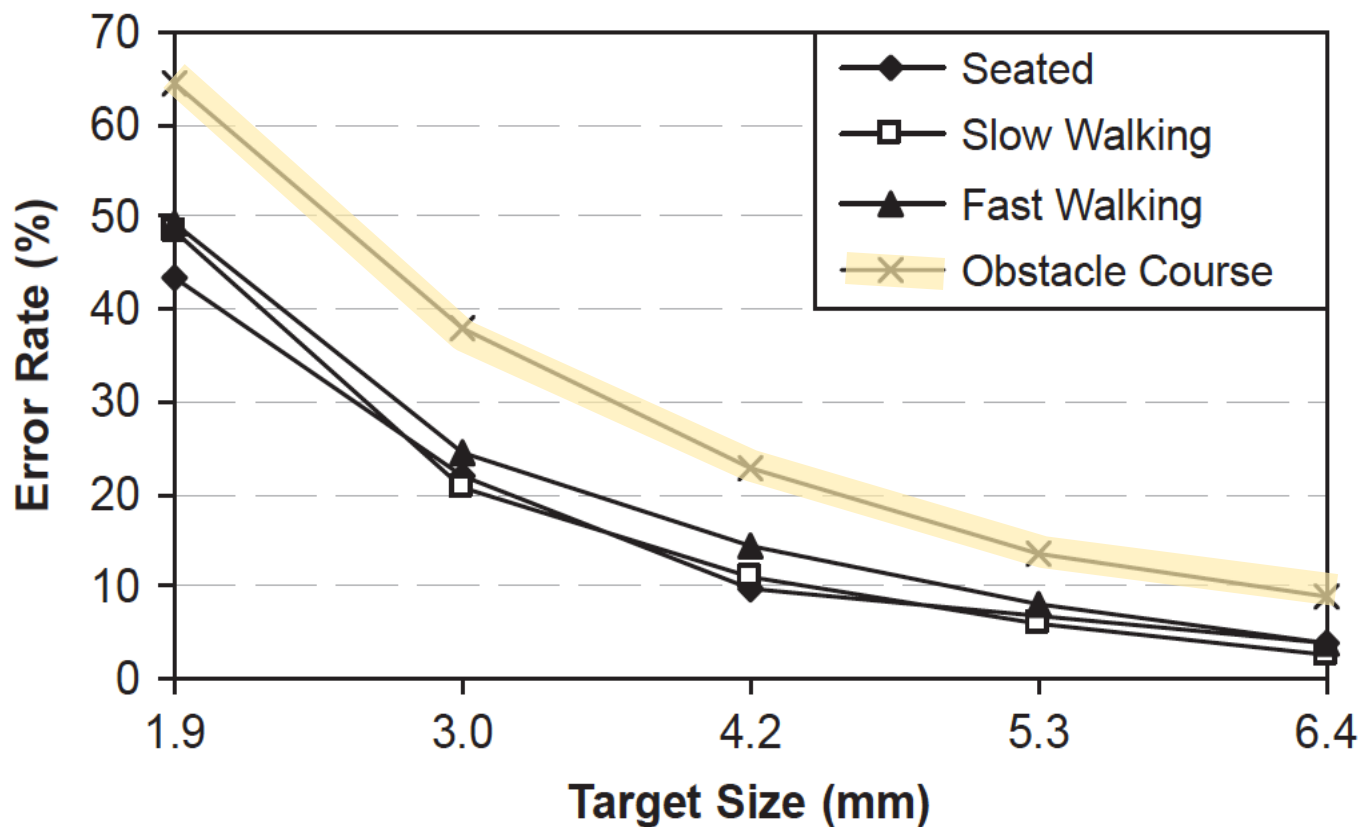
Experiment to understand the impact of using a phone while walking

- Tap on targets when:
  - Seated
  - Walking on treadmill (slow/fast)
  - Walking through Obstacle course
- Measures
  - **Time** to complete task
  - **Errors** when selecting targets



# Input while Walking

	Seated	Slow walking	Fast walking	Obstacle course
Task completion time (s)	457.2 (74.2)	448.1 (94.8)	468.3 (82.1)	526.9 (73.7)





# Reading and Thinking while Walking

Read short text on mobile device and answer questions when **seated** or when **walking through obstacle course**

**Reading Comprehension**

Ratatouille is a dish that has grown in popularity worldwide over the last few years. It features eggplant, zucchini, tomato, peppers, and garlic, chopped, mixed together, sautéed briefly, and finally, cooked slowly over low heat. As the vegetables cook slowly, they make their own broth, which may be extended with a little tomato paste. The name ratatouille comes from

Done

**Question (2/2)**

Ratatouille can best be described as a

- ☐ French pastry
- ☐ sauce to put over vegetables
- ☒ pasta dish extended with tomato paste
- ☐ vegetable stew

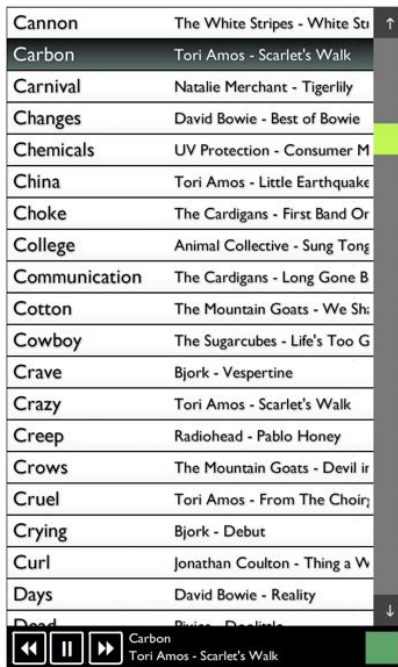
Submit

## Results

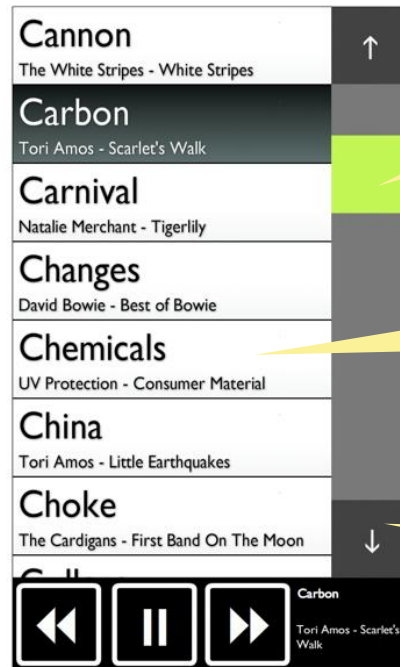
- Answered questions at same speed when seated or walking
- Read more slowly when walking
- Answered more questions incorrectly when walking

