Session 1

The Accessibility tree and ARIA

Slide instructions

`SPACEBAR` to **move forward** through slides.

`SHIFT` & `SPACEBAR` to **move backwards** through slides.

`LEFT ARROW` & `RIGHT ARROW` to **move through sections**.

`ESC` to **see overview** and `ESC` again to exit.

`F` to **enter presentation mode** and `ESC` to exit.

Introduction

A developer's role in accessibility is to make **flexible, transformable** websites.

Flexible sites allow people to use any device, with any setting they need.

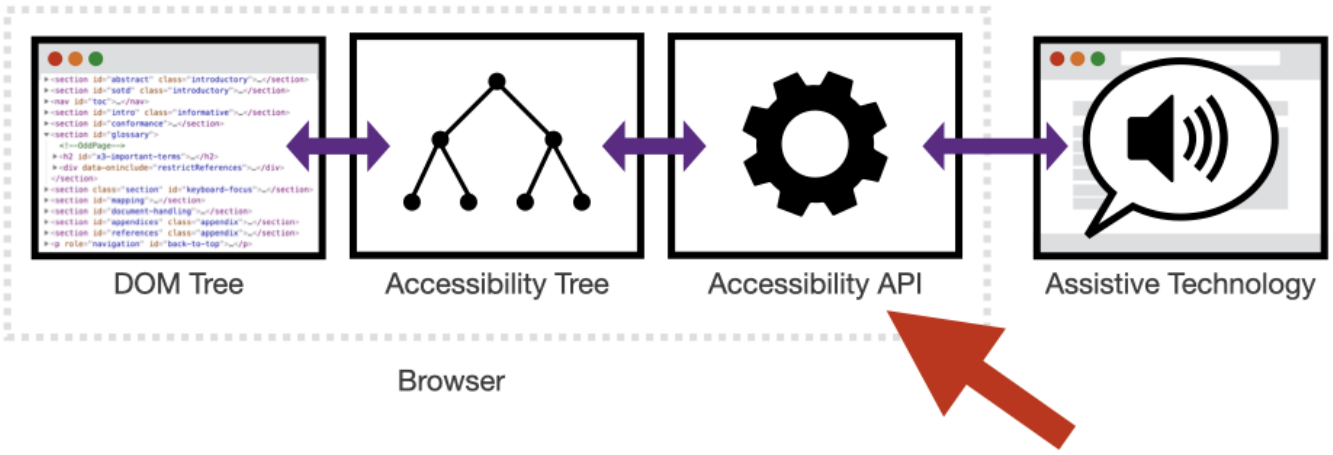
We can provide **sensible defaults** without blocking anyones less common preferences.

Assistive technology can **transform** our site into a format that works for a specific user group.

We need to make sure all the information about a site is **available in the code**.

Accessibility APIs

Accessibility APIs **communicate information about the user interface** from the browser to the assistive technology.



There have been many Accessibility APIs released over the years, all **building on and improving previous versions**.

Year	API	For
1997	Microsoft Active Accessibility (MSAA)	Windows 95
1998	IAccessible	Cross platform
2001	Assistive Technology Service Provider Interface (AT-SPI)	Linux OS
2002	NSAccessibility	Mac OS 10.2
2007	User Interface Automation (UIA)	Windows 7
2007	IAccessible2	Windows and Linux
2009	UI Accessibility API	iOS 3

Browsers generally support one or more of the available accessibility APIs **for the platform they're running on**.

Windows:

- Firefox, Chrome and Opera support: *MSAA/IAccessible* and *IAccessible2*.
- Edge supports *MSAA/IAccessible* and *UIAExpress*.

OSX:

- Safari, Firefox and Chrome support: *NSAccessibility*.

iOS:

- Safari, Firefox and Chrome support *UIAccessibility*.

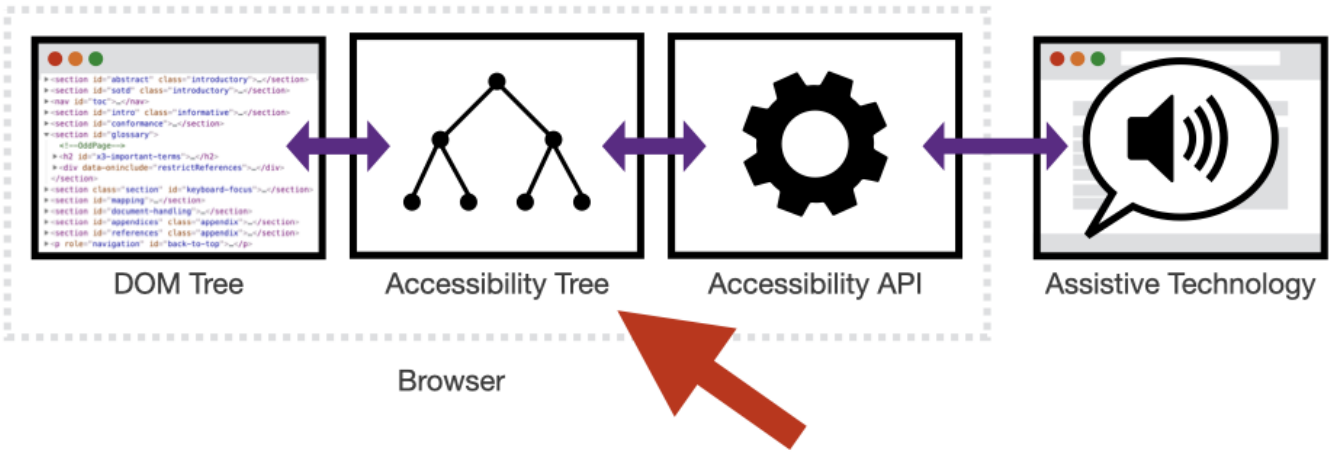
Each browser may have a different Accessibility API, which means **you may get slightly different results**.

When testing new components, you have to test across **a range of different combinations**.

AT	Browsers
Windows: NVDA	Chrome, Firefox, Edge
Windows: JAWS	Chrome, Firefox, Edge
OSX: VoiceOver	Chrome, Firefox, Safari
iOS: VoiceOver	Chrome, Safari
Android: TalkBack	Chrome

Any questions or comments?

Accessibility Tree



The browser uses the DOM, along with further information derived from CSS, to generate an **Accessibility Tree**.

This information is **passed to the relevant accessibility APIs** associated with the browser / operating system.

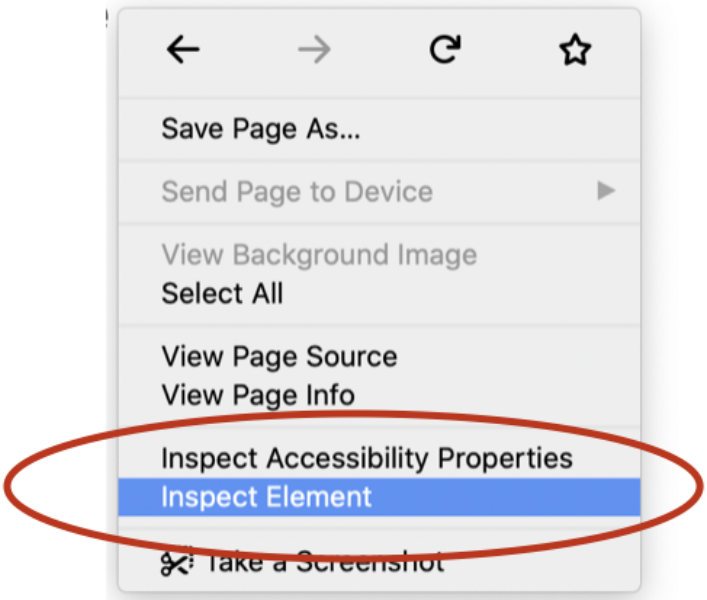
This accessibility tree exposes **information on specific elements** to assistive technologies via the accessibility API.

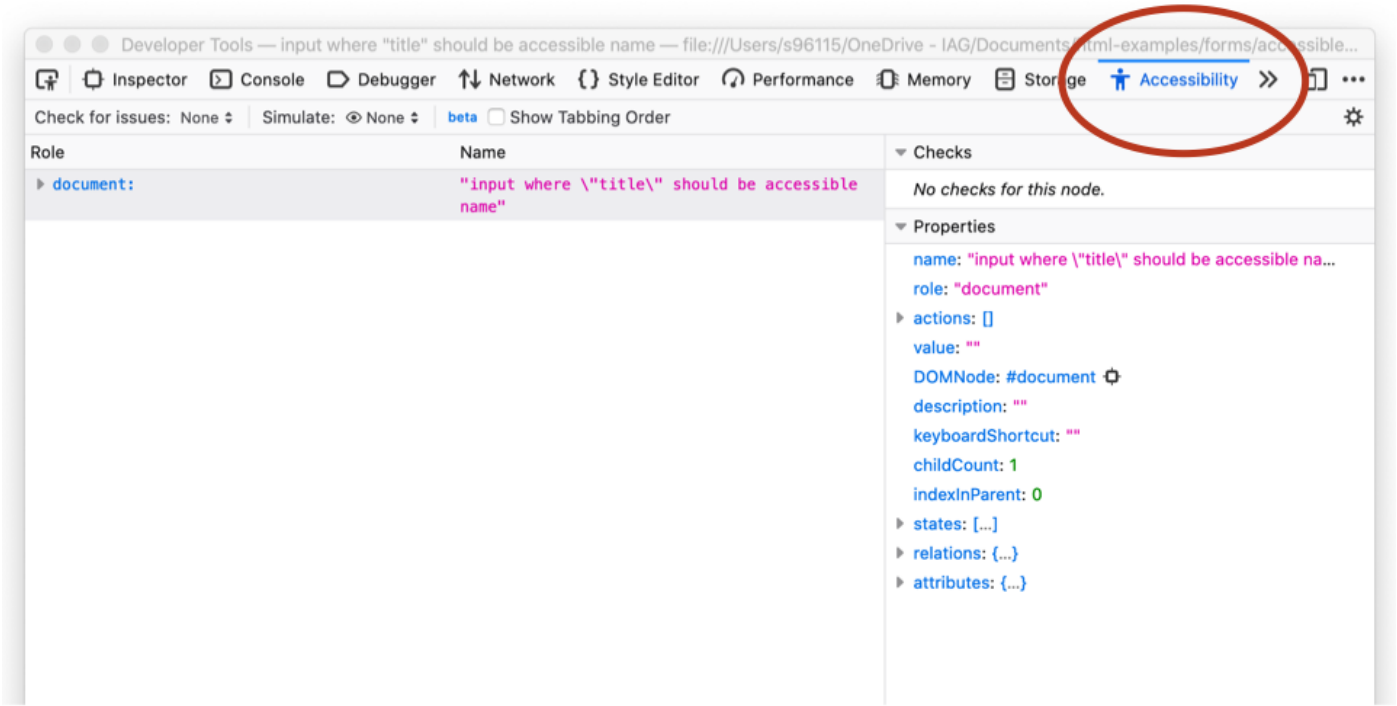
Information includes:

- The **role, name and state** of each object in the content.
- How each object **relates to other objects** in the content.

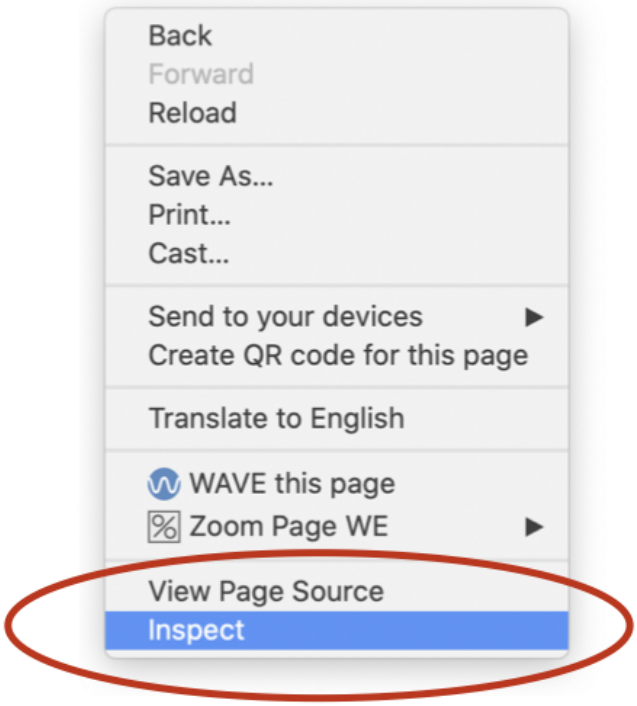
Each browser **will produce a unique accessibility tree** due to their specific combination of source code and accessibility API.

