

Decoding label and name for accessibility



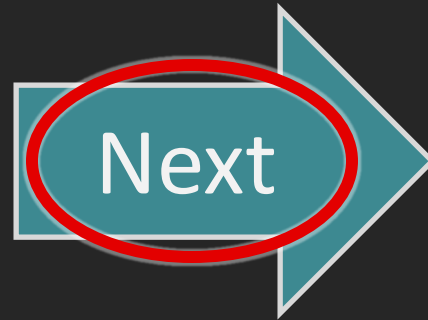
Jon Whiting
WebAIM.org

WCAG 2 “Label” and “Name”

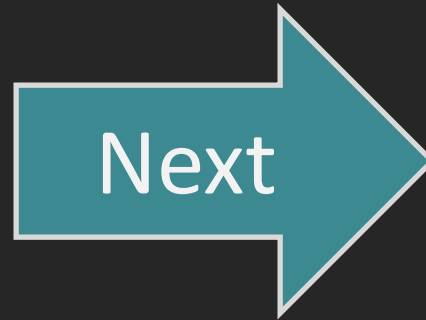
WCAG requires an accessible “label” (2.4.6, 3.3.2) and “name” (1.1.1)

- The **Label** is presented visually
- 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="Next"...>  
</a>
```

Two principles of accessible names

1. An element can only have one accessible name.
2. To be sure of the name, you must test with a screen reader or inspect the HTML.

Tools for testing accessible name

Reveal alternative text for an entire page

- (e.g. [WAVE](#))
 - Strength
 - Shows the whole page at once
 - Stay out of the code
 - Weakness
 - Will not show alt text for all images (only ``)
 - Not always clear what “wins”, Especially if ARIA is being used

Browser Developer tools

E.g., Chrome DevTools

- Strengths
 - Works for more than just `` tags
 - Shows you the thing that “wins”
- Weaknesses
 - Explore one thing at a time
 - Designed for developers

Testing image accessibility

go.nasa.gov/3qLL9Nz

Or Google “NASA #BeAnAstronaut”

“Label” and “Name” are often the same

Code	Label	Name
<pre>Log in</pre>	<u>Log in</u>	“Log in”
<pre><label>First Name: <input type="text"> </label></pre>	First Name: <input type="text"/>	“First Name:”
<pre><button>Activate </button></pre>	Activate	“Activate”