# Interface Adaptation when Walking

- To address situational walking impairment
  - Reduced dexterity and motor control
  - Reduced cognitive ability



sitting interface



walking interface

Varied saliency of visual elements helps address limited attention

Larger visual cues address reduced reading ability

Larger widget targets address impaired dexterity

# Many People have chronic or long term impairments

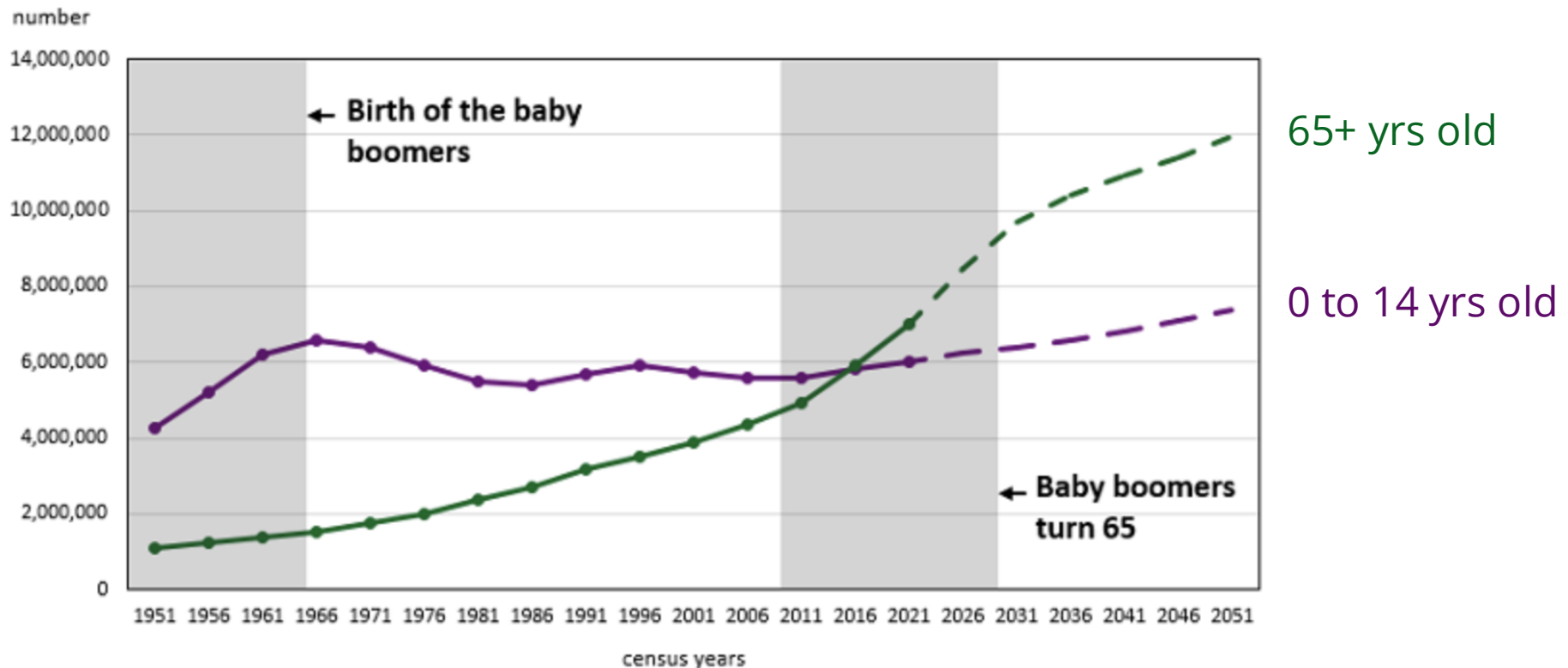
- Visual
  - 1 in 100 people have a significant visual disability
  - 1 in 475 people are legally blind
  - 1 in 2000 people are totally blind
- Hearing
  - 1 in 10 people have a significant hearing impairment
  - 1 in 125 people are deaf
- Motor
  - 1 in 250 people use a wheelchair
- Cognitive
  - 3 in 100\* people have a significant cognitive disability

10 to 20% of population estimated to have a long-term disability  
(3 to 6 million people in Canada)

# Age-Related Impairments

- Reduced motor coordination (fine/gross motor skills)
- Visual and hearing impairments
- Cognitive effects like loss of memory

~25% of Canadians will be over 65 by 2030 (only ~10% in 1991)




# Modern OS Support for Accessibility

- Control cursor from keyboard (motor)
- Adjust acceleration, tracking, precision (motor)
- Speech dictation (visual/motor)
- Magnify portions of the screen, adjust element sizes or font-size, provide full voice dictation (visual)
- Captions / subtitles (audial)


macOS  
accessibility  
features


## Accessibility

### Vision

 VoiceOver

 Zoom

 Display

 Spoken Content

 Descriptions


### Hearing


 Audio


 RTT


 Captions

### Motor

 Voice Control

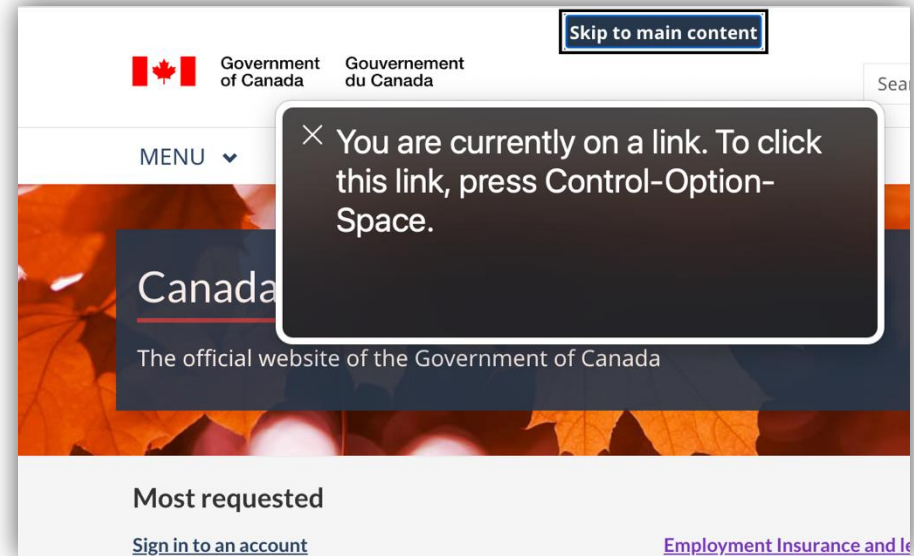
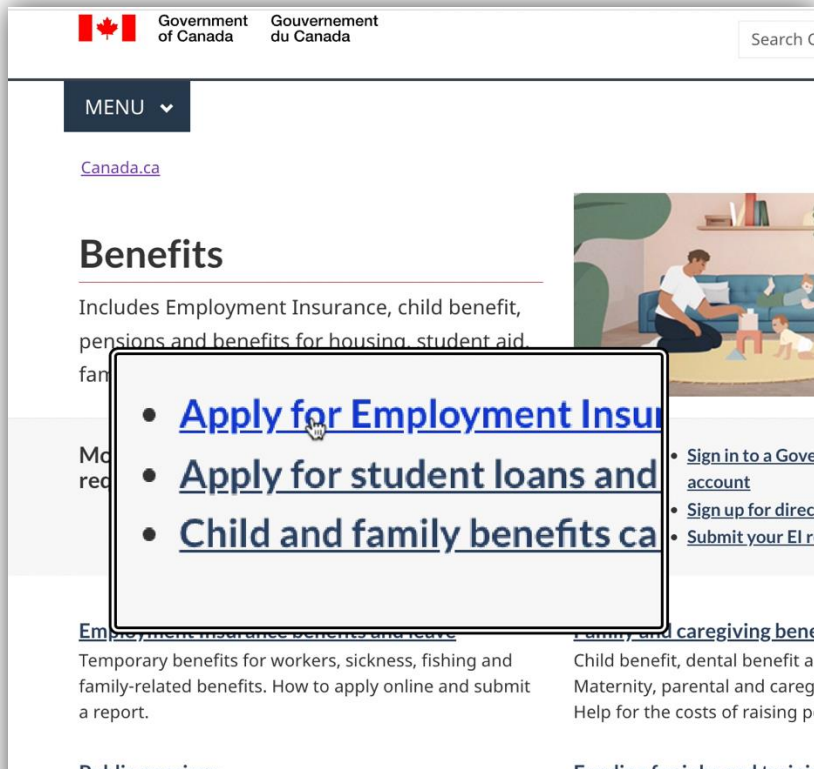
 Keyboard