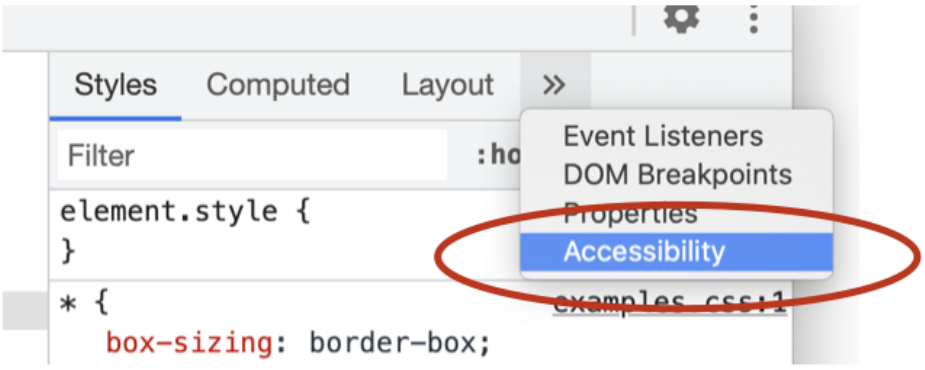
Accessing the Accessibility Tree in Firefox

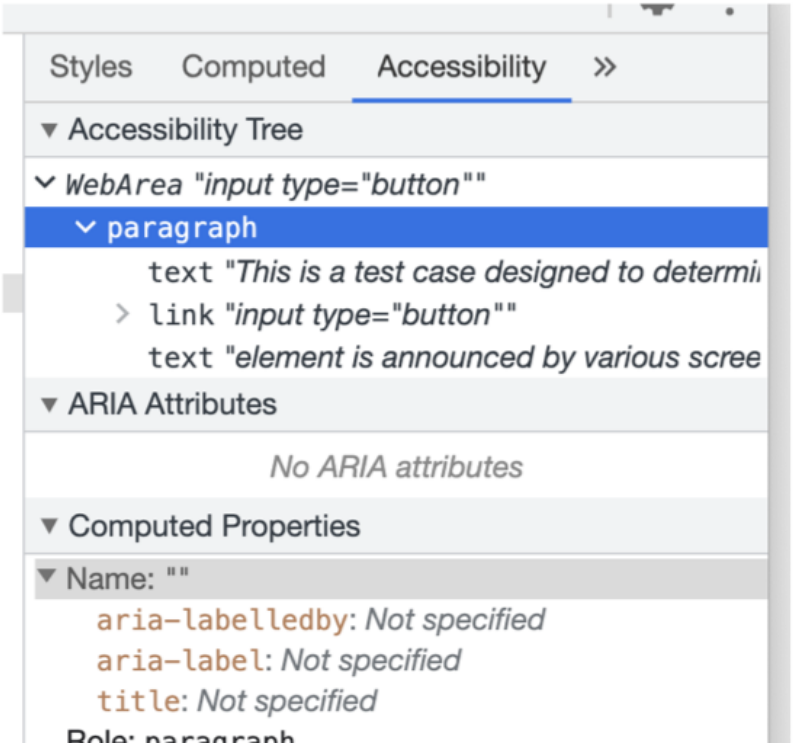


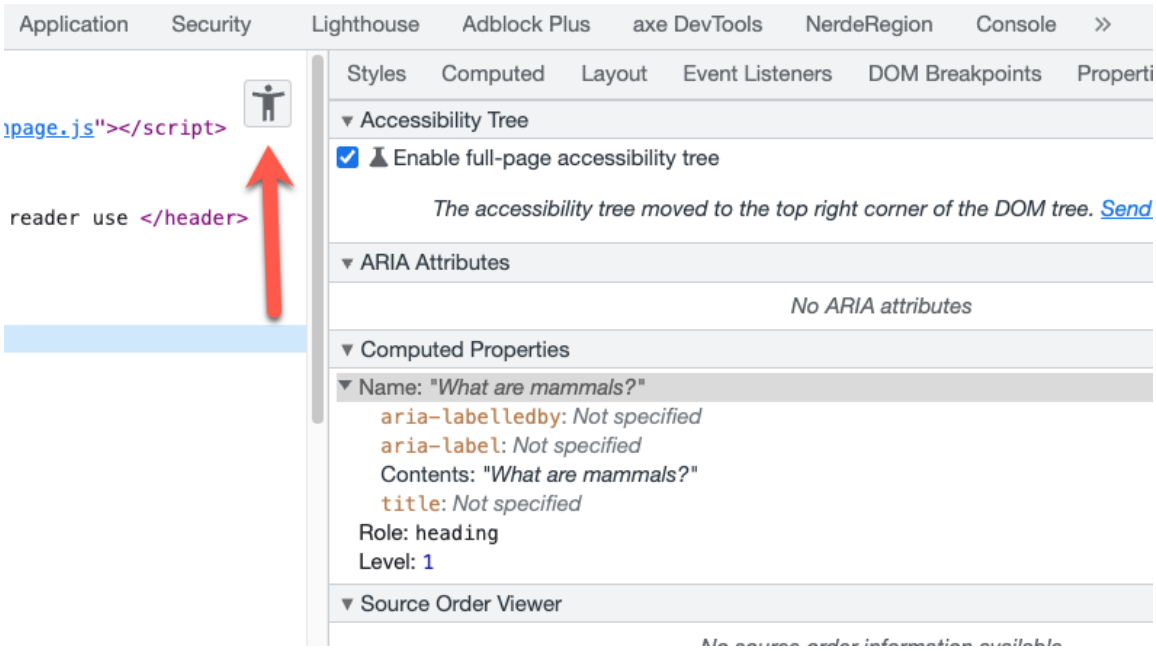


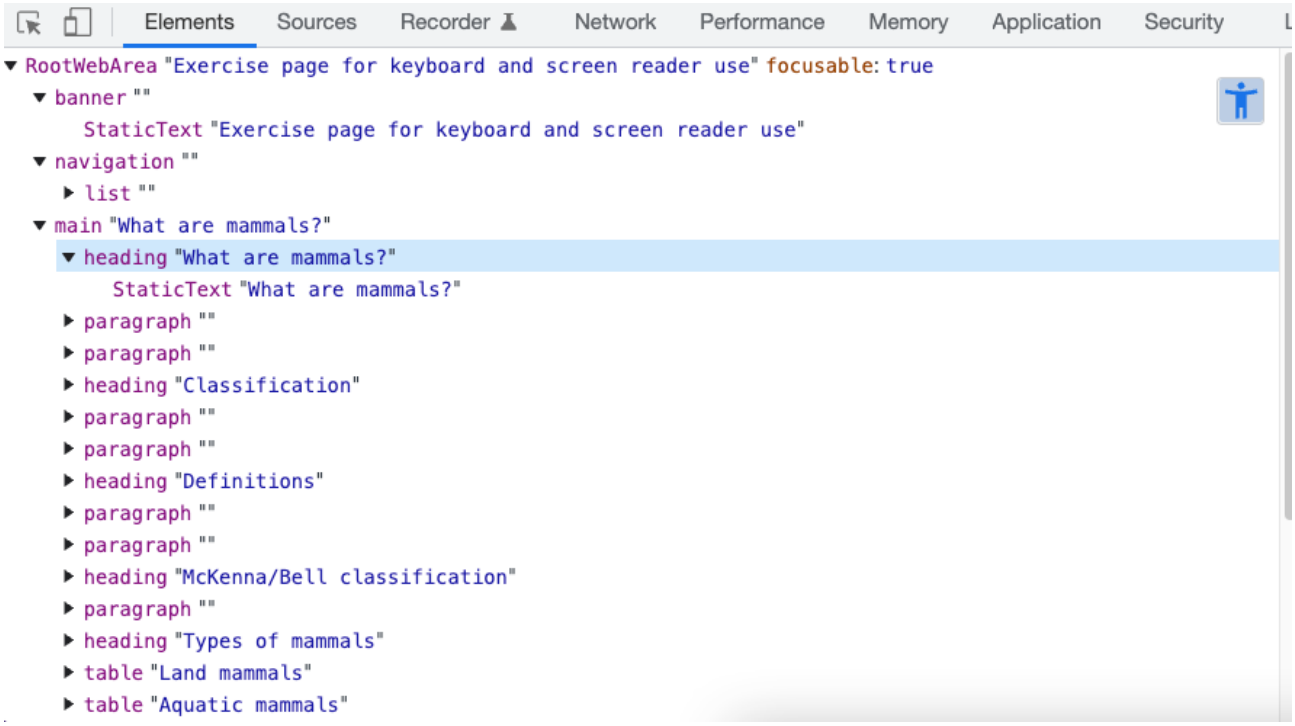
Accessing the Accessibility Tree in Chrome


Back
Forward
Reload
Save As...
Print...
Cast...
Send to your devices
Create QR code for this page
Translate to English
WAVE this page
Zoom Page WE
View Page Source
Inspect


Styles
Computed
Layout
Filter
element.style {
}
* {
box-sizing: border-box;
Event Listeners
DOM Breakpoints
Properties
Accessibility


Styles
Computed
Accessibility
Accessibility Tree
WebArea "input type="button""
paragraph
text "This is a test case designed to determi
link "input type="button""
text "element is announced by various scree
ARIA Attributes
No ARIA attributes
Computed Properties
Name: ""
aria-labelledby: Not specified
aria-label: Not specified
title: Not specified


Application
Security
Lighthouse
Adblock Plus
axe DevTools
NerdeRegion
Console
page.js"></script>
reader use </header>
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Accessibility Tree
Enable full-page accessibility tree
The accessibility tree moved to the top right corner of the DOM tree. Send
ARIA Attributes
No ARIA attributes
Computed Properties
Name: "What are mammals?"
aria-labelledby: Not specified
aria-label: Not specified
Contents: "What are mammals?"
title: Not specified
Role: heading
Level: 1
Source Order Viewer

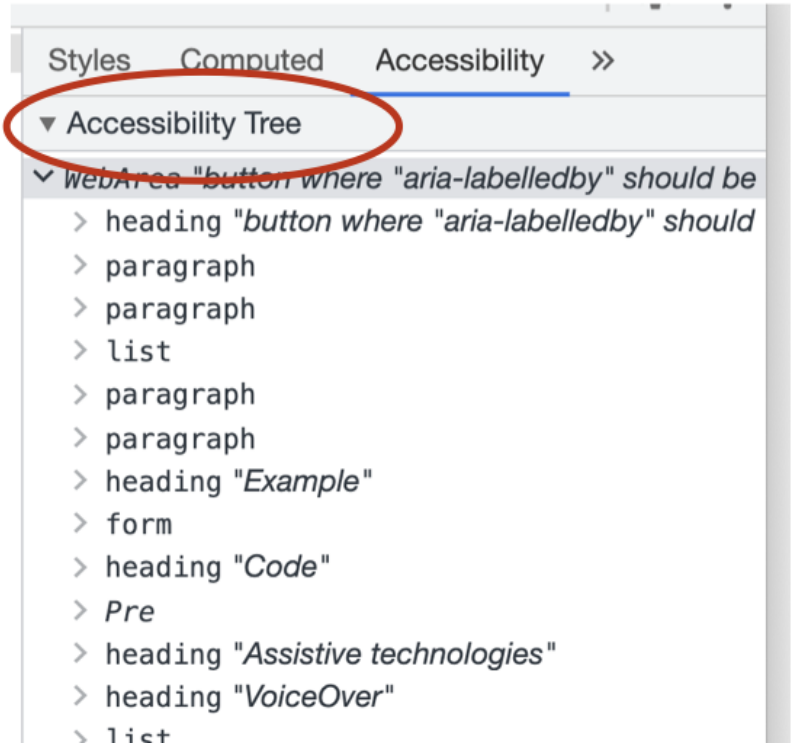


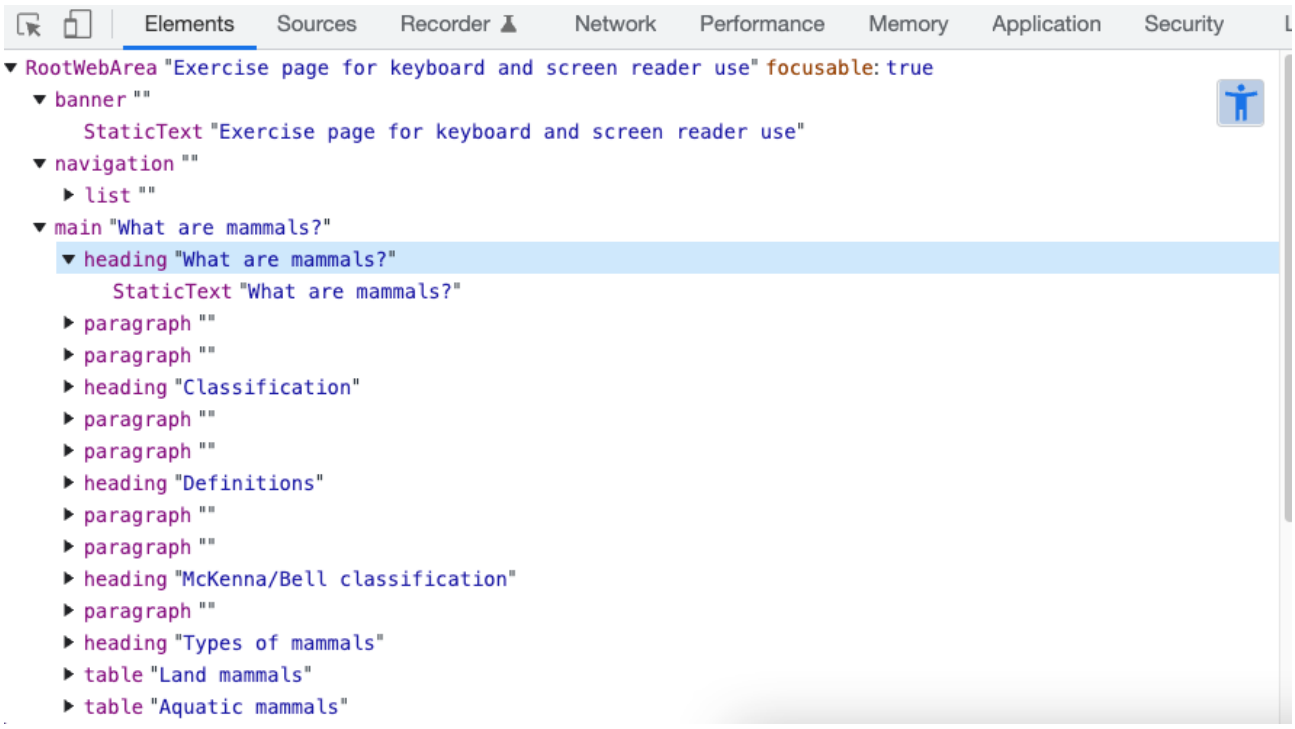
Breaking down Chrome's Accessibility Tree

Chrome's "Accessibility" tab has **three key areas**:

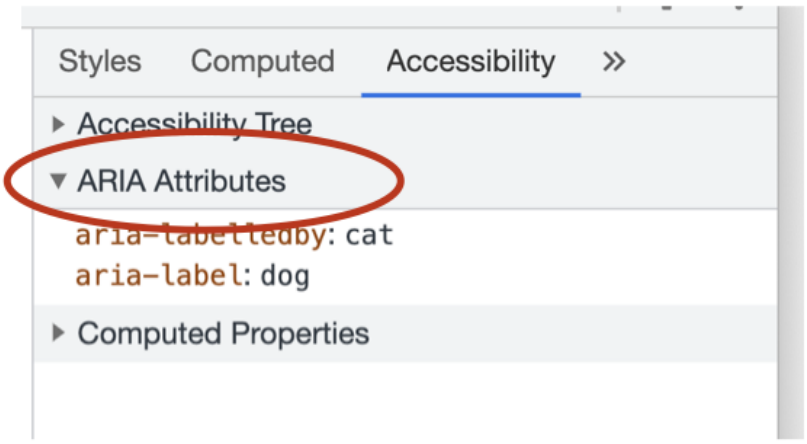
- accessibility tree,
- ARIA attributes, and
- computed properties.

