

**Rachele DiTullio,**  
Senior Accessibility Engineer

# STRUCTURING ACCESSIBLE FORMS



# RACHELE DITULLIO

(SHE/THEY)

- **2000:** First job as a web developer
- **2012:** Masters degree in Information Studies
- **2019-2020:** Certified Professional in Web Accessibility
- **2021:** Accessibility Engineer



# AGENDA

---

Creating form fields

---

Grouping similar fields

---

Marking required fields

---

Validating data formats

---

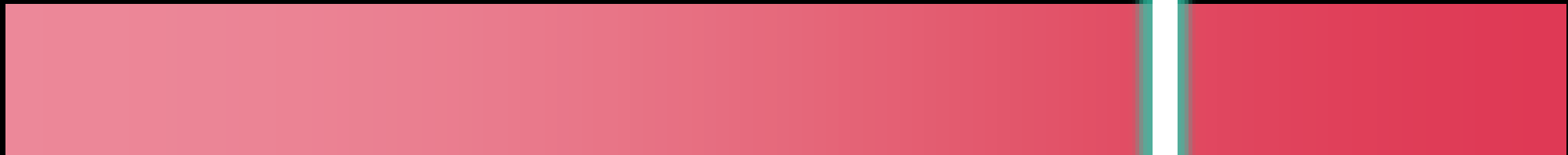
Handling form errors

# WEB ACCESSIBILITY

Web accessibility is the extent to which a website or web application can be used by **disabled people**.

Less

More



# SEMANTIC MARKUP

Native HTML elements first

...then ARIA.

# Register Your Cat

**Cat's name \***

**Your email \***

**Cat's birthday**

Birthday format is *mm/dd/yyyy*.

**Is your cat altered? \***

Not sure  Yes  No

**Cat's markings \***

select an option



**Cat's colors \***

Select at least 1 color.

- Black
- White
- Brown
- Orange
- Gray

# CREATING FORM FIELDS

**Labels** provide visible and accessible names to form **inputs**.

**for** and **id** attributes programmatically link labels and inputs.

# LABEL WITH INPUT

```
<label for="name">Cat's name</label>
```

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



# INPUT TYPES

```
<input type="email" id="email"  
name="email">
```

```
<input type="radio" id="yes"  
name="altered">
```

```
<input type="checkbox" id="orange"  
name="color">
```

Other types include **search**, **url** and **tel**.

# SINGLE SELECT

```
<select id="markings" name="markings">
<option value="">select an
option</option>
<option value="solid">Solid</option>
<option value="multi">Multi-
colored</option>
<option value="tabby">Tabby</option>
.
.
.
</select>
```

# AUTOCOMPLETE ATTRIBUTE

```
<input type="email" id="email"  
name="email" autocomplete="email">
```

**Your email \***

**Cat's birtl** demo1@test.com

# GROUPING SIMILAR FIELDS

**Fieldset** groups related fields, often radio buttons and checkboxes.

**Legend** provides a label for the grouped controls.

# FIELDSET WITH LEGEND

```
<fieldset>  
  <legend>Cat's colors</legend>  
  ...  
</fieldset>
```

# RADIO BUTTONS

```
<label for="yes"> <input name="altered"  
type="radio" id="yes">Yes</label>
```

```
<label for="no"> <input name="altered"  
type="radio" id="no">No</label>
```

```
<label for="maybe"> <input  
name="altered" type="radio" id="maybe"  
checked>Not sure</label>
```

# CHECKBOXES

```
<p><label for="black">  
<input type="checkbox" id="black"  
name="black">Black</label></p>
```

```
<p><label for="white">  
<input type="checkbox" id="white"  
name="white">White</label></p>
```

```
<p><label for="brown">  
<input type="checkbox" id="brown"  
name="brown">Brown</label></p>
```

# MARKING REQUIRED FIELDS

Provide a visible indication of required fields.

Include the either the **required** or the **aria-required="true"** attribute to programmatically mark a field as required.



# REQUIRED & ARIA-REQUIRED

```
<label for="name" ...>  
Cat's name*</label>
```

```
<input type="text" required ...>
```

```
<input type="text" aria-  
required="true" ...>
```

# VALIDATING DATA FORMATS

Add the **pattern** attribute to the `<input>` element and use a regular expression to define the data format.

Include help text indicating the valid data format, e.g. *mm/dd/yyyy*.

# PATTERN ATTRIBUTE

```
<input type="text"  
pattern="^\s*(1[012]|0?[1-9])\/(3[01]|12)[0-9]|0?[1-9])\/((?:19|20)\d{2})\s*$" ... >
```

Birthday format is *mm/dd/yyyy*.

26/05/2020|

Is yo



Please match the requested format.

# ARIA-DESCRIBEDBY ATTRIBUTE

```
<p id="error4" class="help">Birthday  
format is <em>mm/dd/yyyy</em>.</p>
```

```
<input type="text" pattern="..."  
aria-describedby="error4" ...>
```

## Cat's birthday

Birthday format is *mm/dd/yyyy*.

Is your cat altered? \*

Not sure  Yes

No

^ v Done

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
0	<input type="button" value="X"/>	

## INPUTMODE ATTRIBUTE

```
<input  
type="text"  
pattern="..."  
inputmode="numeric"  
...>
```

# NOVALIDATE ATTRIBUTE

```
<form novalidate ...>
```

```
<input type="email" required ...>
```

```
</form>
```

# HANDLING FORM ERRORS

Include the `aria-invalid="true"` attribute on form fields in error.

Provide a visible error message and programmatically link it using the `aria-describedby` attribute.

# ARIA-INVALID ATTRIBUTE

```
<p id="error2">Email format is  
address@domain.com.</p>
```

```
<input type="email" required  
aria-invalid="true"  
aria-describedby="error2" ...>
```



# ERROR MESSAGES

Error messages must be visible, persistent and close to the field in error.

Don't rely on color alone to convey errors. Include an icon with "error" as the text.

# EXAMPLE ERRORS

**Cat's name \***

⚠ Enter cat's name.

**Your email \***

⚠ Email format is *address@domain.com*.

brienne

# SUMMARY

1. Verify each `<input>` element has a defined **type**, **id** attribute, **name** attribute and **autocomplete** attribute (if needed), linked to `<label for="id">`.
2. Group related fields such as radio buttons and checkboxes using `<fieldset>` and `<legend>` elements.
3. Provide data format requirements using the **pattern** attribute.
4. Mark **required fields** visually and programmatically.
5. Add **aria-invalid="true"** to fields in error; programmatically link help and error text.

**Rachele DiTullio**, Senior Accessibility Engineer

**Email:** [racheleditullio@protonmail.com](mailto:racheleditullio@protonmail.com)

**Slides:**

[racheleditullio.com/talks/accessible-forms/](https://racheleditullio.com/talks/accessible-forms/)

**THANK YOU**



# RESOURCES

**Demo:** <https://racheleditullio.com/accessible-form-example/>

**Code Pen:** <https://codepen.io/racheleditullio/pen/oNwZdOg>

**GitHub:** <https://github.com/racheleditullio/accessible-form-example>