 Pointer Control

 Switch Control

# Interface Enhancements for Visual Impairments

- zoom screen or specific area, increase font size
- high contrast colours, dark mode, remove animations
- screen reader, voice input
- real world magnifier





# **BLIND PERSON USING A COMPUTER?**



How A Blind Person Uses A Computer

- <https://youtu.be/UzffnbBex6c>



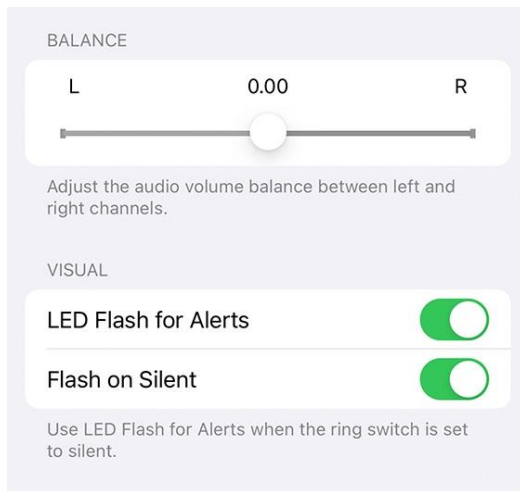
Seeing AI: Making the visual world more accessible

- <https://youtu.be/DybczED-GKE>

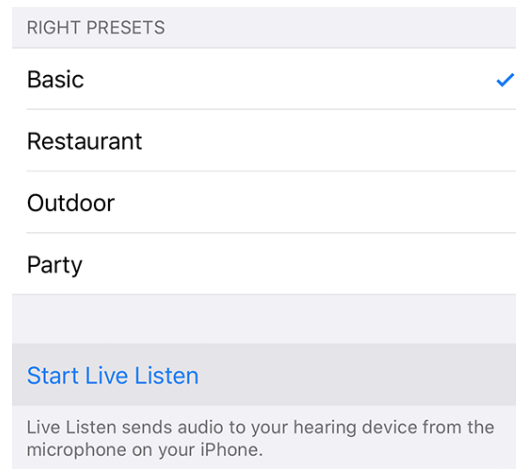


# Enhancements for Hearing Impairments

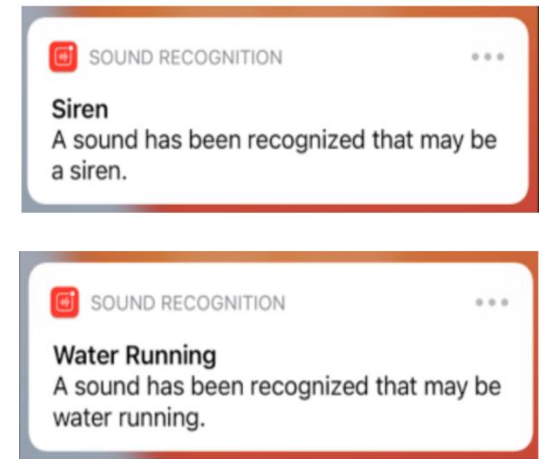
- Show audio alerts visually (e.g. vibrate, flashlight alarm)
- Realtime audio processing to filter background noise and amplify the voice of another person
- Monitor audio for certain sounds and send alert (e.g. baby crying)



