

Using DevTools Accessibility tab when manual testing HTML and ARIA.

IT Accessibility Community Meeting
August 6, 2024

Before diving into the **DevTools Accessibility tab**, let's review three basic concepts about **HTML elements**:

Accessible names

Roles

Accessible descriptions

Accessible names

Accessible names are short text strings that **provide an assistive technology user with a label for the element.**

Accessible names convey the **purpose of the element**.

Accessible names **distinguish elements from one another.**

Each **accessible name** should be unique.

Roles

Most **HTML** elements have a 'role'.

Roles define the element's function – especially for assistive technologies.

The `<input type="radio">` element has role of **radio**.

The `<a>` element has a role of **link**.

The `<div>` element has no inherent role.

Accessible descriptions

Sometimes, an HTML element's short accessible name may need **additional context**.

For example, **instructions or error messages** associated with form controls.

If applied correctly, additional information is provided as an **accessible description**.

How names, roles and
descriptions are announced

Address

Enter full address

```
<! ---- Accessible name --->
```

```
<label for="userAddress">Address</label>
```

```
<span id="inst">Enter full address</span>
```

```
<input id="userAddress" type="text"/>  
aria-describedby="inst" required />
```

<! ---- Role --->

```
<label for="userAddress">Address</label>  
<span id="inst">Enter full address</span>  
<input id="userAddress" type="text"/>  
aria-describedby="inst" required />
```

```
<! ---- Accessible description --->
```

```
<label for="userAddress">Address</label>
```

```
<span id="inst">Enter full address</span>
```

```
<input id="userAddress" type="text"/>  
aria-describedby="inst" required />
```

```
<! ---- State --->
```

```
<label for="userAddress">Address</label>  
<span id="inst">Enter full address</span>  
<input id="userAddress" type="text"/>  
aria-describedby="inst" required />
```

Windows / NVDA and JAWS

1. Accessible name: **Address**
2. Role: **textbox**
3. State: **required**
4. Description: **Enter full address**

OSX / VoiceOver

1. Accessible name: **Address**
2. State: **required**
3. Role: **textbox**
4. Description: **Enter full address**

How do screen readers access information?

Via **the accessibility tree**.

What is the accessibility tree?

Document Object Model

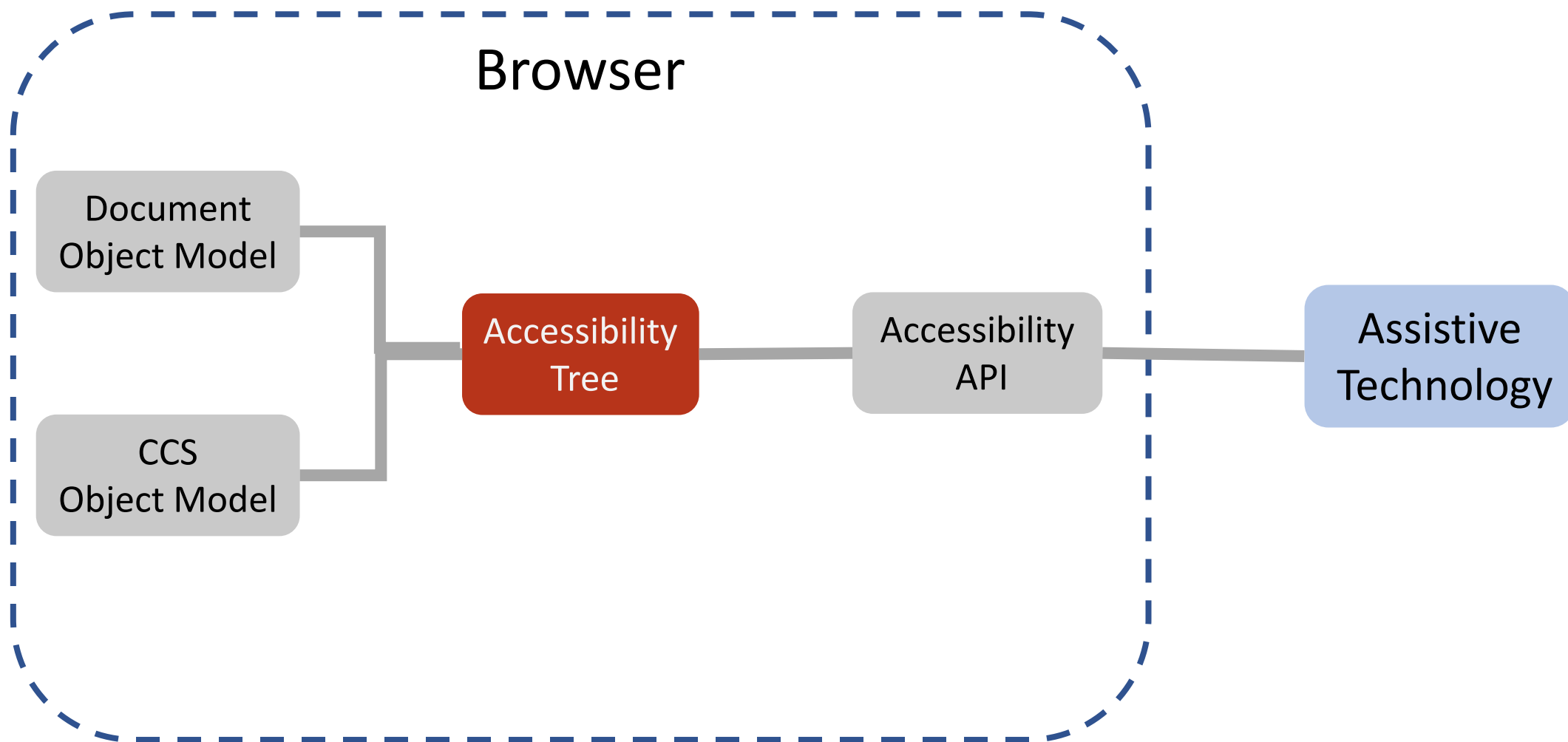
The accessibility tree is a simplified version of the Document Object Model.