WCAG label and name requirements

- [1.1.1 Non-text Content](#) – Equivalent names for non-text content
- [1.3.1 Info and Relationships](#) – Name presented correctly
- [2.4.6 Headings and Labels](#) – Labels “describe topic or purpose”
- [2.5.3 Label in Name](#) – Accessible name contains the label
- [3.3.2 Labels or Instructions](#) – Labels when “content requires user input”
- [4.1.2 Name, Role, Value](#) – Name for controls can be “programmatically determined”



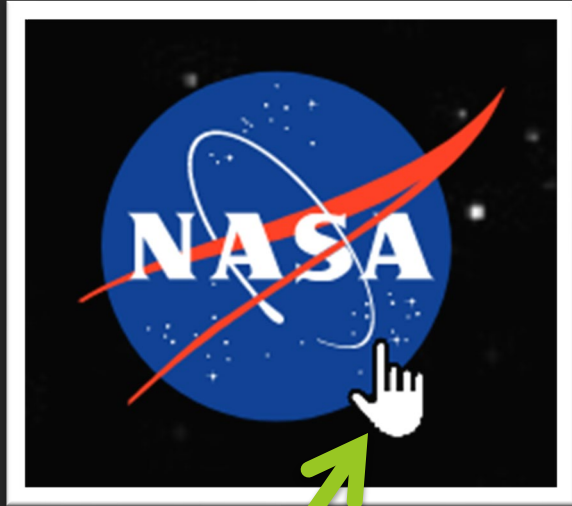
These aren't the SCs you're looking for...

--Patrick H. Lauke

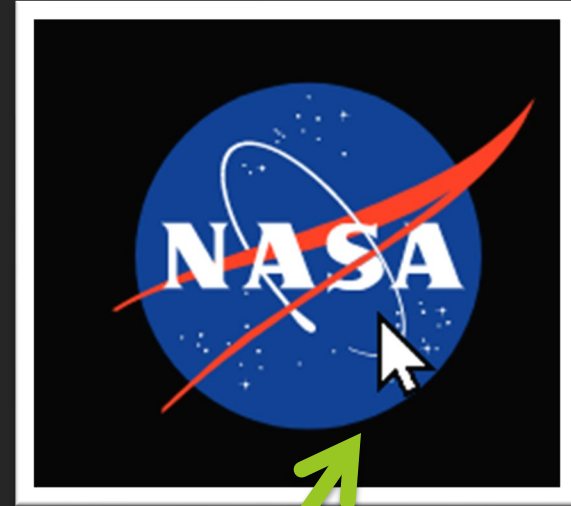
2.5.3 – Label in Name (Level A)

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

“User interface components”

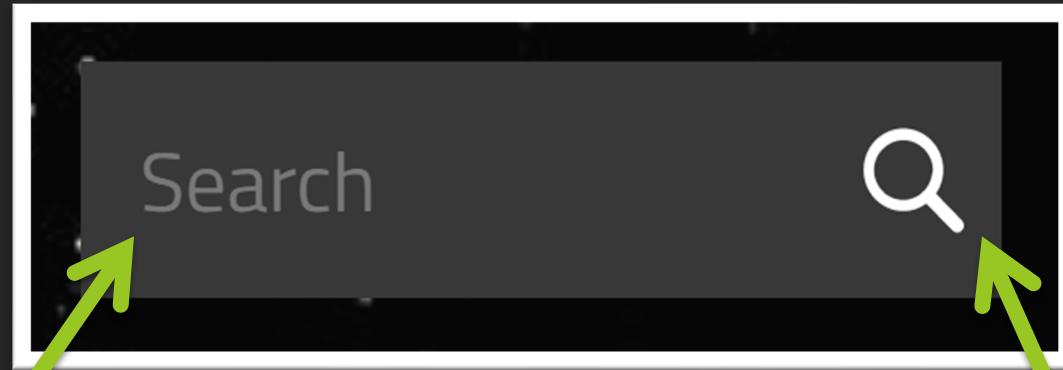


UI component



NOT a UI component

“Labels that **include text** or images of text”

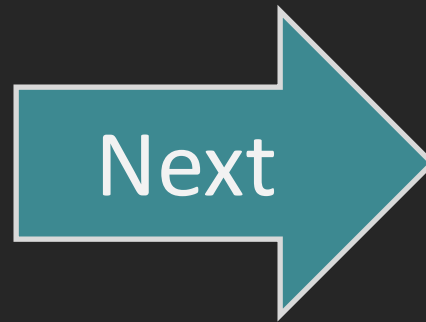


Text in label

No text in label

“The name **contains** the text presented”

Label not in Name – Fails 2.5.3



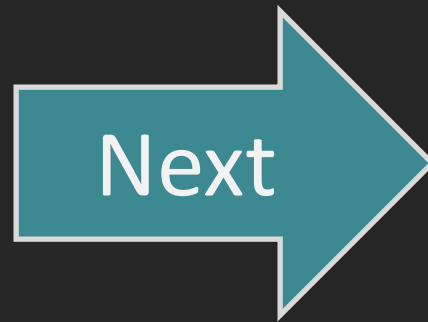
```
<a href="page2.php"...>  
  <img alt="Continue"...>  
</a>
```

Label *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>
```

Does this pass or fail 2.5.3?



```
alt="information"
```

How about this?



```
alt="Nasa"
```

What if you have more than one candidate for the accessible name?