**IOS Sensory Alerts**



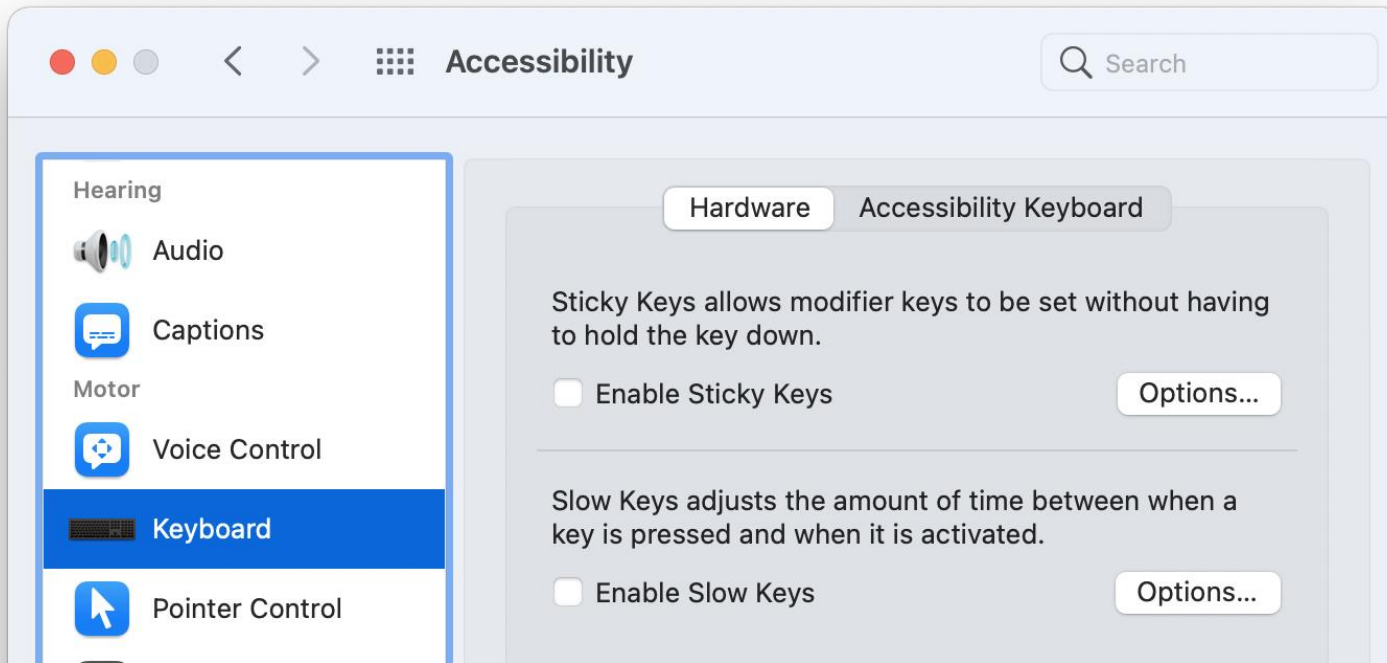
**IOS Live Listen**



**IOS Sound Recognition**

# Enhancements for Motor Impairments

- Sticky keys, slow keys, and filter keys
- Reduce key repeat rate
- Eye tracking
- Voice input
- Physical switches and "puffers"
- Brain-computer interfaces (BCI)



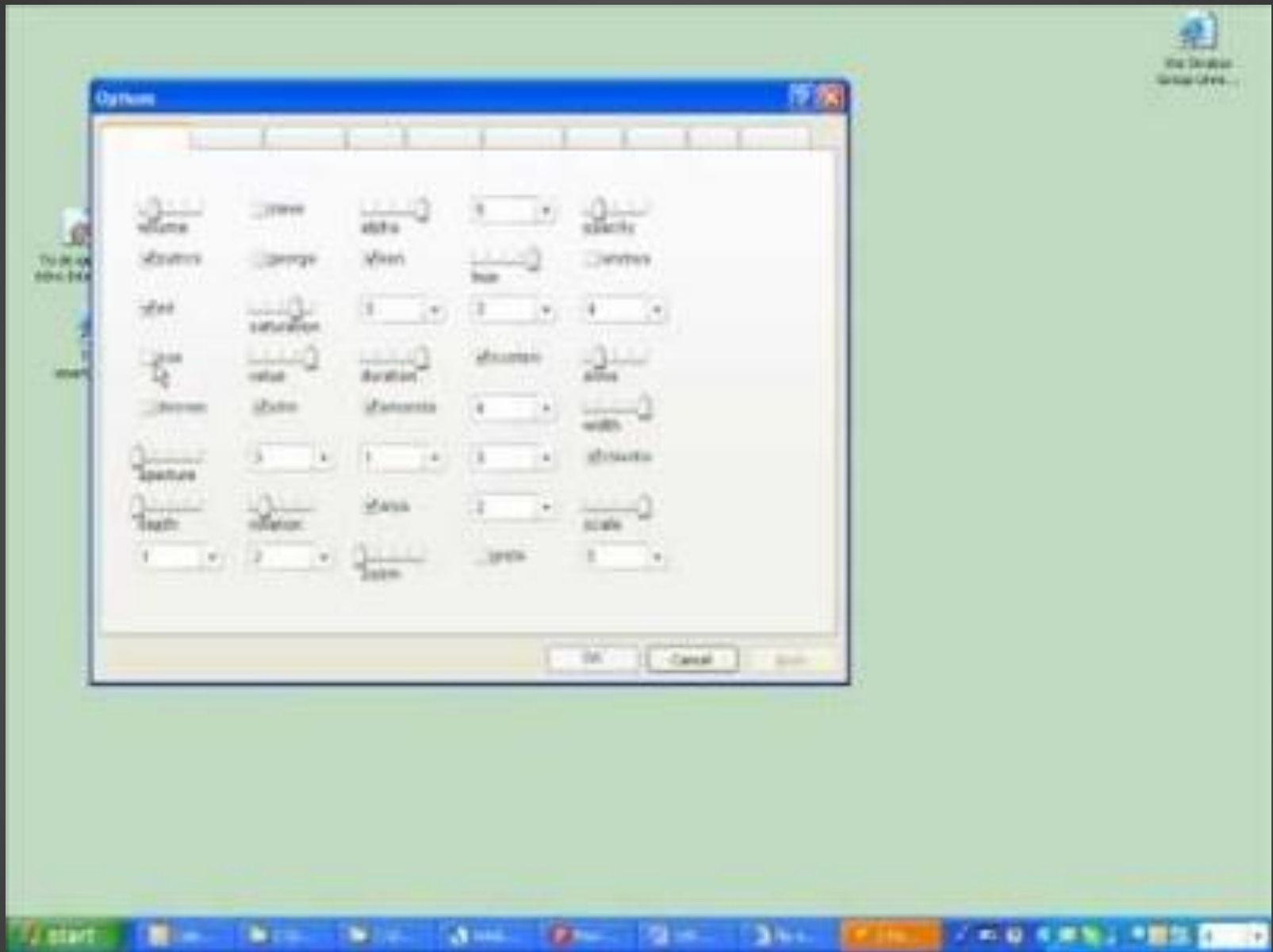


RockyNoHands: The Gamer Who Can Beat You With His Mouth

- <https://youtu.be/ZMvikz2cA-8?si=T2UYvMuBZApsdLB9&t=7>

# Enhancements for Cognitive Impairments

- word prediction, grammar and spelling check
- text-to-speech
- augmenting text with icons and pictures
- "slow down interface"
  - avoid sudden state changes
  - reduce or remove unnecessary animations (esp. flickering)
  - eliminate time sensitive actions



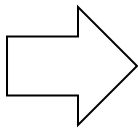
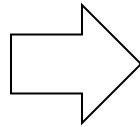
Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects

- <https://youtu.be/oQPTiqMGd60>



# The “Curb Cut” Effect

- Laws and programs designed to benefit vulnerable groups, such as people with a disability, often end up benefiting all of society



# Video Closed-Captioning

- Captions benefit more than people with hearing impairments
  - enables watching TV in silence (while someone else is sleeping) or in noisy environments (like sports bar)
  - helps children learn to read or adults to learn a second language
  - native language captions enable viewers to enjoy foreign language films and television (i.e. "sub-titles")
  - source for searching and data mining video content



# Legal Obligations in Canada

- The Accessible Canada Act (since 2019)
  - government and federally regulated organizations (e.g. transportation, broadcasting, telecommunications, financial)
  - expected to use Web Content Accessibility Guidelines (WCAG)
  - fines up to \$250K
- Accessibility for Ontarians with Disabilities Act (since 2005)
  - applies to all Ontario government websites
  - applies to Ontario public and private entities (50+ employees)
  - must adhere to Web Content Accessibility Guidelines (WCAG) 2.0
  - fines up to \$100K



# Legal Obligations Elsewhere

- United States Disabilities and Rehabilitation Act "Section 508"
  - any organization doing business with federal agency or receives federal funding
- Americans with Disabilities Act (ADA)
  - non-profits, business, local and state governments
- European Union Web Accessibility Directive
  - all government websites
  - any organizations financed through public contracts

POLICY —

# Lawsuit over web site accessibility for the blind becomes class action

A federal judge has certified a class action lawsuit against Target after ...