As you inspect an element in the DOM, the relationships between each element are presented in the **Accessibility Tree tab**.

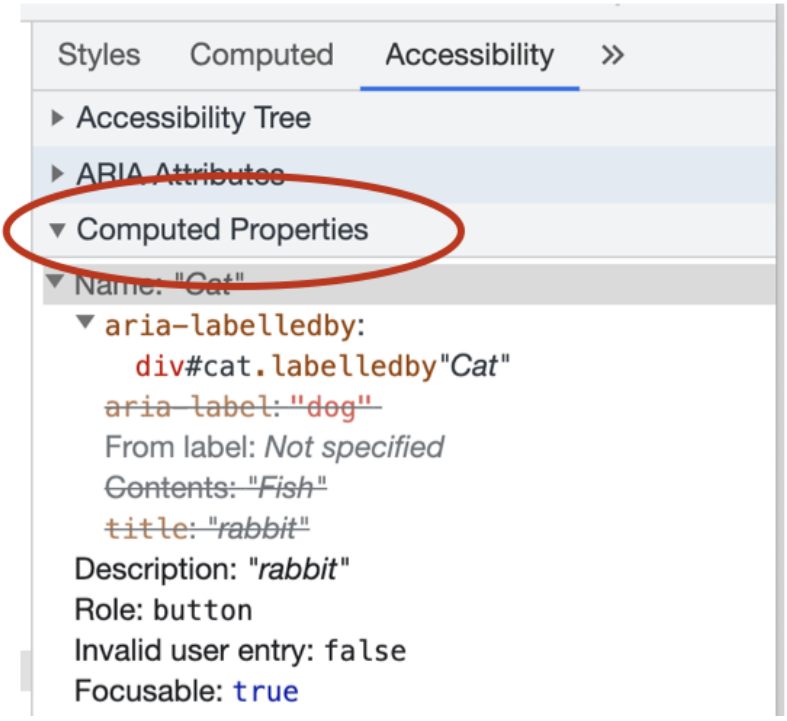

Styles
Computed
Accessibility
Accessibility Tree
WebArea "button where "aria-labelledby" should be
heading "button where "aria-labelledby" should
paragraph
paragraph
list
paragraph
paragraph
heading "Example"
form
heading "Code"
Pre
heading "Assistive technologies"
heading "VoiceOver"


Elements
Sources
Recorder
Network
Performance
Memory
Application
Security
RootWebArea "Exercise page for keyboard and screen reader use" focusable: true
banner ""
StaticText "Exercise page for keyboard and screen reader use"
navigation ""
list ""
main "What are mammals?"
heading "What are mammals?"
StaticText "What are mammals?"
paragraph ""
paragraph ""
heading "Classification"
paragraph ""
paragraph ""
heading "Definitions"
paragraph ""
paragraph ""
heading "McKenna/Bell classification"
paragraph ""
heading "Types of mammals"
table "Land mammals"
table "Aquatic mammals"

As you inspect an element in the DOM, any relevant ARIA attributes are presented in the **ARIA Attributes tab**.



As you inspect an element in the DOM, all the computed properties of that element are presented in the **Computed Properties tab**.



Any questions or comments?

Exercise: Viewing the Accessibility Tree

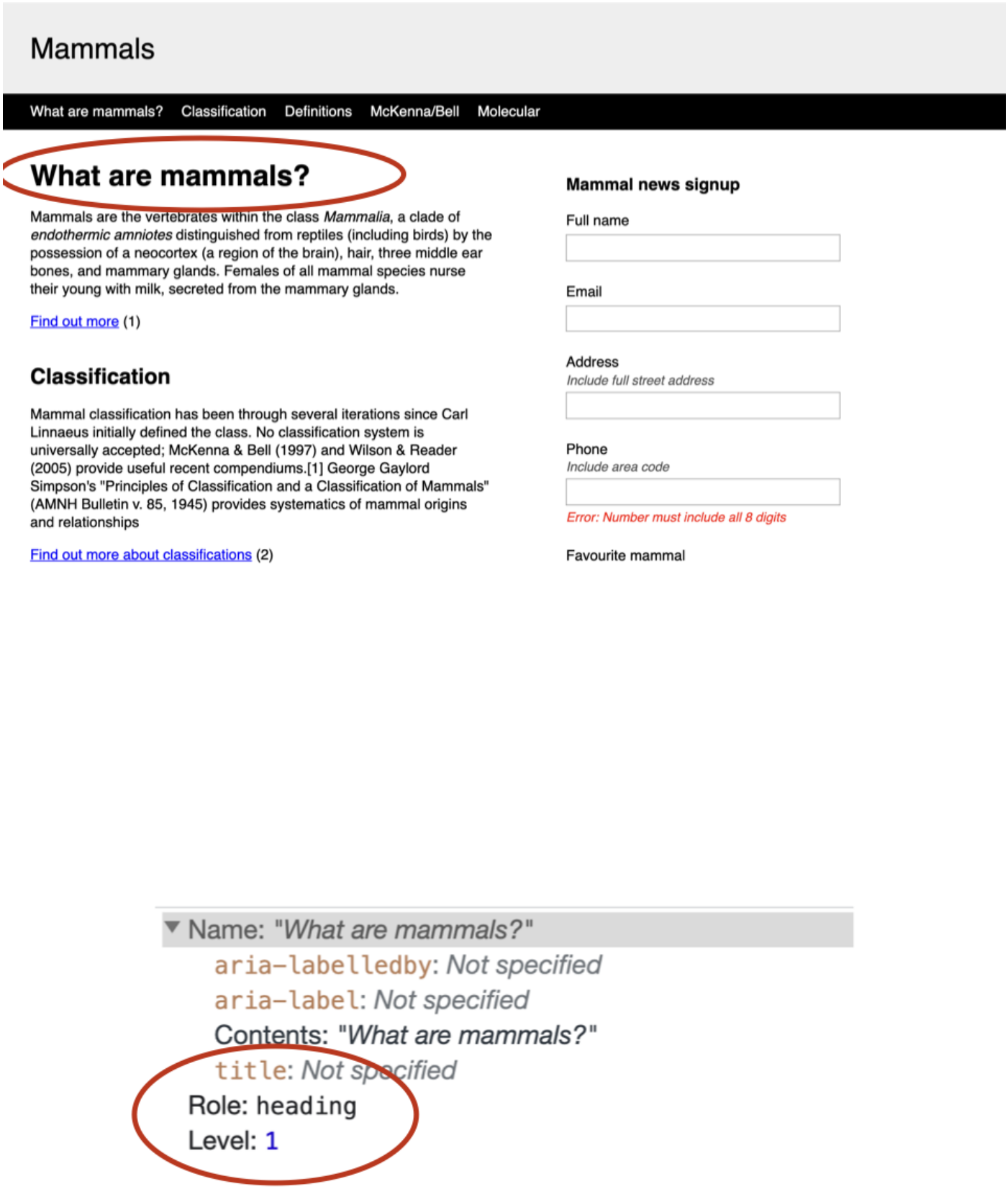
Accessing the exercise:

https://russmaxdesign.github.io/exercise/.

Browsers use the DOM tree to create a second tree, called the **accessibility tree**.

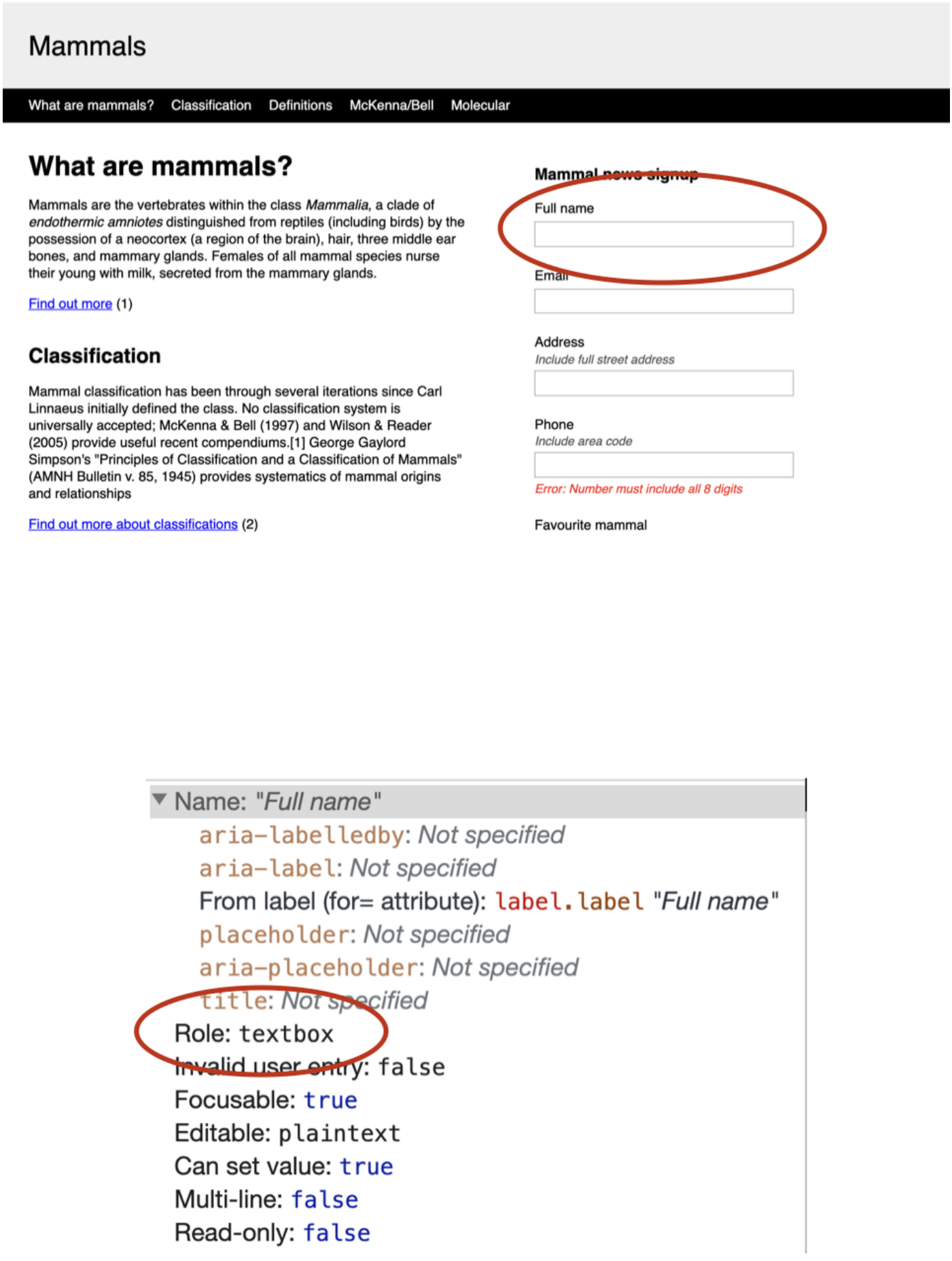
1. Heading role and level

Inspect the **“What are mammals?”** heading. What is this element’s `role` and `level` in the accessibility panel?