1. If the control has an `aria-label` or an `aria-labelledby` attribute the accessible name is the value of the attribute using the algorithm defined in [Accessible Name and Description: Computation](#).
2. Otherwise use the associated `label` element(s) *accessible name*(s) - if there are multiple, concatenate by DOM order, delimited by spaces.
3. If the *accessible name* is still empty, then: use the control's `title` attribute.
4. Otherwise use the control's `placeholder` attribute.
5. If none of the above yield a usable text string there is no *accessible name*.

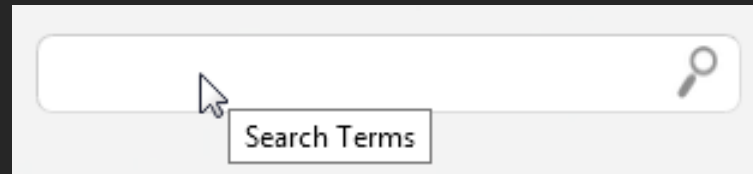
Accessible Name Computation

Form <label>

First Name:

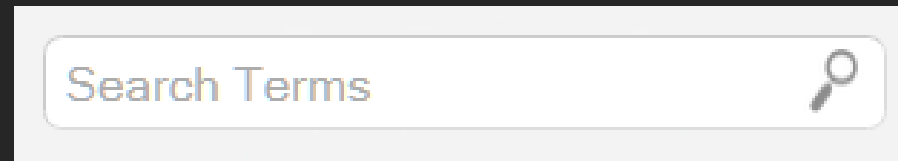
```
<label for="fname">First Name:</label>  
<input type="text" id="fname"...>
```

title attribute (if no <label>)



```
<input title="Search Terms"...>
```

placeholder attribute (if no title or <label>)



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

Placeholder is read by a screen reader (accessible name),
but is not a suitable “label”

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



Card number



Expires

CSC

1/22

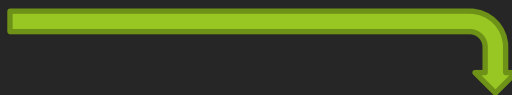


3 digits



Enter a valid expiration date

Add

`<label>` 
First Name:

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

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

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

First Name:

 `aria-labelledby`

```
<span id="fname1abel1">First name:</span>
```

```
<input type="text"  
aria-labelledby="fname1abel1">
```



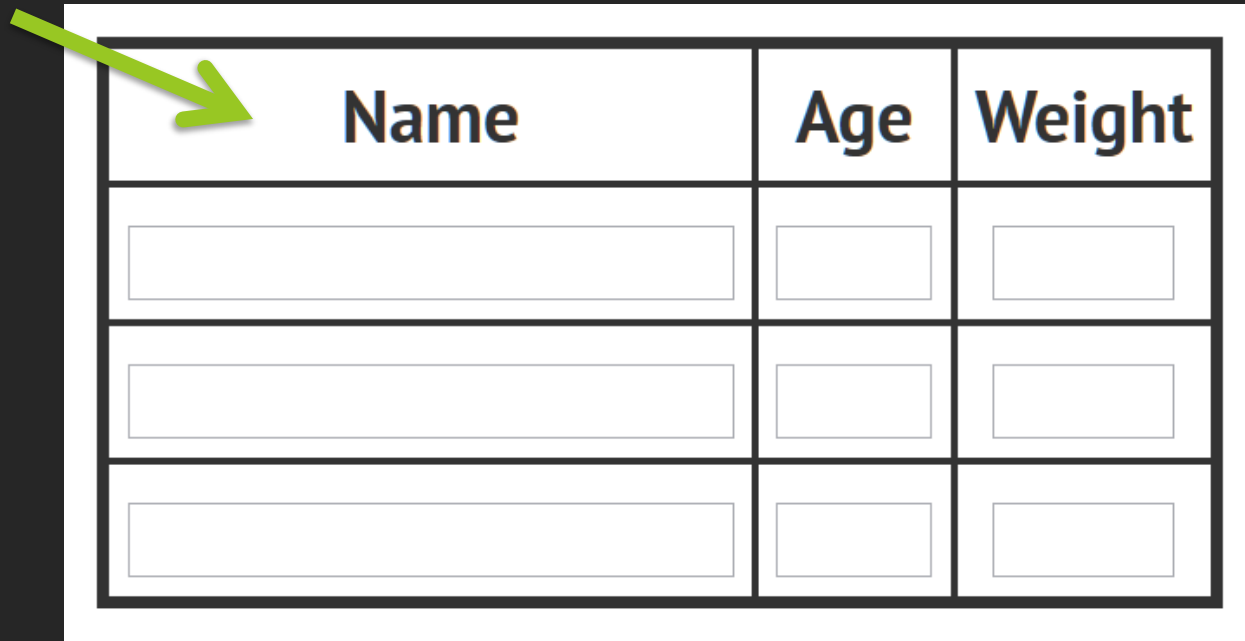
```
<label for="fname" id="fnamelabel">  
First name:</label>
```