The accessibility tree contains accessibility-related information for most HTML elements.

Accessibility tree Information

- Name
- Role
- Description
- State
- Value

This information is given to assistive technologies so that users can **understand, navigate and interact with web documents.**



Each **browser** creates **its own version of the accessibility tree** and their own Accessibility APIs.

Why use DevTools Accessibility tab?

DevTools has a feature called the '**Computed properties**' panel.

The **Computed properties** panel gives us a range of accessibility information in one location.

How to access DevTools
Accessibility tab.

Activity 1:

- Go to the [Section508.gov](https://www.section508.gov/) page.

<https://www.section508.gov/>

Step 1: Open Developer Tools

1. Right-click anywhere on the page.
2. Select 'Inspect'.
3. This will open DevTools.

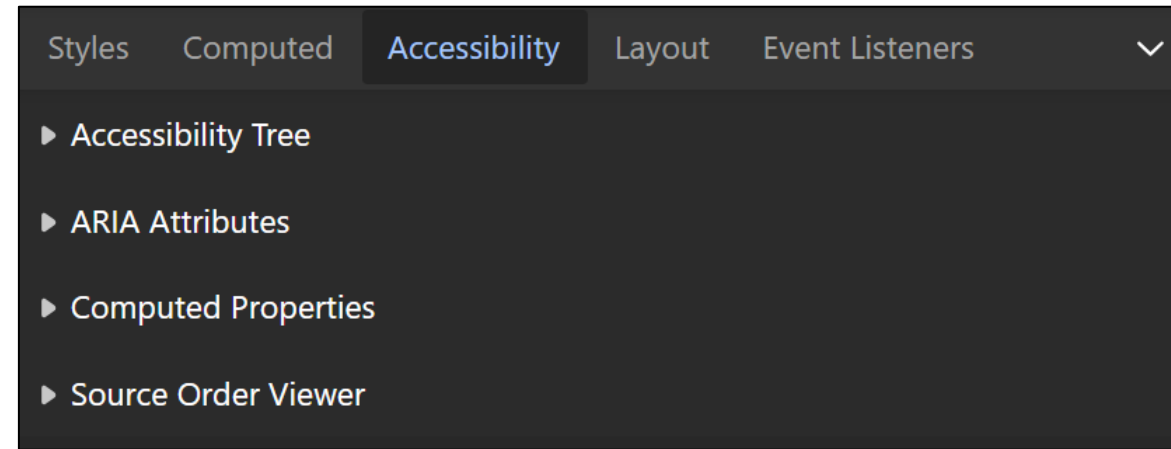
Alternatively, you can use:

- Windows: Control + Shift + C.
- Mac: Command + Opt + I

Step 2: Select the 'Accessibility' tab

The Accessibility tab displays:

- 'Accessibility tree' panel
- 'ARIA Attributes' panel
- 'Computed Properties' panel
- 'Source Order Viewer' panel



The Accessibility tab demo

1. Inspect the 'View instructions' link.

In the Computed properties panel, find the element's name and role.

Answer:

Name: View Instructions

Role: link

2. Inspect the 'Search' <input>

In the Computed Properties panel, find the element's name and role:

Answer

Name: *"Search"*

Role: *"searchbox"*

3. Add text into the 'Search' <input>

Does the element now have a value in the Computed Properties panel?

Answer

Value: *'training'*

This tells assistive technologies what the user has added to the form field before submitting the form, **allowing the user to review the information**

4. Inspect the 'Our Mission' heading

In the Computed Properties panel, find the element's role and level:

Answer

Role: *heading*

Level: 2

5. Inspect the image in Our Mission section

In the Computed Properties panel, find the element's name and role:

Answer

Name: "A portion of a computer keyboard with shining Accessibility text and symbol in place of the enter key"

Role: *image*

6. Inspect the icon for 'Find your 508 PM'

Does this element provide any information in the Computed Properties panel?