Mammals
What are mammals? Classification Definitions McKenna/Bell Molecular
What are mammals?
Mammals are the vertebrates within the class Mammalia, a clade of endothermic amniotes distinguished from reptiles (including birds) by the possession of a neocortex (a region of the brain), hair, three middle ear bones, and mammary glands. Females of all mammal species nurse their young with milk, secreted from the mammary glands.
Find out more (1)
Classification
Mammal classification has been through several iterations since Carl Linnaeus initially defined the class. No classification system is universally accepted; McKenna & Bell (1997) and Wilson & Reader (2005) provide useful recent compendiums.[1] George Gaylord Simpson's "Principles of Classification and a Classification of Mammals" (AMNH Bulletin v. 85, 1945) provides systematics of mammal origins and relationships
Find out more about classifications (2)
Mammal news signup
Full name
Email
Address
Include full street address
Phone
Include area code
Error: Number must include all 8 digits
Favourite mammal
Name: "What are mammals?"
aria-labelledby: Not specified
aria-label: Not specified
Contents: "What are mammals?"
title: Not specified
Role: heading
Level: 1

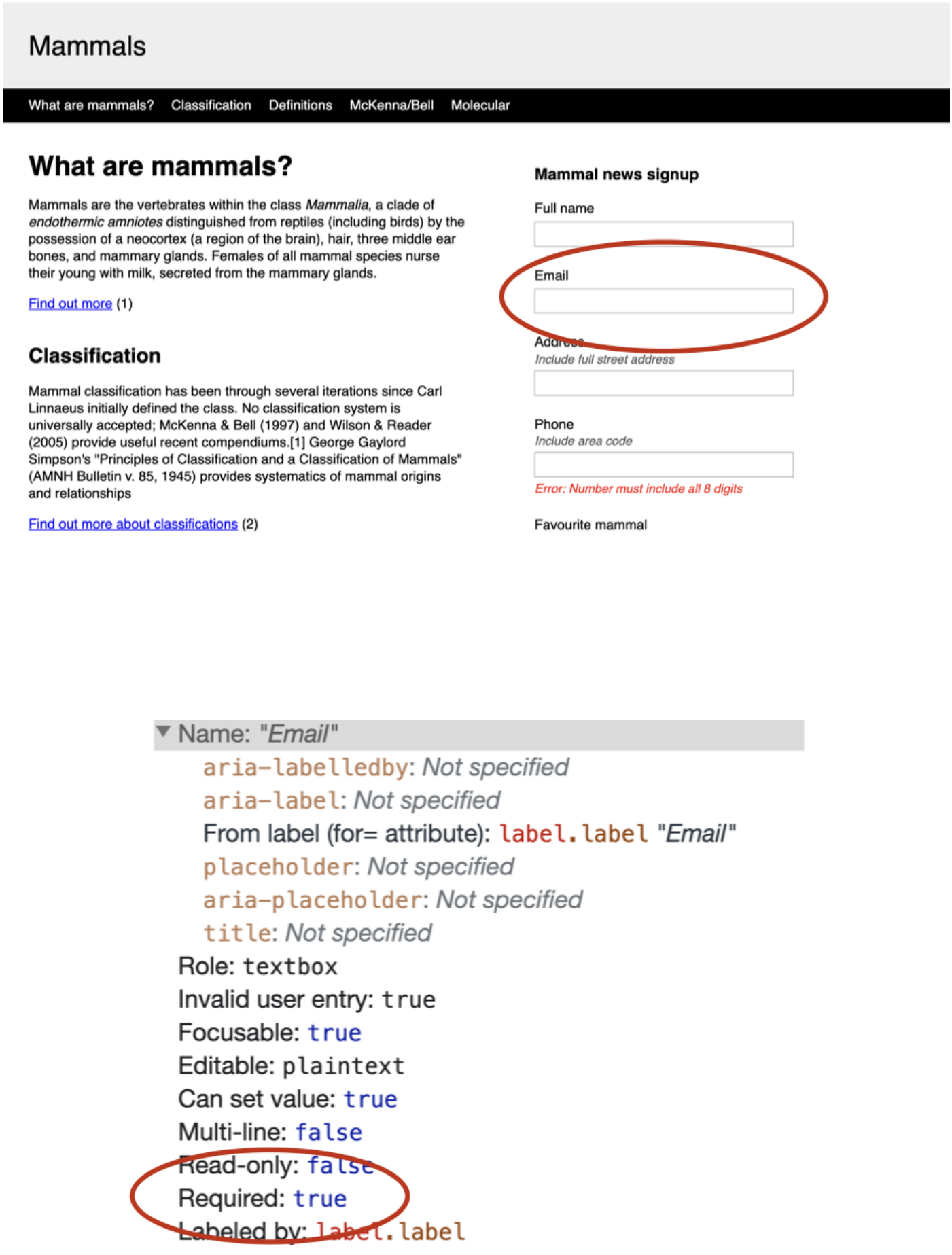
2. Input role

Inspect the **“Full name”** `<input>`. What is this elements `role` in the accessibility panel?


Mammals
What are mammals? Classification Definitions McKenna/Bell Molecular
What are mammals?
Mammals are the vertebrates within the class Mammalia, a clade of endothermic amniotes distinguished from reptiles (including birds) by the possession of a neocortex (a region of the brain), hair, three middle ear bones, and mammary glands. Females of all mammal species nurse their young with milk, secreted from the mammary glands.
Find out more (1)
Classification
Mammal classification has been through several iterations since Carl Linnaeus initially defined the class. No classification system is universally accepted; McKenna & Bell (1997) and Wilson & Reader (2005) provide useful recent compendiums.[1] George Gaylord Simpson's "Principles of Classification and a Classification of Mammals" (AMNH Bulletin v. 85, 1945) provides systematics of mammal origins and relationships
Find out more about classifications (2)
Mammal news signup
Full name
Email
Address
Include full street address
Phone
Include area code
Error: Number must include all 8 digits
Favourite mammal
Name: "Full name"
aria-labelledby: Not specified
aria-label: Not specified
From label (for= attribute): label.label "Full name"
placeholder: Not specified
aria-placeholder: Not specified
title: Not specified
Role: textbox
Invalid user entry: false
Focusable: true
Editable: plaintext
Can set value: true
Multi-line: false
Read-only: false

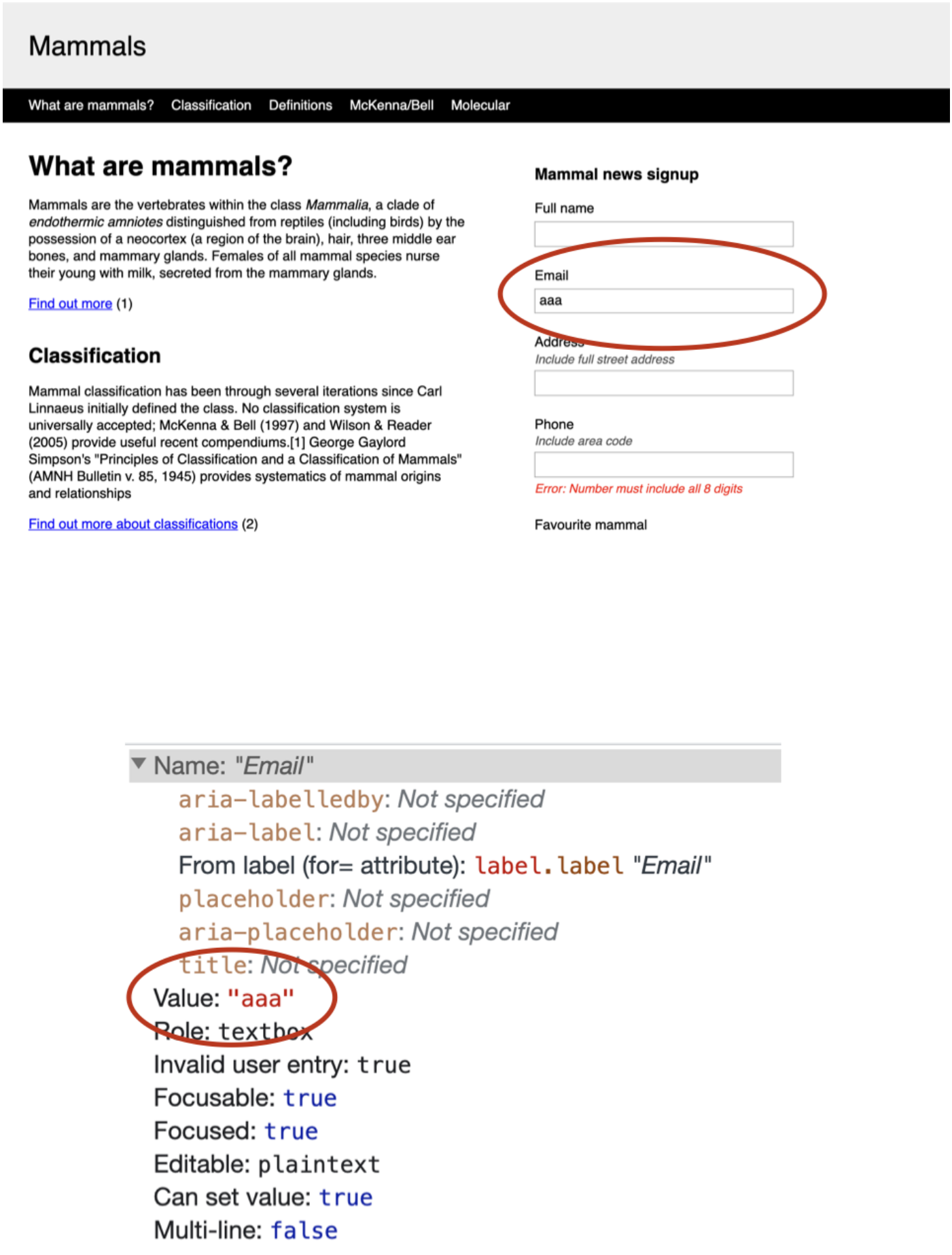
3. Required property

Inspect the **“Email”** `<input>`. Is this element defined as being `required` in the accessibility panel?


Mammals
What are mammals? Classification Definitions McKenna/Bell Molecular
What are mammals?
Mammals are the vertebrates within the class Mammalia, a clade of endothermic amniotes distinguished from reptiles (including birds) by the possession of a neocortex (a region of the brain), hair, three middle ear bones, and mammary glands. Females of all mammal species nurse their young with milk, secreted from the mammary glands.
Find out more (1)
Classification
Mammal classification has been through several iterations since Carl Linnaeus initially defined the class. No classification system is universally accepted; McKenna & Bell (1997) and Wilson & Reader (2005) provide useful recent compendiums.[1] George Gaylord Simpson's "Principles of Classification and a Classification of Mammals" (AMNH Bulletin v. 85, 1945) provides systematics of mammal origins and relationships
Find out more about classifications (2)
Mammal news signup
Full name
Email
Address
Include full street address
Phone
Include area code
Error: Number must include all 8 digits
Favourite mammal
Name: "Email"
aria-labelledby: Not specified
aria-label: Not specified
From label (for= attribute): label.label "Email"
placeholder: Not specified
aria-placeholder: Not specified
title: Not specified
Role: textbox
Invalid user entry: true
Focusable: true
Editable: plaintext
Can set value: true
Multi-line: false
Read-only: false
Required: true
Labeled by: label.label

4. Input value

Type some content into the **“Email”** `<input>` field. Does the element now have a `value` in the accessibility panel?


Mammals
What are mammals? Classification Definitions McKenna/Bell Molecular
What are mammals?
Mammals are the vertebrates within the class Mammalia, a clade of endothermic amniotes distinguished from reptiles (including birds) by the possession of a neocortex (a region of the brain), hair, three middle ear bones, and mammary glands. Females of all mammal species nurse their young with milk, secreted from the mammary glands.
Find out more (1)
Classification
Mammal classification has been through several iterations since Carl Linnaeus initially defined the class. No classification system is universally accepted; McKenna & Bell (1997) and Wilson & Reader (2005) provide useful recent compendiums.[1] George Gaylord Simpson's "Principles of Classification and a Classification of Mammals" (AMNH Bulletin v. 85, 1945) provides systematics of mammal origins and relationships
Find out more about classifications (2)
Mammal news signup
Full name
Email
aaa
Address
Include full street address
Phone
Include area code
Error: Number must include all 8 digits
Favourite mammal
Name: "Email"
aria-labelledby: Not specified
aria-label: Not specified
From label (for= attribute): label.label "Email"
placeholder: Not specified
aria-placeholder: Not specified
title: Not specified
Value: "aaa"
Role: textbox
Invalid user entry: true
Focusable: true
Focused: true
Editable: plaintext
Can set value: true
Multi-line: false

5. Checked property

Check **“Yes”** from the “Subscribe to newsletter” checkbox group. Does the checkbox now have a `checked` status in the accessibility panel?

their young with milk, secreted from the mammary glands.

Find out more (1)

Classification

Mammal classification has been through several iterations since Carl Linnaeus initially defined the class. No classification system is universally accepted; McKenna & Bell (1997) and Wilson & Reader (2005) provide useful recent compendiums.[1] George Gaylord Simpson's "Principles of Classification and a Classification of Mammals" (AMNH Bulletin v. 85, 1945) provides systematics of mammal origins and relationships

Find out more about classifications (2)

Definitions

If Mammalia is considered as the crown group, its origin can be roughly dated as the first known appearance of animals more closely related to some extant mammals than to others. *Ambondro* is more closely related to monotremes than to therian mammals while *Amphilestes* and *Amphitherium* are more closely related to the therians; as fossils of all three genera are dated about 167 million years ago in the Middle Jurassic, this is a reasonable estimate for the appearance of the crown group.

Find out more (3)

McKenna/Bell classification

Email

Address
Include full street address

Phone
Include area code

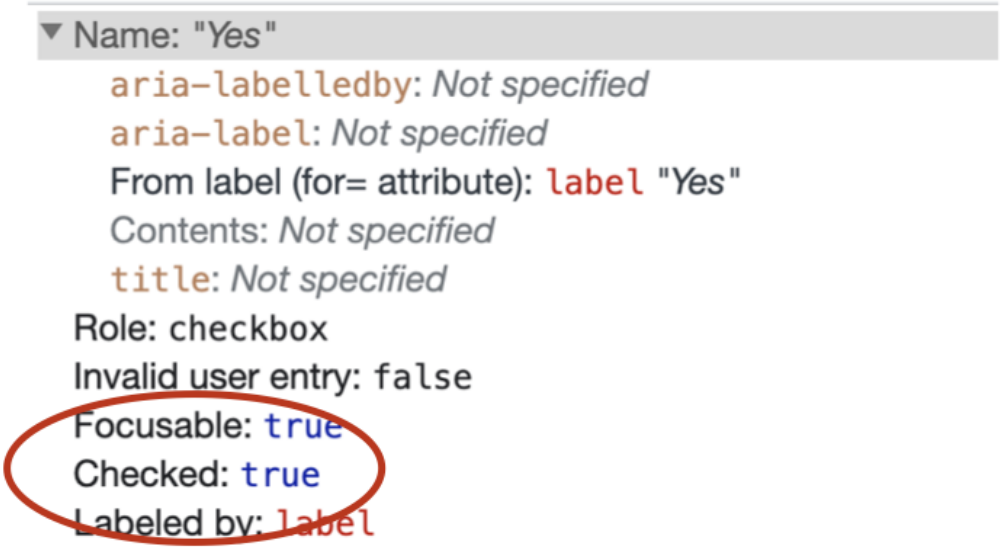
Error: Number must include all 8 digits

Favourite mammal
Choose a mammal

Subscribe to newsletter
- [x] Yes
- [ ] No

Subscription method
- ( ) Weekly
- ( ) Quarterly
- ( ) Yearly

Submit



6. Changing the semantics

Add `role="button"` to an `<h2>` element and **check the accessibility tree**.