NATE ANDERSON - 10/3/2007, 1:34 PM

A lawsuit brought in 2006 by a [blind student at the University of California-Berkeley](#) has now morphed into a class action case against US retailer Target. A federal judge has just certified a nationwide class in the case, which alleges that Target's web site is not fully accessible to the blind. It's a case that could help establish the ways in which the Americans With Disabilities Act applies to the Internet, and it has already generated a ruling that, in California at least, commercial web sites must be accessible.

The case focuses on the alleged lack of descriptive "alt" tags in Target's HTML, making the site difficult to navigate with screen reading software. The use of image maps is also claimed to make the site inaccessible.

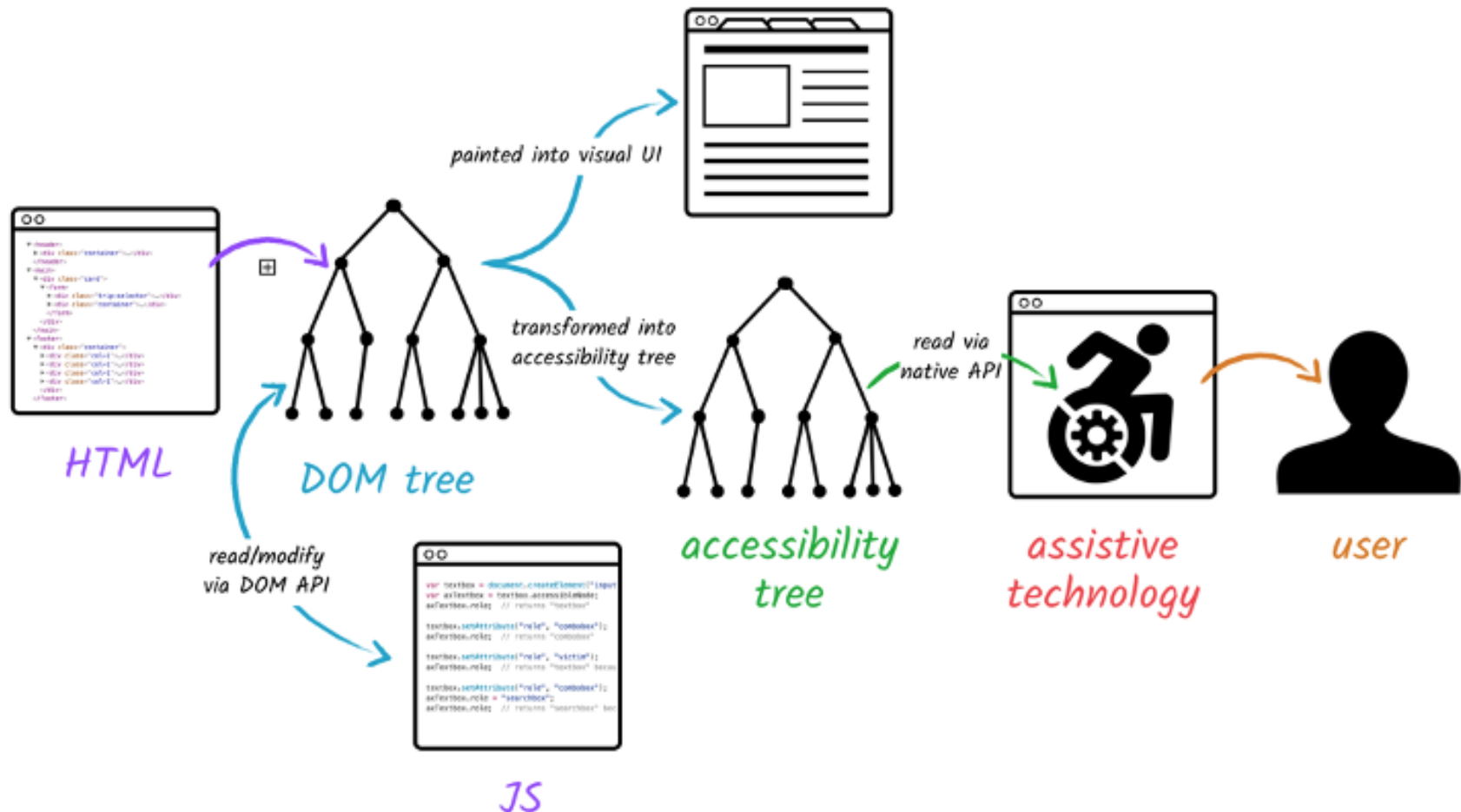
Public locations in the real world have long been required to abide by the ADA, but the law was written in the days before the Web, and it remains unclear how it should be applied to web sites. One of the lawyers from Disability Rights Advocates, which is handling the case, sees inaccessibility as a simple issue of discrimination, online or off.

"Target Corporation has led a battle against blind consumers in a key area of modern life: the Internet economy," said Larry Paradis in a statement after the ruling. "The court's decision today makes clear that