Answer

Accessibility node not exposed

Some elements are **not exposed in the accessibility tree.**

7. Inspect the Key dates table

Click the View Instructions link under FY24 Governmentwide 508 Assessment. Inspect the Key Dates table.

In the Computed Properties panel, find the <table> element's name and role:

Answer

Name: *"Table listing of five key dates related to the reporting required under Section 752 of the Consolidated Appropriations Act of 2023"*

Role: *table*

Essential elements within the table **also have roles:**

Answer

<tr> role = *row*

<th> role = *columnheader or rowheader*

<td> role = *gridcell*

Other properties...

Properties found in the DevTools Computed Properties(depending on the element) are:

- **Name:** [accessible name as a text string]
- **Role:** [pre-defined list of valid 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]

ARIA alters the accessibility tree

Accessible Rich Internet Applications

ARIA is a set of custom HTML attributes that **add information to the accessibility tree.**

Information is intended to help assistive technology users understand the **name, role or state** of elements.

First rule of ARIA is **Don't use ARIA when semantic HTML elements can be used.**

Activity 2:

Go to the [Section508.gov](https://www.section508.gov/) Home page.

We can see how ARIA attributes can affect the **accessibility tree**.

1. Testing role="search"

Inspect the Search Input.

Check the parent <div> element's role in the Computed Properties panel:

Answer

Role: *Search*

We can use ARIA to add semantics to an element that has none.

2. Testing aria-label

On the Section508.gov home page, Click the Policy and Management link and inspect the Management link in the secondary navigation.

Check the <a> link element's name in the Computed Properties:

Answer

Name: *'Management – Submenu'*

ARIA **overrides contents** of the element 'Management'.

3. Testing aria-expanded

Expand the Management sub list.

Check the <a> Link element's 'Expanded' state in the Computed **Properties**:

Answer

Expanded: *true*

We can use ARIA to inform assistive technology users **when an element is expanded or collapsed**.

DevTools is helpful when **reviewing the accessibility of elements.**

Accessibility tree

accessible name generated for each element.

Computed properties

- All possible names (from strongest to weakest).
- Where the names come from (When relevant).
- Which one wins.

Competing names

This `<input>` element has **five possible accessible names** applied.

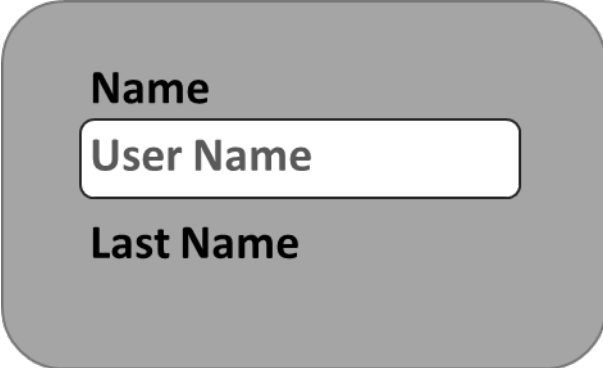
An **aria-labelledby** value of 'Last Name'.

An **aria-label** value of 'Applicant'.

A **<label>** value of 'Name'.

A **title** value of 'First Name'.

A **placeholder** value of 'User Name'.



A diagram of a form element. It consists of a gray rounded rectangle containing a white text input field. Above the input field is the label 'Name'. Below the input field is the label 'Last Name'. The input field contains the placeholder text 'User Name'.


```
<label for="userName">Name</label>
<input id="userName" type="text" title="First Name"
placeholder="User Name" aria-label="Applicant"
aria-labelledby="surname"/>
<span id="surname">Last Name</span>
```

```
<label for="userName">Name</label>
<input id="userName" type="text" t
placeholder="User Name" aria-label
aria-labeledby="surname"/>
<span id="surname">Last Name</span>
```

▼ Computed Properties

▼ Name: "Last Name"

▼ aria-labeledby:

span#surname "Last Name"

~~aria-label: "Applicant"~~

~~From label (for= attribute): label "Name"~~

~~title: "First Name"~~

~~placeholder: "User Name"~~

~~aria-placeholder: Not specified~~

~~title: "First Name"~~

Description: "First Name"

Role: textbox

Invalid user entry: false

Focusable: true

Focused: true

Editable: plaintext

Can set value: true

Multi-line: false

Read-only: false

Required: false

Labeled by: span#surname

The tricky title attribute

You may find the **title** attribute used to **provide an accessible name** for some elements.

However, it is weak and is often **overridden by other methods**.

But the title is **sneaky**! If it loses the accessible name.

IF **aria-describedby** is present, the **title will not be used** as the accessible description.

Too good to be true?

DevTools Accessibility tab is **amazing**. Right?

While **DevTools** Accessibility tab is very useful,
it **does not replace** any **harmonized testing
process** your organization uses.

Time for any questions?

Thank You!

Shawn Garmer
IT Accessibility Specialist
U.S. Department of Education
National Digital Accessibility Team | Office for Civil Rights
shawn.garmer@ed.gov | 202-987-1286