As you can see, the `role: heading` has now been **replaced with** `role: button`.

This shows how we can **completely change the nature of an element** in the accessibility tree - which is then communicated to assistive technologies.

A wide range of properties

There are a wide range of possible properties that can be presented as part of the accessibility tree, **depending on the element being inspected**.

Some of the **key accessibility tree properties include**:

```
Name: [ accessible name as a text string ]
Role: [ pre-defined list of roles ]
Description: [ description as a text string ]
Value: [ current value as a text string ]
Required: true | false
Expanded: true | false
Checked: true | false
Disabled: true | false
Described by: [element #id]
Labeled by: [element #id]
```

Any questions or comments?

Quiz: Guessing some terms

Does anyone know what the terms **“WAI” and “ARIA” stand for**?

What is ARIA?

Web Accessibility Initiative (WAI)

WAI is a working group that develops standards and support materials to help implement accessibility.

Accessible Rich Internet Applications (ARIA)

ARIA is a standard that defines a way to make websites and web apps more accessible to people with disabilities.

ARIA defines how to **make websites more accessible**, focussing on:

- Dynamically injected content.
- Non-native widgets/components.

WAI-ARIA 1.0 was published as a completed **W3C Recommendation on 20 March 2014**.

WAI-ARIA 1.1 was published as a completed **W3C Recommendation on 14 December 2017**.

WAI-ARIA 1.2 was published as a completed **W3C Working Draft 18 December 2019**. This means it is still not the official standard yet!

Any questions or comments?

How does ARIA work?

You could think of ARIA as the **bridge** between the HTML we have now, and the HTML we wish we had.

But how does it work? ARIA can be defined in **three simple statements**:

- ARIA is a series of custom HTML attributes.
- These attributes can be used to change and augment the accessibility tree.
- They provide additional information to assistive technologies.

ARIA can be used to change and augment the accessibility tree **in the following six ways**:

1. Add or modify semantics to the accessibility tree

i.e. adding semantics to generic elements via the role attribute.

```
<div role="button"></div>
```

2. Apply states to the accessibility tree

i.e. informing assistive technologies of a widget's current state.

```
<button aria-expanded="true"></button>
```

3. Provide or improve accessible names in the accessibility tree

i.e. adding additional labelling to elements.

```
<button aria-label="Close modal"></button>
```

4. Provide descriptions to the accessibility tree

i.e. adding descriptions to elements.

```
<input type="text" aria-describedby="a1">
<span id="a1">Error message</span>
```

5. Establish relationships in the accessibility tree

i.e. informing assistive technologies of the relationships between specific elements, that may not be possible via the DOM.

```
<button role="tab" aria-controls="a1"></button>
<div role="tabpanel" id="a1"></div>
```

6. Informing assistive technologies about possible changes to the DOM

i.e. Defining a region as “live” because it may change.

```
<div aria-live="polite"></div>
```

ARIA does not do...

ARIA does not do any of the following:

1. modify an element's visual appearance
2. modify the element's behaviour
3. add focusability, or
4. add keyboard functionality.

ARIA roles, states and properties

ARIA attributes are **broken down into three categories**:

- Roles
- States
- Properties

ARIA roles are HTML attributes that are added to elements using:
`role="[role-type]"`

ARIA roles can be used to **add or change the semantic meaning** of HTML elements in the accessibility tree.

For example: *Is the element a menu, slider, spinner, progress bar?*

```
<div role="tabpanel"></div>
```

A list of all roles, including how they have changed in different versions of ARIA.

ARIA states and properties are HTML attributes that are added to elements using:
`aria-[state|property]=[property-type]`

ARIA states **define the current state** of HTML elements in the accessibility tree.

For example: *Is the element checked or disabled*?

```
<input aria-disabled="true" type="text">
```

ARIA properties **define the purpose or relationships** of HTML elements in the accessibility tree.

For example: *Does the element have a description? Does the element interact with other elements?*

```
<input aria-describedby="format">
<span id="format">(dd/mm/yyyy)</span>
```

Browser support and complexity

The different accessibility APIs mean that browsers and platforms can interpret your name, role and property information differently.

Each brand of assistive technology will prioritise different experiences for their users.

A11ySupport.io and HTML test cases have cross-browser support information for ARIA.

Any questions or comments?

Exercise: Creating a fake checkbox

Accessing the exercise:

- Open Fake checkbox - start at CodePen.io.
- View Fake checkbox - finished to check your results.

Imagine you have to make a checkbox group, but you have to to use `<div>` elements **instead of native checkboxes**.

As you will see:

- **Some basic JavaScript** is already in place.
- **Keyboard focus** is in place via `tabindex="0"`.

However, this widget is completely **inaccessible to assistive technologies**. Let's look at it in the accessibility panel.

- The elements have **no role defined**.
- The elements have **no checked state defined**.
- The `<h3>` heading is **not programmatically associated** with the checkbox group.

Step 1:

Add `role="checkbox"` to each of the fake checkboxes.

```
<div
  class="checkbox"
  tabindex="0"
  role="checkbox"
>
  Lettuce
</div>
```

```
<div
  class="checkbox"
  tabindex="0"
  role="checkbox"
>
  Tomato
</div>
```

```
<div
  class="checkbox"
  tabindex="0"
  role="checkbox"
>
  Mustard
</div>
```

Step 2:

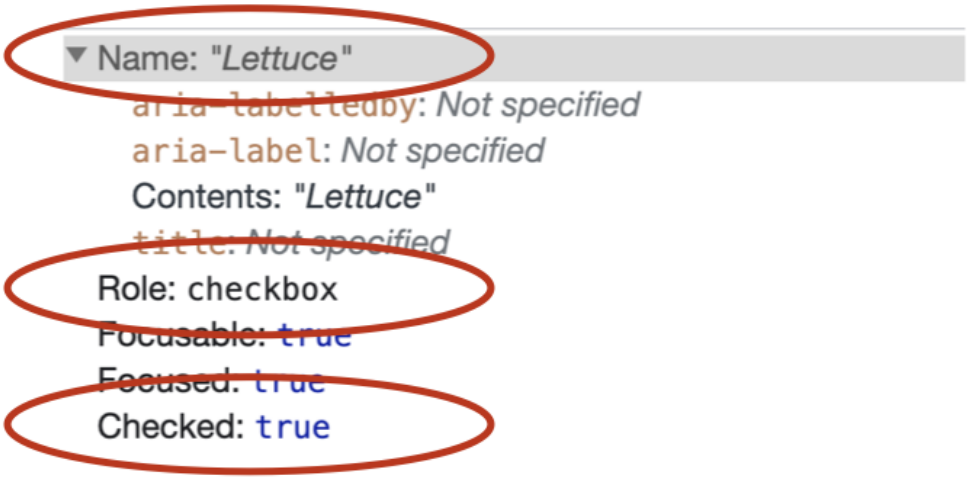
- Add `aria-checked="true"` to the first fake checkbox.
- Add `aria-checked="false"` to other fake checkboxes.

```
<div
  class="checkbox"
  tabindex="0"
  role="checkbox"
  aria-checked="true"
>
  Lettuce
</div>
```

```
<div
  class="checkbox"
  tabindex="0"
  role="checkbox"
  aria-checked="false"
>
  Tomato
</div>
```

```
<div
  class="checkbox"
  tabindex="0"
  role="checkbox"
  aria-checked="false"
>
  Mustard
</div>
```

If we inspect any of the fake checkboxes in the accessibility tree, **they now have roles and states**.



Step 3:

Add `role="group"` to the parent container so that we can create a fake `<fieldset>`.

```
<div
  role="group"
>
</div>
```

Step 4:

Add `aria-labelledby="group-label"` to the parent container so we can give it an accessible name.

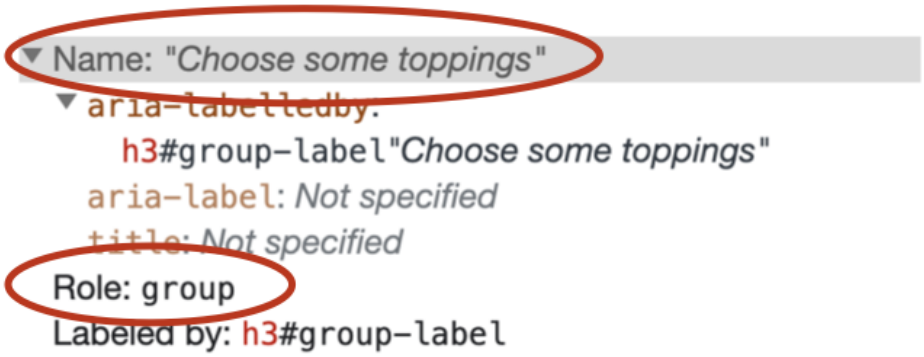
```
<div
  role="group"
  aria-labelledby="group-label"
>
</div>
```

Step 5:

Add `id="group-label"` to the `<h3>` element. Now it will operate like a fake `<legend>`.

```
<h3 id="group-label">Choose some toppings</h3>
```

If we inspect the parent container in the accessibility tree, **it now has an accessible name and a role**.



This is a quick example to show how ARIA is used to **augment the accessibility tree** to help assistive technologies.

Forms and assistive technologies

Screen reader users can navigate through a page using **a range of different methods**.

However, when some screen readers reach a form element they switch from **“Browse mode” to “Forms mode”.**

In “Forms mode”, the **only way to navigate through the form fields** is via the `TAB` key.

In this mode, focus will move from one focusable element to the next. Any elements that do not receive focus **will be ignored**.

For this reason, we need to **programmatically associate** all relevant information to the form field.

This makes sure that all the relevant information is **announced when the form field is in focus**.

Relevant information could include:

- The form field's label.
- Additional information to help users fill in the form field.
- Error messages associated with the form field.
- If the form field is required.

For the rest of this session, we will focus on:

- How to **create labels for form fields** - using accessible names and visible text labels.
- How to **programmatically associate labels and additional information** with form fields.

In the following sessions, we'll look at programmatically associating **required fields and error messages** with form controls.

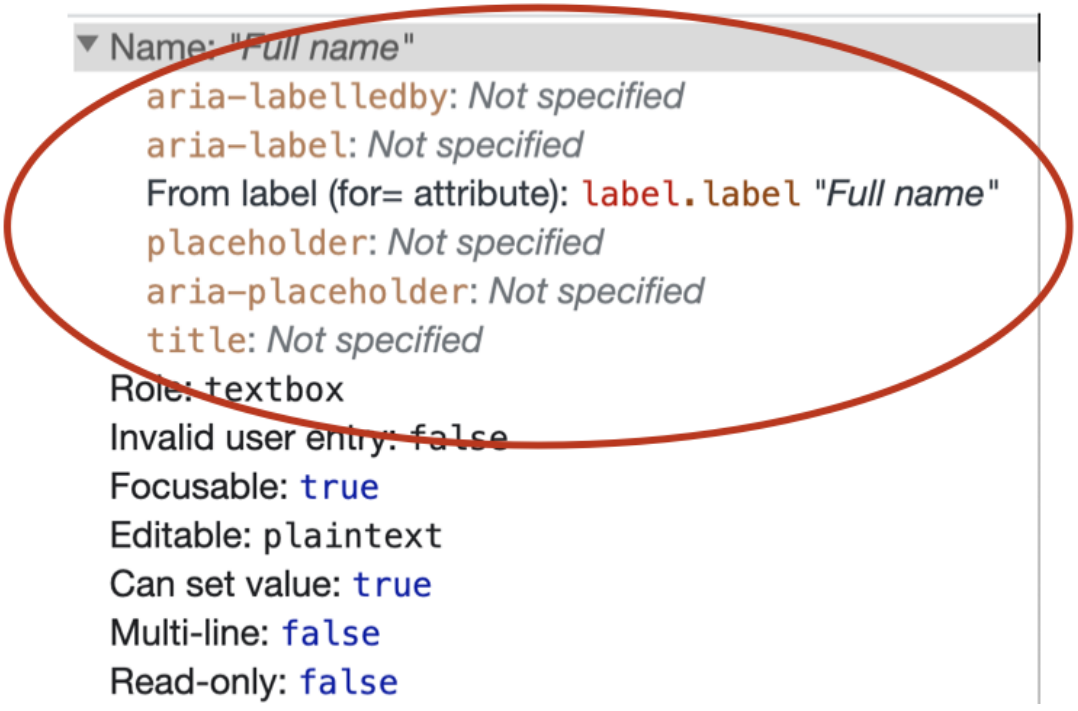
Accessible names

Accessible names are names given to elements in the accessibility tree.