# **Implementing Accessible Interfaces**

# Accessibility Tree

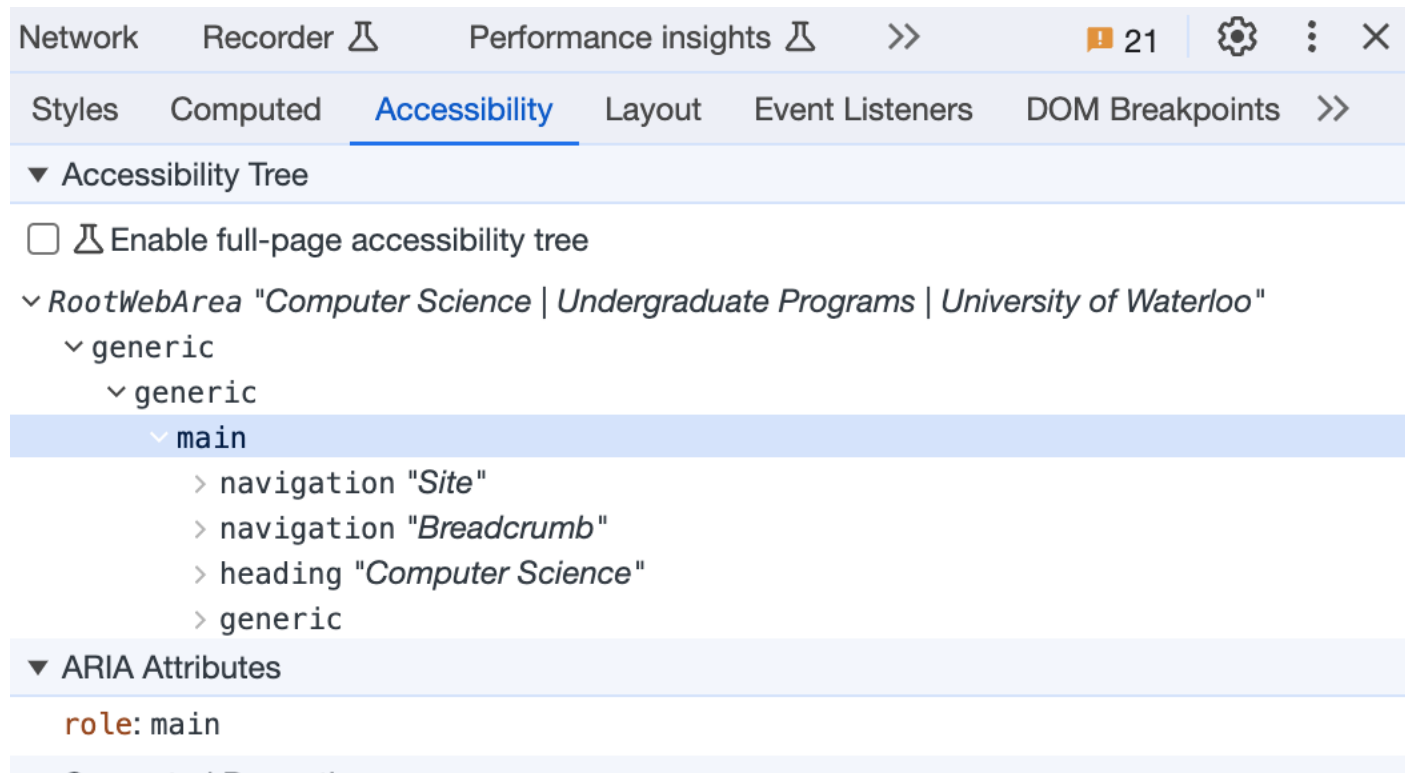
- The browser generates an accessibility tree from the DOM with accessibility-related information for most HTML elements
  - name, description, role, state



# DevTools Accessibility Tab

- CMD-SHIFT-P and search for "Accessibility"
- Enable Accessibility tab
  - can view accessibility tree (and other features)

CMD-SHIFT-P is the  
"Command Menu" in  
DevTools



# Implementing for Web Accessibility

Web Content Accessibility Guidelines (WCAG)

<https://www.w3.org/WAI/standards-guidelines/wcag/>

- Include alt text for information images

```
<p>0123 456 7890</p>
```



0123 456 7890

- Use headings, and use them correctly
- Give links unique and descriptive names

## \* Use colour with care

- Use tables for tabular data, not for layout

## \* User ARIA roles and landmarks (but only when necessary)

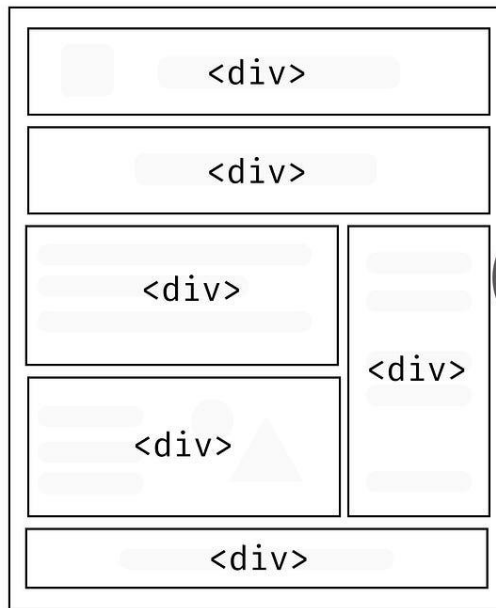
- Make dynamic content accessible
- Make all content accessible using keyboard too

## \* Design forms for accessibility

<https://webaccess.berkeley.edu/resources/tips/web-accessibility>  
<https://www.w3.org/WAI/tutorials/images/informative/>

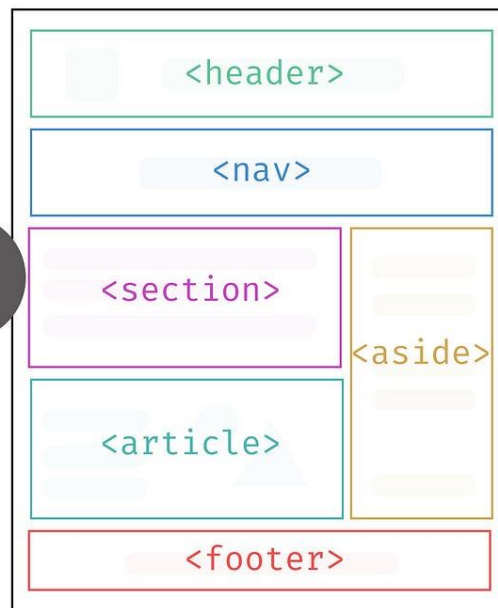
# Semantic HTML

- Also called "POSH, or Plain Old Semantic HTML"
- Don't use `<div>` and `<span>` for everything
- A semantic elements clearly describes the content meaning



**Ambiguous Sections**

VS



**Clear Sections**

`<article>`  
`<aside>`  
`<details>`  
`<figcaption>`  
`<figure>`  
`<footer>`  
`<header>`  
`<main>`  
`<mark>`  
`<nav>`  
`<section>`  
`<summary>`  
`<time>`  
...