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

Unnecessary markup (Rule #1 of ARIA),
but label will only be read once.

One label for multiple controls

`id="nameLabel"`



Name	Age	Weight
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

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

Multiple labels for multiple controls

`id="officenum"`

`id="cyndi"`

Name	Office Number	Phone
Cyndi	<input type="text"/>	<input type="text"/>
Jared	<input type="text"/>	<input type="text"/>
Jonathan	<input type="text"/>	<input type="text"/>

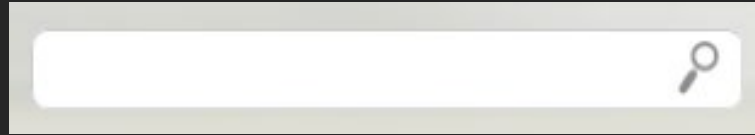
```
<input type="text" name="office1"
aria-labelledby="cyndi officenum">
```

Form values as labels for other inputs

Name	Office Number	Phone
<input type="text" value="Cyndi"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="Jared"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="Jonathan"/>	<input type="text"/>	<input type="text"/>

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


aria-label



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

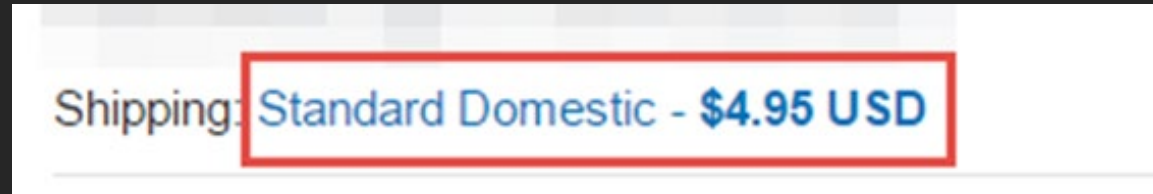
ARIA *labels* override default accessible *names*

WCAG Label → First Name:

Accessible Name

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

...and not just in forms



```
<a... aria-label="Change shipping  
method">Standard Domestic - $4.95 USD</a>
```

Also applies to alternative text, button text, etc.

 cannot have an accessible name

[Download the Employment Application](#) 

```
<a href="app.pdf">Download the Employment Application  
<span class="PDFlink" aria-label="PDF"></span>  
</a>
```

Neither can <div>, <p>, <i>, etc.

Three possible solutions

Download the Employment Application

1. `Download the Employment Application(PDF)`
2. `Download the Employment Application`
3. `Download the Employment Application`

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>
```

Evaluate accessible name and description with Chrome DevTools

1. Use “Inspect Mode Tooltip” (top-left corner) to inspect the form field.
2. Open accessibility tab
3. Review “Accessibility Tree” and “ARIA Attributes”
4. Using “Computed Properties”, find the accessible name and description

Thank You!

webaim.org

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