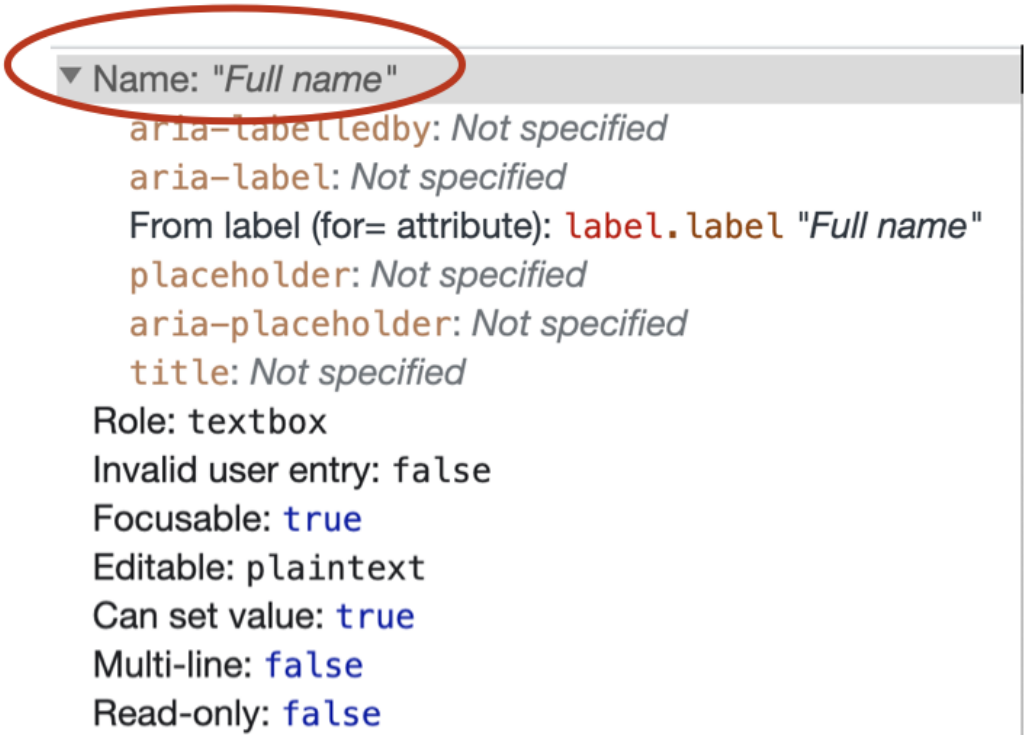
Accessible names are very important for assistive technology users as they **help to identify elements within the accessibility tree**.

Accessible names are defined as **text strings**.

Chrome's accessibility tree shows **all the possible options** that could be used to provide the accessible name.

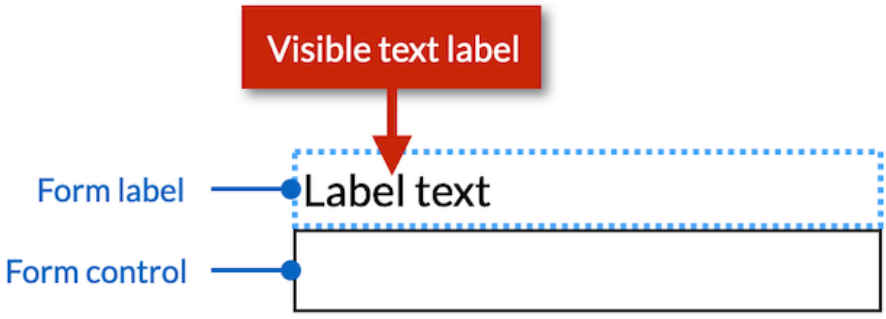


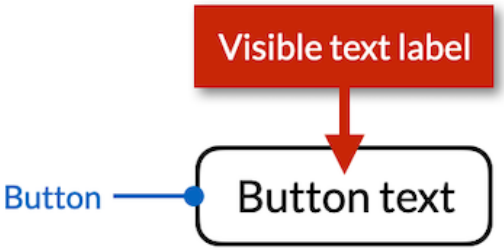
It also displays the **final computed accessible name** as a text string.

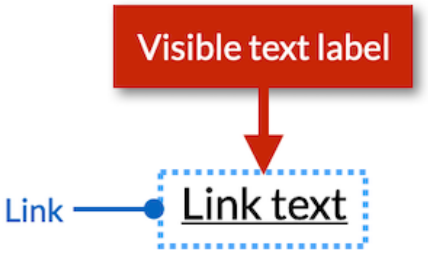


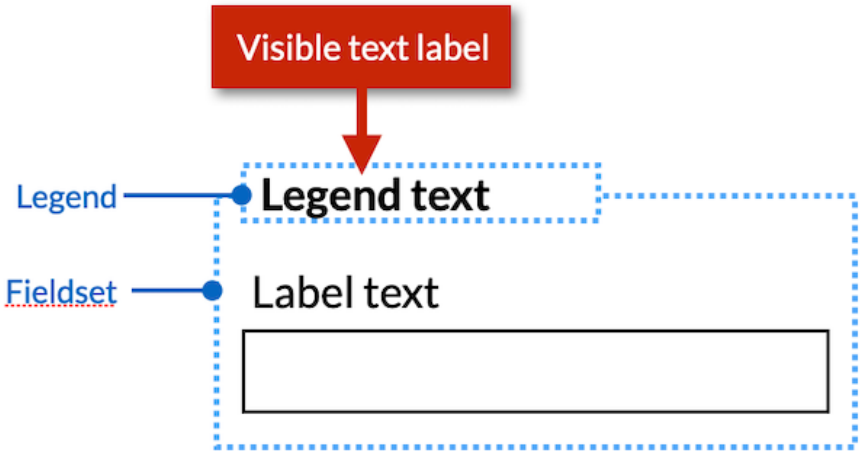
Visible text labels

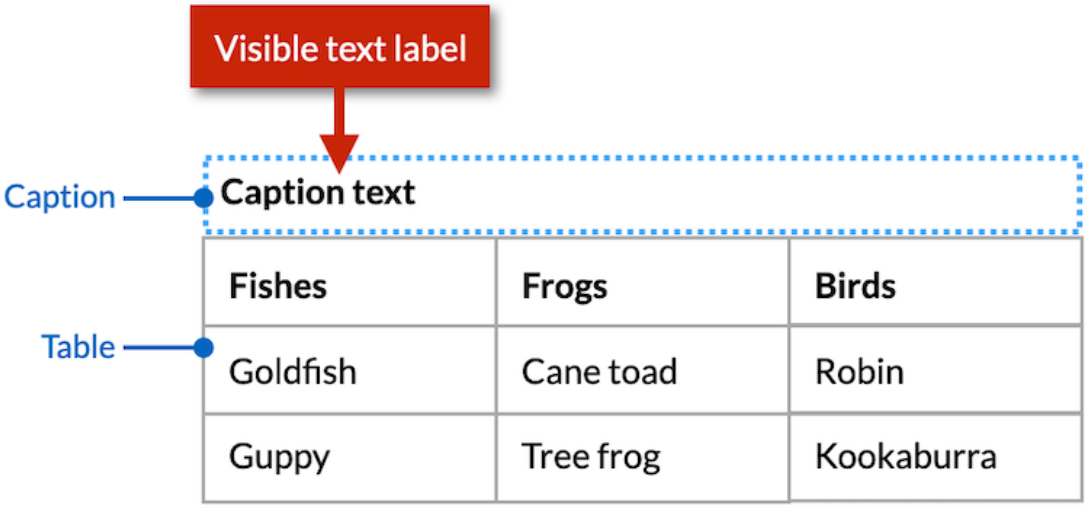
A visible text label is a label that is **presented on-screen as text**. Visible labels can include:

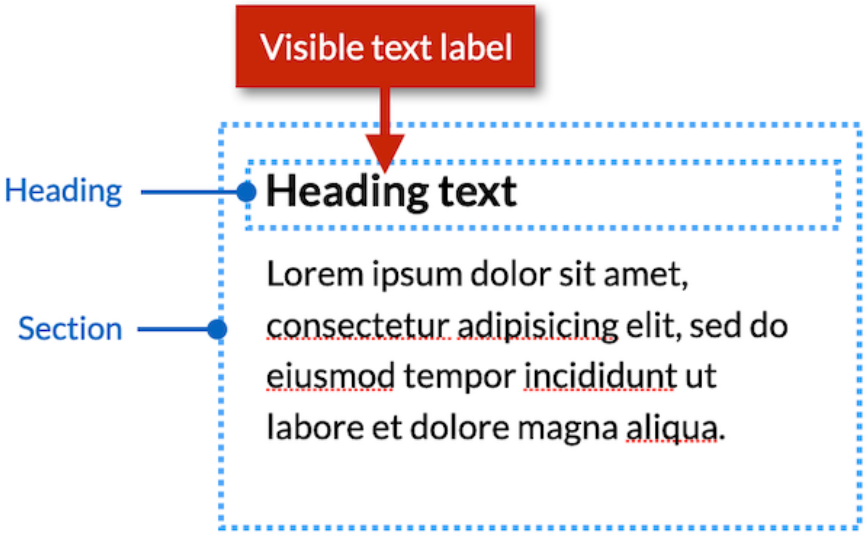

Visible text label
Form label
Label text
Form control


Visible text label
Button
Button text


Visible text label
Link
Link text


Visible text label
Legend
Legend text
Fieldset
Label text

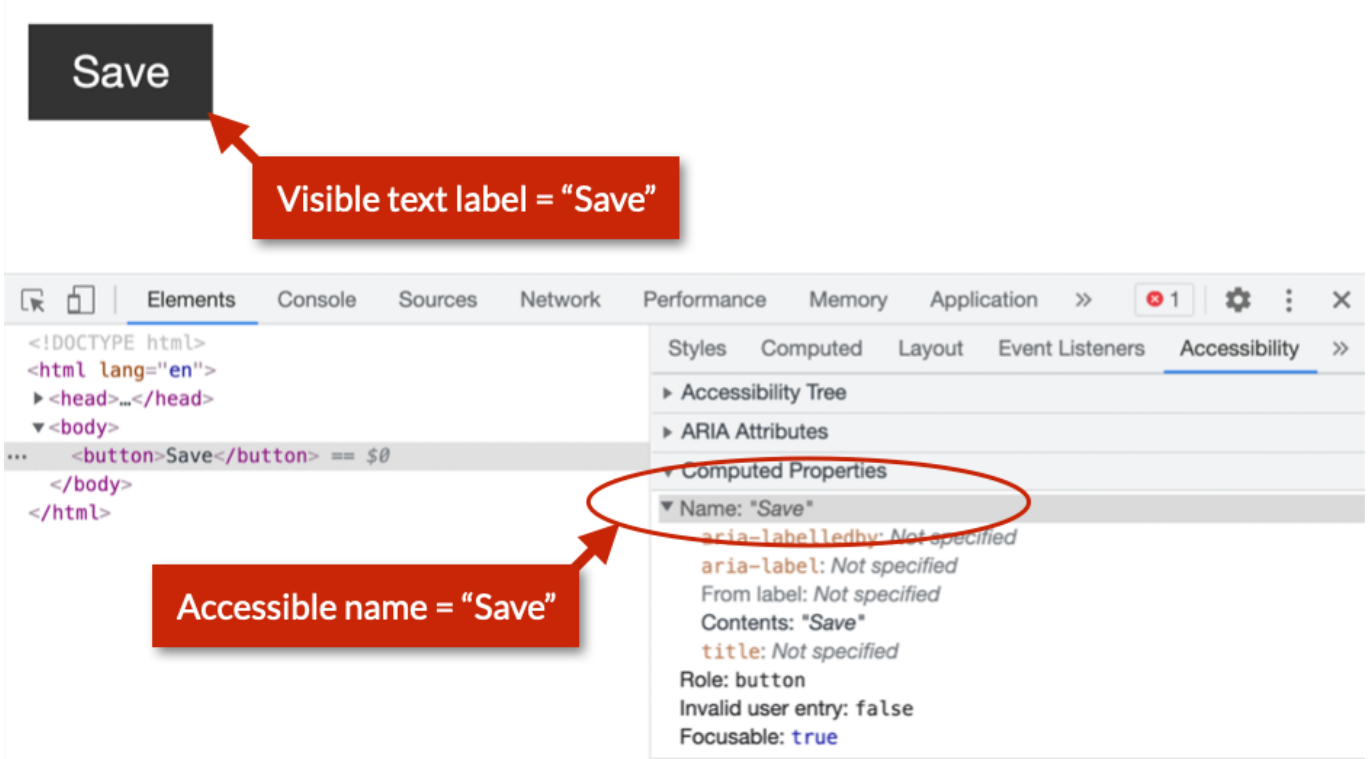

Visible text label
Caption
Caption text
Table
Fishes
Frogs
Birds
Goldfish
Cane toad
Robin
Guppy
Tree frog
Kookaburra


Visible text label
Heading
Heading text
Section
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