# html / semantic-good.html

```
<h1>Cool Blog 1999</h1>
<p>A blog about my adventures.</p>
<h2>A trip to the beach</h2>
<p>Yesterday, I went to the beach to see the ocean.</p>
<p>I brought these three things:</p>
<ol>
  <li>A bathing suit</li>
  <li>Sunscreen</li>
  <li>A towel</li>
</ol>
<h3>Swimming in the Ocean</h3>
<p>
  Once I found a good spot. I put on my bathing suit and
  went into
  the water.
</p>
<p>It was cold at first, but I got used to it.
<h3>Relaxing on the Beach</h3>
<p>
  After swimming, I laid on my towel and relaxed. I put on
  sunscreen to protect my skin from the sun.
</p>
```

try navigating page  
with screenreader





# html / semantic-bad.html

```
<div class="site">Cool Blog 1999</div>
```

```
A blog about my adventures. <br />
```

```
<div class="blog-title">A trip to the beach</div>
```

```
Yesterday, I went to the beach to see the ocean. <br /><br />
```

```
I brought these three things: <br /><br />
```

```
<div class="list">
```

```
  1. A bathing suit <br />
```

```
  2. Sunscreen <br />
```

```
  3. A towel
```

```
</div>
```

```
<div class="heading">Swimming in the Ocean</div>
```

```
Once I found a good spot. I put on my bathing  
into
```

```
the water. <br /><br />
```

```
It was cold at first, but I got used to it.
```

```
<div class="heading">Relaxing on the Beach</div>
```

```
After swimming, I laid on my towel and relax  
sunscreen
```

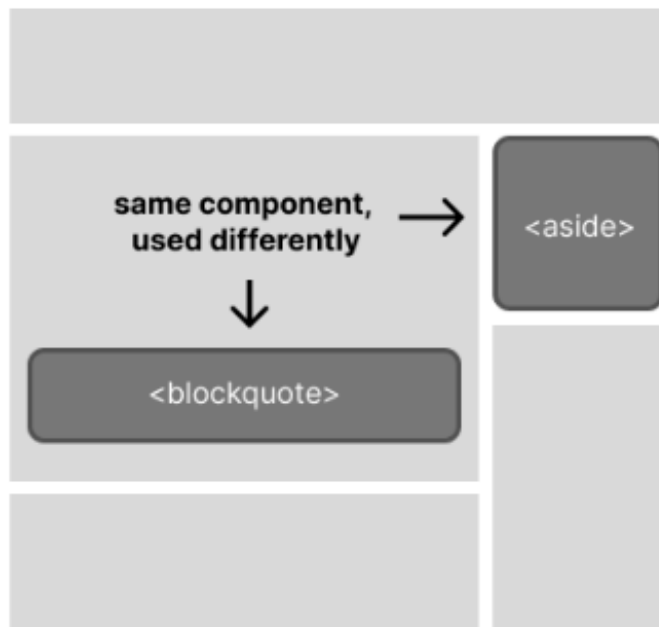
```
to protect my skin from the sun.
```

try navigating page  
with screenreader



# Semantic HTML with Preact Components

- Avoid `<div>` for component root if semantic element possible
- Use `<Fragment>` instead of `<div>` if root has no semantic purpose
- Some components could have different semantic roles
  - Use "as Element" pattern to set component root element ...



## as-element

```
type CalloutProps = {  
  as: any;  
  children: ComponentChildren;  
};
```

```
export default function Callout({  
  as: Element = "div",  
  children,  
}: CalloutProps) {  
  return <Element className="callout">{children}</Element>;  
}
```

// try changing the `as` prop to aside, blockquote, h1, etc.

// then check the rendered HTML

```
render(  
  <Callout as="aside">as-element content</Callout>,  
  document.body  
)
```

# Skip Link

- Add link at top of page so screen readers can "skip to content"

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

...

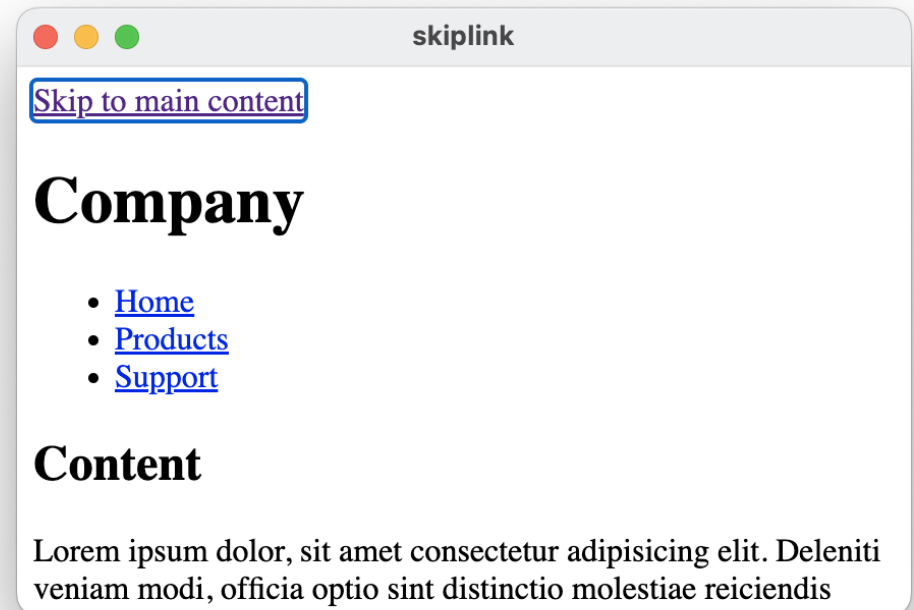
```
</header>
```

```
<section id="maincontent">
```

- CSS to hide until focused

```
.skip {  
  position: absolute;  
  left: -9999px;  
}
```

```
.skip:focus {  
  position: static;  
}
```



# ARIA Attributes

- ARIA (Accessible Rich Internet Applications)
  - a specification to add semantics to elements
- ARIA Roles
  - Use ONLY if not possible to use an HTML semantic element
  - examples of ARIA roles without HTML equivalent:  
toolbar, tooltip, feed, math, presentation, note
- ARIA States and Properties
  - Use ONLY for attributes not supported on HTML element
  - examples:  
aria-required, aria-checked, aria-disabled
- ARIA attributes only change the accessibility tree

# forms

- Accessible forms

```
<label for="name">Name:</label>
```

```
<input id="name" type="text" autocomplete="name" />
```

```
<fieldset>
```

```
  <legend>Choose a shipping method:</legend>
```

```
  <input id="overnight" type="radio"
        name="shipping" value="overnight" />
```

```
  <label for="overnight">Overnight</label>
```

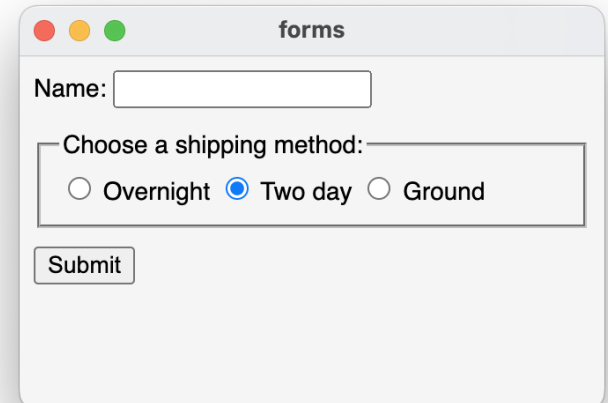
```
  <input id="twoday" type="radio"
        name="shipping" value="twoday" />
```

```
  <label for="twoday">Two day</label>
```

```
  <input id="ground" type="radio"
        name="shipping" value="ground" />
```

```
  <label for="ground">Ground</label>
```

```
</fieldset>
```



The screenshot shows a browser window with the title "forms". Inside the window, there is a form with the following elements: a text input field labeled "Name:", a legend box containing the text "Choose a shipping method:" and three radio buttons labeled "Overnight", "Two day", and "Ground". The "Two day" radio button is selected. Below the legend box is a "Submit" button.

# Colour Perception

- Human ability to discriminate colours
  - colour vision deficiency (“colour blindness”)
  - **context**
- It’s harder to tell two colours apart when
  - the colours are pale
  - the object is small or thin
  - the colour patches are far apart



**FIGURE 5.4**

Factors affecting the ability to distinguish colors: (A) paleness, (B) size, (C) separation.

# Colour Contrast

Ratio of perceived luminance for two colours (e.g. foreground text and background), examples:

1.0 ( on white)	8.6 (blue on white)
1.4 (green on white)	21.0 (black on white)
4.0 (red on white)	

- WCAG Colour Contrast Guidelines  
(most cases, some specific contrast for logos, large text, etc.)

- **Minimum (Level AA)** at least 4.5

Gray (#767676) on white

Purple (#CC21CC) white

Blue (#000063) on gray (#808080)

Red (#E60000) on yellow (#FFFF47)

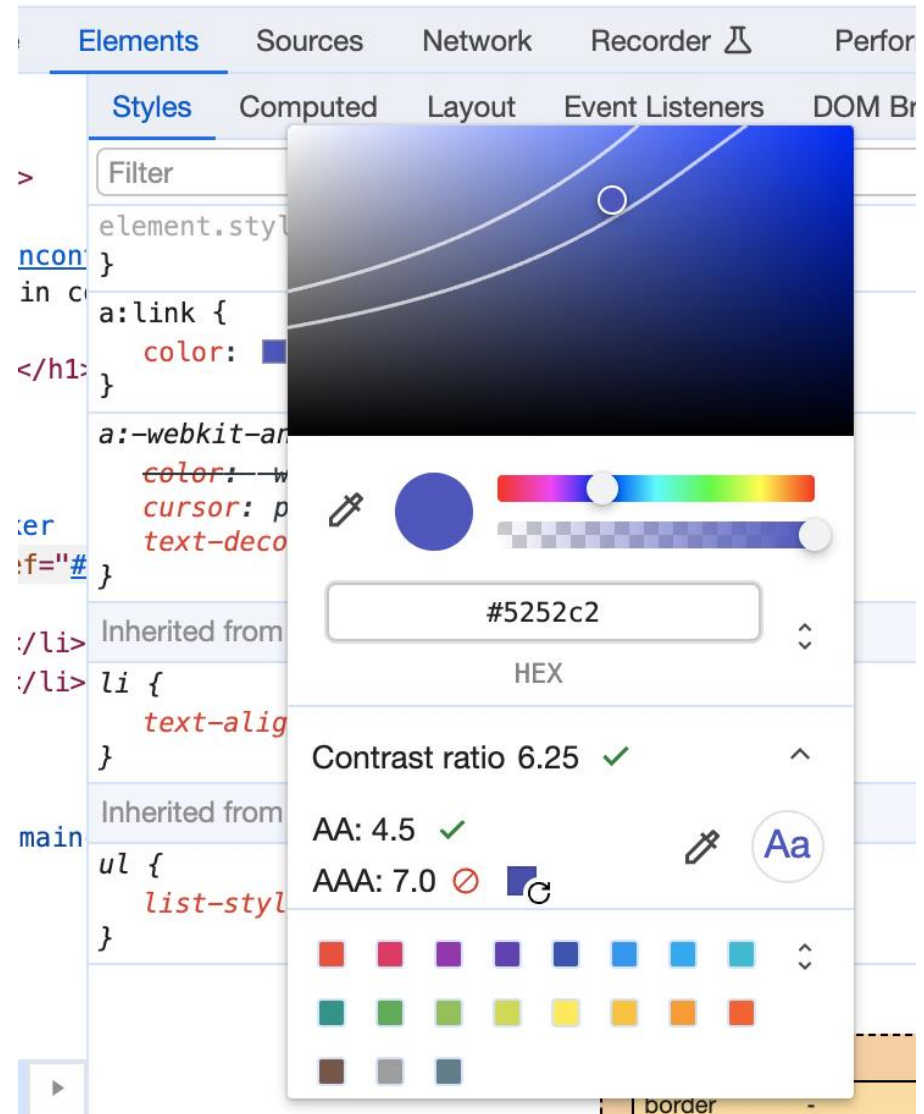
examples with  
contrast 4.5

- **Enhanced (Level AAA)** at least 7.0



# DevTools Colour Contrast Tool

- Find contrast value in DevTools
  - Inspect element
  - Click style colour swatch
  - Click "Contrast ratio X.XX" to display contrast thresholdlines in colourspace



# Basic A11y Testing Methods

- Disconnect your mouse and try to use your app
  1. Tab and Shift+Tab to focus element
  2. Enter to activate element
  3. Arrow keys when needed (e.g. menu, dropdown)
- Test with a screenreader
  - e.g. VoiceOver on MacOS, Narrator in Windows
- A11y linters/checkers

The screenshot displays the Wave WebAIM accessibility evaluation tool interface. The browser address bar shows the URL `https://uwaterloo.ca`. The tool's sidebar on the left provides a summary of findings:

- Errors:** 5 (indicated by a red 'x' icon)
- Contrast Errors:** 7 (indicated by two red circles)
- Alerts:** 6 (indicated by a yellow triangle icon)
- Features:** 22 (indicated by a green checkmark icon)
- Structural Elements:** 26 (indicated by a blue triangle icon)
- ARIA:** 15 (indicated by a purple cube icon)

Below the summary, there is a 'View details' button. The main content area shows a preview of the University of Waterloo website with a yellow cookie consent popup overlaid. The popup contains the text: "We use cookies on this site to enhance your user experience". It also includes a link to "Accept all" and a button labeled "Accept all". The tool highlights several accessibility issues on the page, such as the missing `tabindex` attribute on the "Accept all" button and the missing `role` attribute on the document element. The tool's interface includes a top navigation bar with icons for various accessibility features and a bottom navigation bar with a "Code" button.

## Accessibility Testing Tools

- <http://wave.webaim.org/report#/uwaterloo.ca>