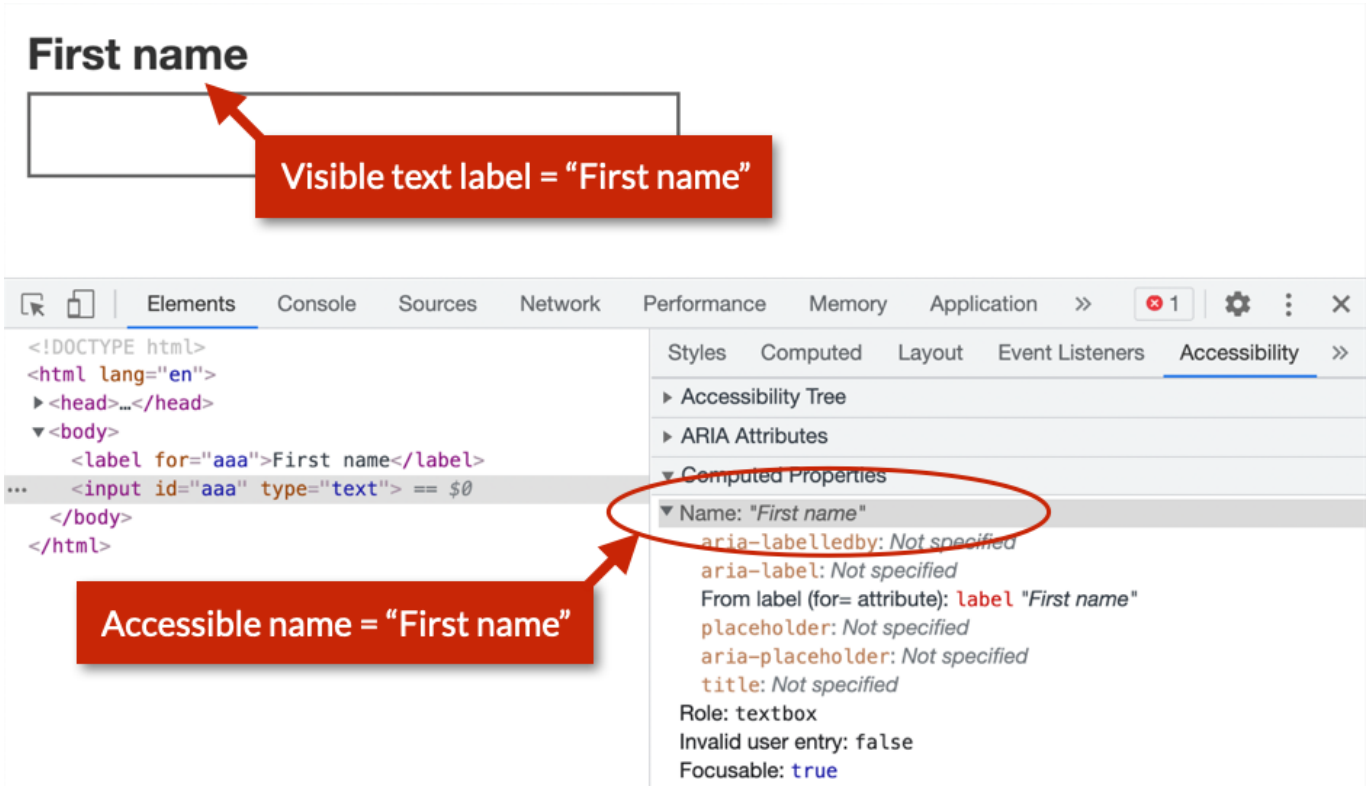
Do accessible names and visible text labels have to match?

In some cases, the accessibility name **matches the visible text label**.

```
<button>Save</button>
```



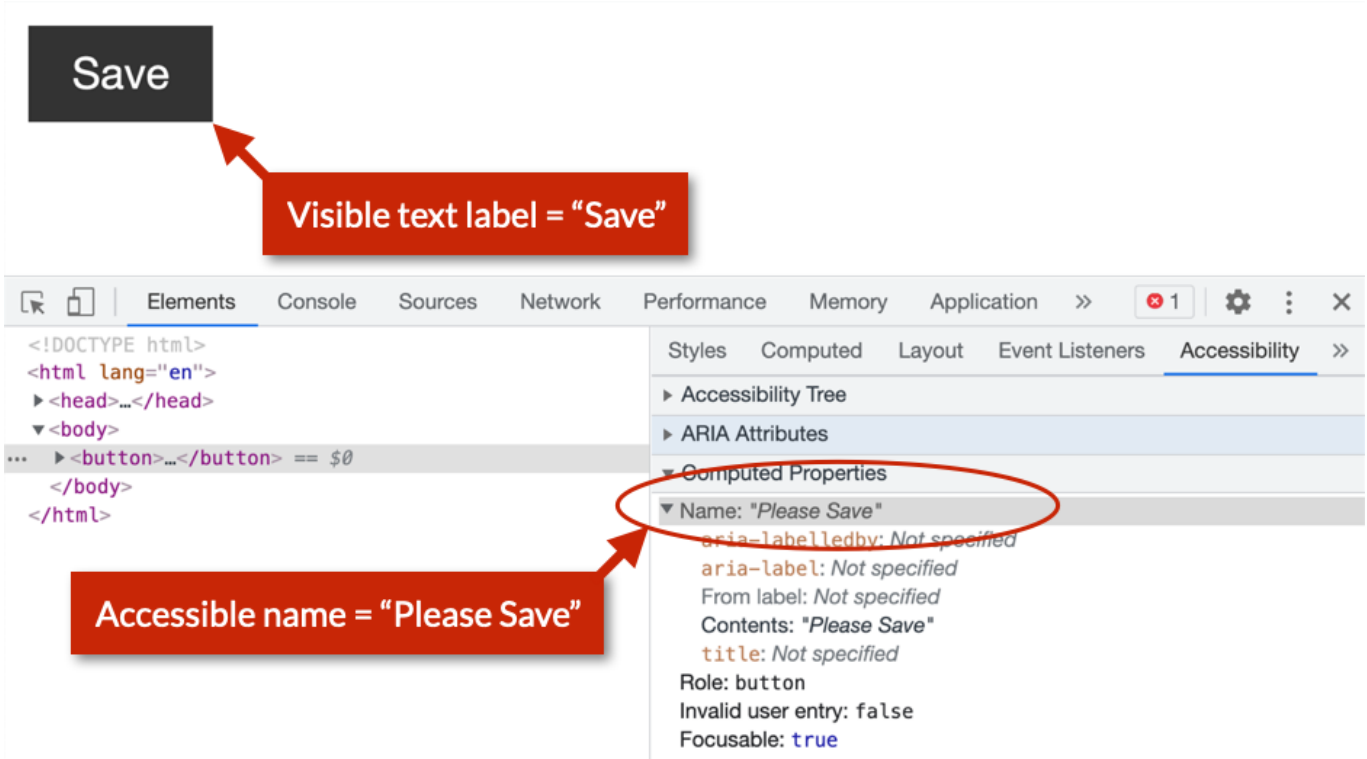
```
<label for="aaa">First name</label>
<input id="aaa" type="text">
```



It is possible to create an accessible name that is **partially or even entirely different** from the visible text label.

For example, the accessible name could include additional content **before or after the visible text label.**

```
<button>
  <span class="sr-only">Please</span>
  Save
</button>
```



However, accessible names must match **at least some aspect of the visible text label**.

A mismatch between the visible text label and the accessible name could **cause confusion for some specific types of users**.

For example, some people with mobility issues use speech recognition tools. If the visible text label doesn’t match the accessible name in the API, **the control won't respond to their command**.

And some sighted users use screen readers for additional assistance. If the visible text label doesn’t match the accessible name, **this could be confusing**.

This mismatch could also be confusing if screen reader users **need to describe aspects of the UI** to sighted users and vice versa.

Any questions or comments?

`aria-labelledby`

This is where a primary element is **given a label** (another name for an accessible name) by a secondary elements text string.

The `aria-labelledby` attribute is **applied to the primary element**, and the matching ID value is applied to the secondary element.

```
<section aria-labelledby="aaa">
  <h3 id="aaa">Contact details</h3>
</section>

Section accessible name: "Contact details"
```

The secondary element **does not need to be a child** of the primary element.

```
<p id="bbb">Buy Lawn Mower</p>
<button aria-labelledby="bbb">Buy</button>

Button accessible name: "Buy Lawn Mower"
```

More than one label can be applied to the primary element. They need to be space separated values in the `aria-labelledby` attribute.

```
<p id="ccc">Buy Lawn Mower</p>
<button aria-labelledby="ccc ddd">Buy</button>
<p id="ddd">On special</p>

Buttons accessible name: "Buy Lawn Mower On specia
```

Exercise: Using `aria-labelledby`

Accessing the exercise:

DEVELOPER EXERCISE: Using aria-labelledby.

`aria-label`

The `aria-label` attribute is used to provide a label **directly to the element itself**.

```
<button aria-label="Close and return">
  Close
</button>

Buttons accessible name: "Close and return"
```

Exercise: Using `aria-label`

Accessing the exercise:

DEVELOPER EXERCISE: Using aria-label.

Exercise: Adding accessible names

Accessing the exercise:

DEVELOPER EXERCISE: Adding accessible names.

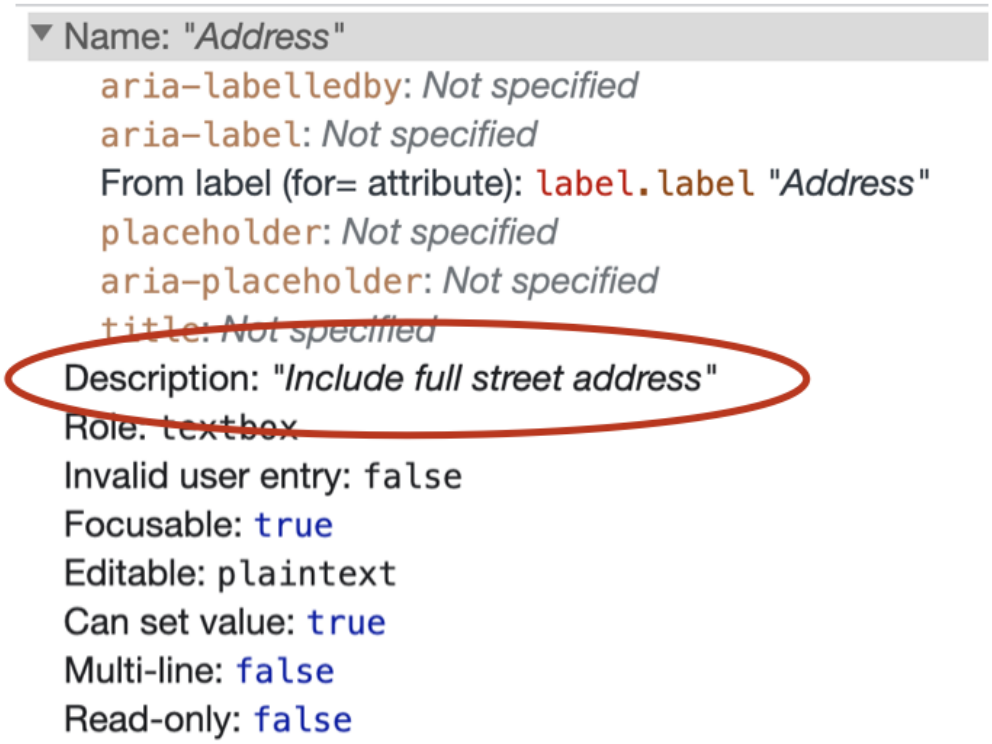
Accessible descriptions

In some cases, accessible objects may need more than just an accessible name to **provide additional context.**

For example, **help text or error messages** associated with individual form controls.

This additional information is referred to as an **“accessible description”** if it is exposed in the accessibility tree.

Accessible descriptions are **defined as text strings**.



The accessible description **might or might not** be presented visually on-screen.

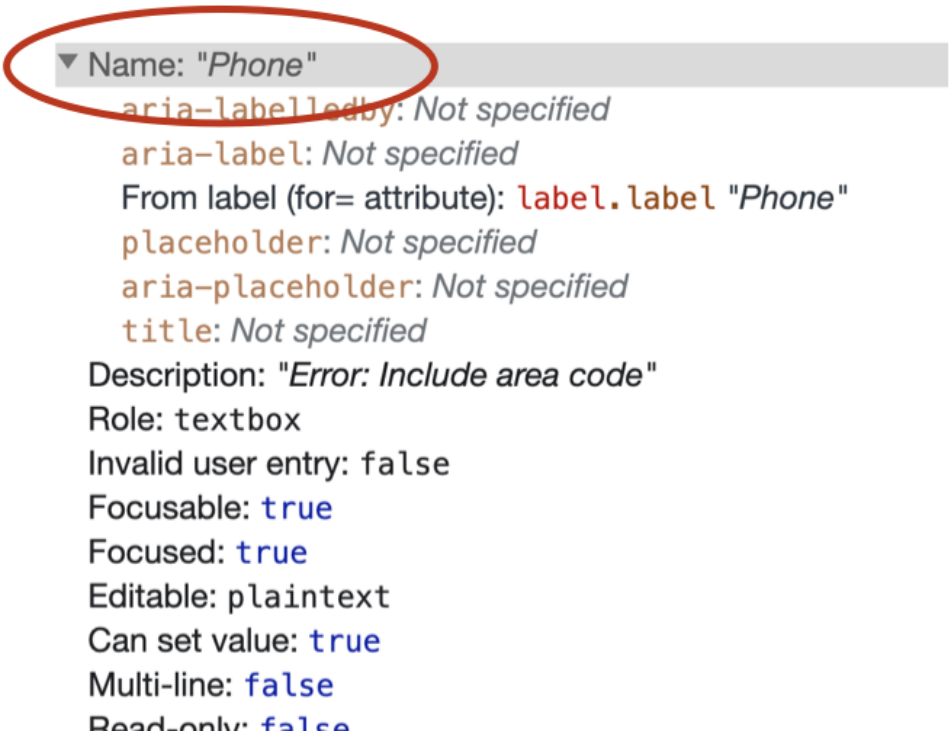
Accessible names and descriptions are **two totally separate concepts**. They fill two different slots in the accessibility tree.

This is where a primary element is **given a description** by a secondary elements text string.

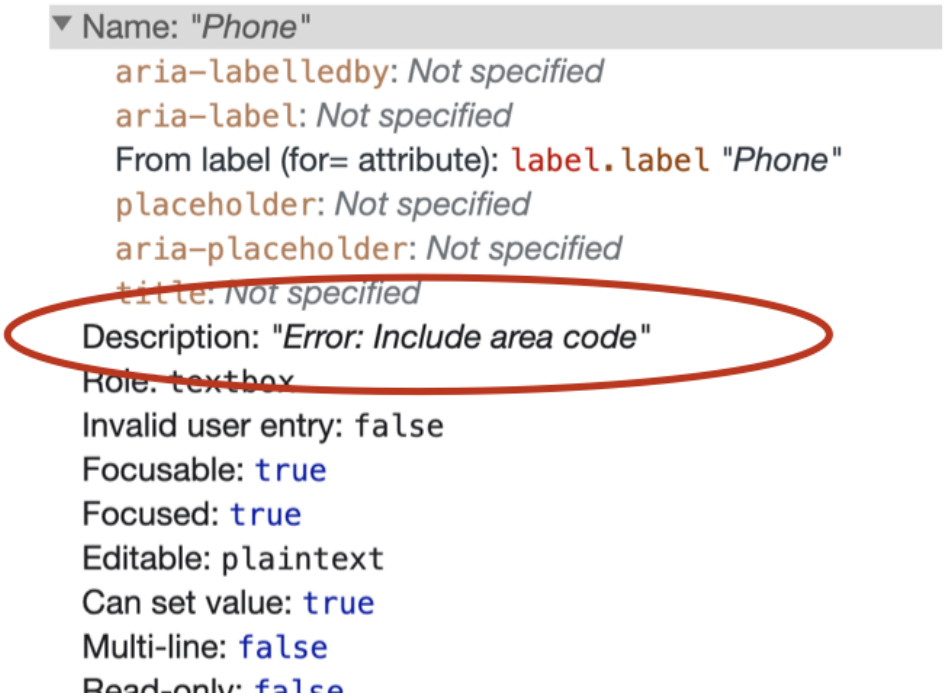
The `aria-describedby` attribute is **applied to the primary element**, and the matching `ID` value is applied to the secondary element.

```
<label for="a">Phone</label>
<input id="a" type="text" aria-describedby="i1">
<p id="i1">Error: Include area code</p>
```

This `<input>` element already **has an accessible name of “Phone”**.



However, it will now have an accessible description of **“Error: Include area code”** in the accessibility tree.



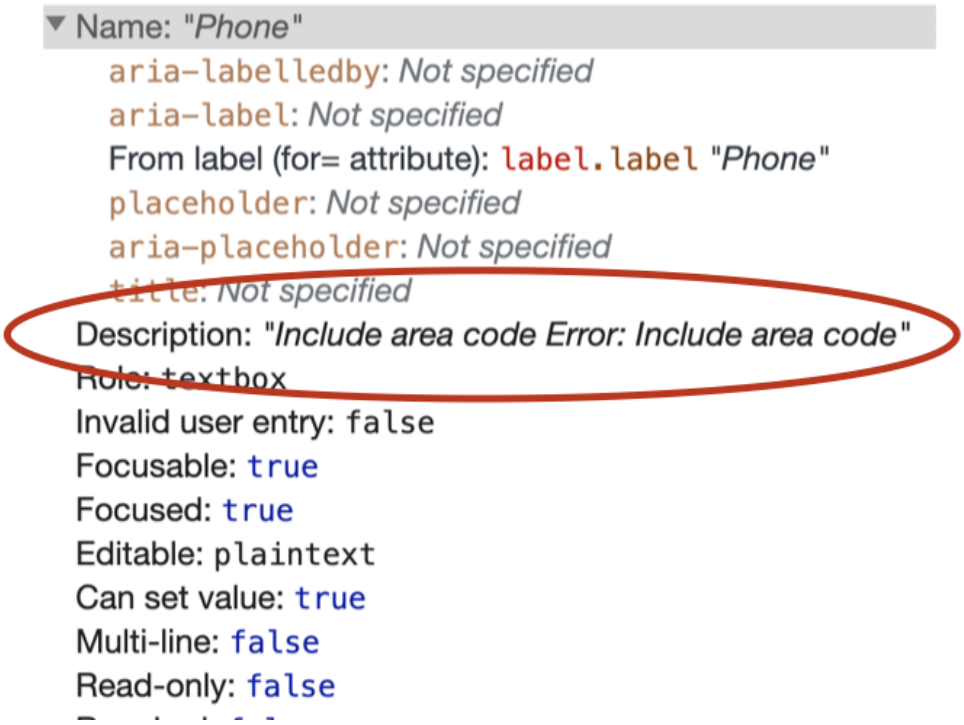
The `aria-describedby` attribute **cannot provide an accessible name** in the accessibility tree - it is just for descriptions.

Multiple descriptions can be applied to a single element using `aria-describedby`. They are defined using space-separated values.

```
<label for="a">Phone</label>
<p id="hint1">Include an area code</p>
<input id="a" type="text" aria-describedby="hint1
<p id="error1">Error: Include area code</p>
```

The order that these elements are defined within the aria-describedby value **determines the order in which the descriptions are announced**.

In this case, because `hint1` is defined before `error1` the **accessible description** will be:
“Include an area code Error: Include area code”



Any questions or comments?

Exercise: Adding accessible descriptions

Accessing the exercise:

DEVELOPER EXERCISE: Adding accessible descriptions.

Which one wins?

So, what happens when you use **multiple methods** to provide an accessible name or description for an element?

Let's look at an example with an `<input>` has **three possible sources** for the accessible name.

```
<!-- An associated label -->

<label for="a">Name</label>
<input
 id="a"
 type="text"
 aria-label="Full name"
 placeholder="Add your full name"
>
```

```
<!-- An aria-label attribute -->

<label for="a">Name</label>
<input
 id="a"
 type="text"
 aria-label="Full name"
 placeholder="Add your full name"
>
```

```
<!-- A placeholder attribute -->

<label for="a">Name</label>
<input
 id="a"
 type="text"
 aria-label="Full name"
 placeholder="Add your full name"
>
```

In examples like this, which of these three sources would **be used to define the accessible name**?

Despite multiple possible sources, **only one accessible name** can be defined and applied for any element.

Accessible names should be **defined in the following order**:

1. If present, use `aria-labelledby`.
2. Otherwise, if present, use `aria-label`.
3. Otherwise, if present, use:
    - The elements content OR
    - The elements `alt` OR
    - Content from `<label>`, `<legend>`, `<caption>`
4. Otherwise, if present, use `title`.
5. Otherwise, if present, use `placeholder`.
6. Otherwise, there is no accessible name.

Always try to use **the most semantic method** to apply accessible names and avoid ARIA solutions where possible.

```
<!-- Button name from text string -->
<button>Button text</button>
```

```
<!-- Link name from text string -->
<a href="#">Link text</a>
```

```
<!-- Image name from alt attribute value -->
<img src="a.png" alt="Alt attribute text">
```

```
<!-- Input name from label text string -->

<label for="a">Label text</label>
<input id="a" type="text">
```

```
<!-- Fieldset name from legend text string -->

<fieldset>
  <legend>Legend text</legend>
</fieldset>
```

```
<!-- Table name from caption text string -->

<table>
  <caption>Caption text</caption>
</table>
```

Accessible descriptions should be **defined in the following order**:

1. If present, use `aria-describedby`.
2. Otherwise, if present, use `title`.
3. Otherwise, there is no accessible description.

The `title` attribute

If there is another accessible name option available, the `title` will **move to the accessible description slot**.

However, if the `aria-describedby` is present, the `title` will **not be used as the description either**.

The `title` attribute **should only ever be used** as an accessible name for `<frame>` and `<iframe>` elements.

And, the `title` attribute **should never be used as the accessible description** for elements.

The `placeholder` attribute

Despite the rules for calculating names, the `placeholder` **is used instead of the** `title` in most browser/screen reader combinations.

If another accessible name is present, the `placeholder` is **still announced** in most browser/screen reader combinations.

In some browser/screen reader combinations, the `placeholder` is announced **as part of the accessible name**.

In other browser/screen reader combinations, it is announced **as part of the accessible description**.

The `placeholder` should never be used **as the only visible text label or accessible name** for elements.

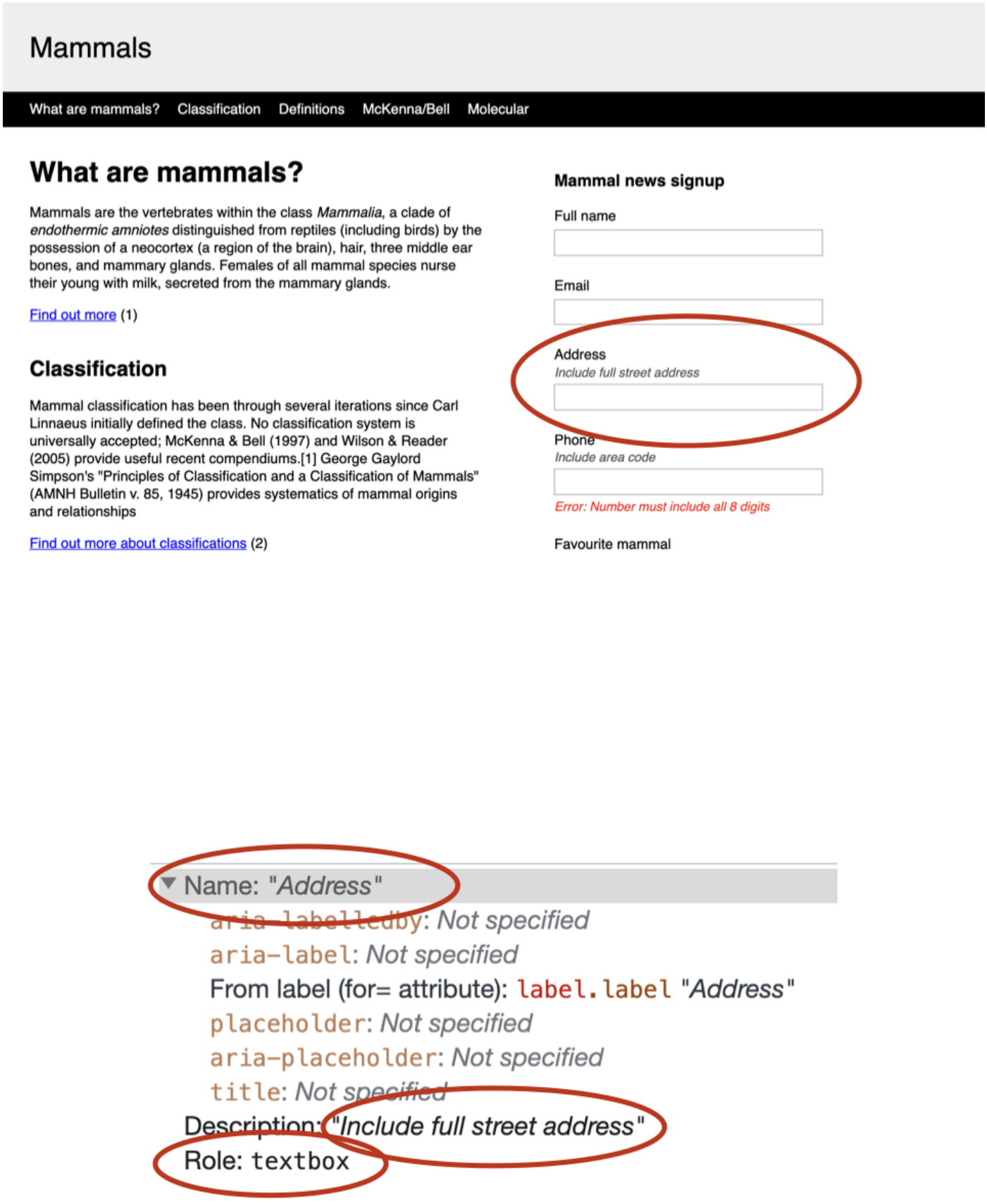
How names and descriptions are announced

How are the accessible name and description **announced in different screen readers**?

Screen readers announce objects within the accessibility tree **in the following order**:

- Accessible name
- Role
- Description, then
- Additional instructions.

Let's use the "Address" `<input>` again to **see the announcement order** in different screen readers.


Mammals
What are mammals? Classification Definitions McKenna/Bell Molecular
What are mammals?
Mammals are the vertebrates within the class Mammalia, a clade of endothermic amniotes distinguished from reptiles (including birds) by the possession of a neocortex (a region of the brain), hair, three middle ear bones, and mammary glands. Females of all mammal species nurse their young with milk, secreted from the mammary glands.
Find out more (1)
Classification
Mammal classification has been through several iterations since Carl Linnaeus initially defined the class. No classification system is universally accepted; McKenna & Bell (1997) and Wilson & Reader (2005) provide useful recent compendiums.[1] George Gaylord Simpson's "Principles of Classification and a Classification of Mammals" (AMNH Bulletin v. 85, 1945) provides systematics of mammal origins and relationships
Find out more about classifications (2)
Mammal news signup
Full name
Email
Address
Include full street address
Phone
Include area code
Error: Number must include all 8 digits
Favourite mammal
Name: "Address"
aria-labelledby: Not specified
aria-label: Not specified
From label (for= attribute): label.label "Address"
placeholder: Not specified
aria-placeholder: Not specified
title: Not specified
Description: "Include full street address"
Role: textbox

With VoiceOver on the desktop, **there is a pause** after the accessible name and role are announced.

OSX / VoiceOver / Chrome

- Accessible name: “Address”.
- Role: “Edit text”.
- [Pause].
- Description: “Include full street address”.
- Additional instructions: “You are currently on a text field. To enter text in this field, type”.

Windows / NVDA / Chrome

- Accessible name: “Address”.
- Role: “Edit”.
- Description: “Include full street address”.
- Additional instructions: “Blank”.

Windows / JAWS / Chrome

- Accessible name: “Address”.
- Role: “Edit”.
- Description: “Include full street address”.
- Additional instructions: “Type in text”.

Any questions or comments?

Recap

Developers create accessible sites by using **flexible** markup and styles, with all information available via the accessibility API for assistive technology to **transform** as needed.