Peel Region digital components

Table of contents

Header, font, and colours

Standard header

The Peel Region logo

The logo is available in white and black. Use the white logo on a dark background and the black logo on a light background.

The dimensions are 200px wide by 53px tall. Do not resize the logo.

Peel logo assets

Font

Open Sans with weights of 300, 400, and 600.

- Light (used for summary text): 300
- Normal body text: 400
- Bold: 600

Font sizes

Base font:

Should be the equivalent of 16px / 1em / 100% of browser default.

```
body{
font-size: 100%;
color: #333;
font-family: "Open Sans", Arial, Helvetica, sans-serif;
}
```

Heading 1:

```
h1{
font-size: 2.25em;
font-weight: 600;
margin: 13px 0 5px;
line-height: 1.25em;
```

Heading 2:

```
h2{
color: #1F1F1F;
font-size: 1.5em;
font-weight: 600;
margin: 40px 0 20px;
}
```

Heading 3:

```
h3{
font-size: 1.175em;
font-weight:600;
}
```

Element colours

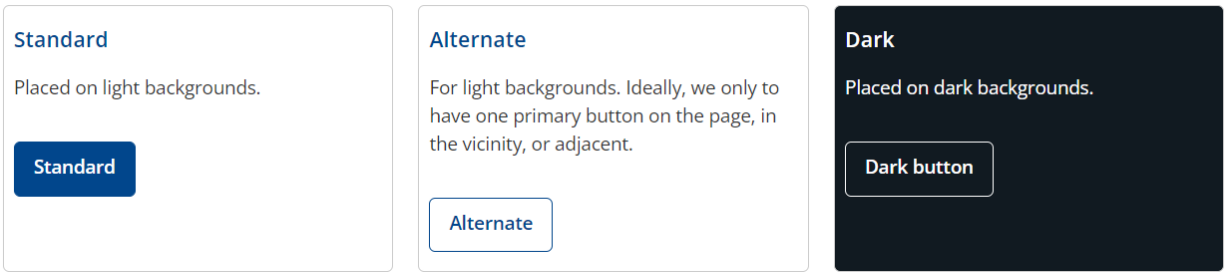
Colour	Name	HEX	RGB	HSL	Sample use
	Dark blue	#01468c	rgb (1 70 140)	hsl (210deg 99% 28%)	Header and footer
	Light blue	#f3f8fd	rgb (243 248 253)	rsl (210deg 71% 97%)	Callout-box background
	Light blue border	#6e91b4	rgb (110 145 180)	hsl (210deg 32% 57%)	Callout-box and border

Buttons

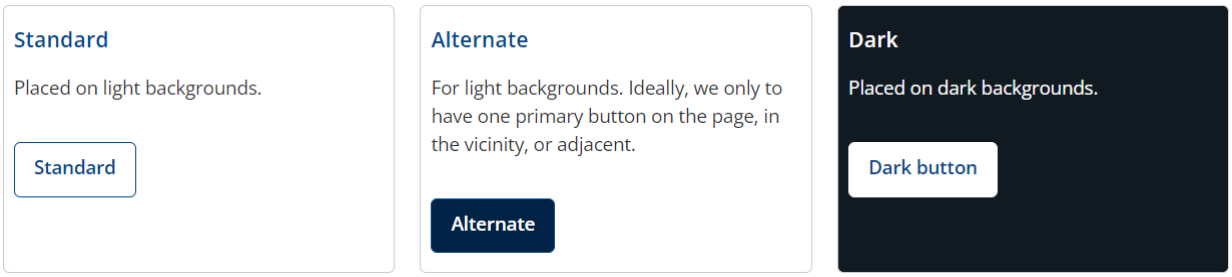
- **Standard:** Used on light backgrounds.
- **Alternate:** For light backgrounds.
- **Dark:** Used on dark backgrounds.

Try not to place more that one primary button next to each other on a page.

Default states



:hover and :focus states



Forms

Building and styling forms to the Peel Region's brand and standards.

User instructions

Instructions that apply to the entire form should be placed before the <form> element to ensure that users are aware of the requirements before they begin. In-line instructions should be placed throughout the form within the label of the form control.

Instructions should always use active voice, not passive. When structuring sentences place the subject before the verb (action word). For example:

- Instead of: "Proof of income must be provided for each business."
- Use: "Each business must provide proof of income."

Write in first and second-person point of view. Use "we" and "you" whenever possible. For example:

- Instead of: "The applicant should check for accuracy."
- Use: "You should check for accuracy."

Avoid using "please" in form instructions or error messages.

Required vs. optional fields

- All fields should be considered required unless labelled as optional.
- Do not indicate required fields by using asterisks or any other method.
- Include the word "optional" in parenthesis at the end and as part of the label text.

Demographic fields

Do not ask the following demographic questions: sex assigned at birth, gender identity, cultural background, language, religion, immigration status, disability, age, education level, employment status, family status, household income, housing status, etc. unless this data is essential for the service to be performed (e.g. a medical form).

Make all form entries undoable

We don't use the *Reset* button, so it is important to offer users an escape route for errors on form entries. For text fields or check boxes, the user can erase the entry and revert to the original state.

When using radio buttons and drop-down lists, the default state should have nothing selected, especially if it's not a required *either/or* question.

Elements

Fieldset

Include: this is a group related fields arranged into logical blocks or sets. A legend tag is a caption for the fieldset.

Placeholder text

Avoid: placeholder text is not a replacement for labels. Even though it elimintes visual clutter, it may cause usability problems.

Field labels

Provide labels to identify all form controls, including text fields, checkboxes, radio buttons, and drop-down menus. In most cases, this is done by using the <label> element. This element explicitly associates a form control with a label.

Standard fields

These fields are common controls on most forms. Where applicable, formats and validation rules are noted. These should be written exactly ash shows, including capitalization.

Field label	Validation	Comment
First name		Enter first and last name separately unless application requires a combined name.
Last name		
Email	Validate for proper email format	Do not label as "address"
Phone number	10 digits	Allow user to enter an unformatted number (i.e., 9995551111) and automatically format as either (999) 555-1111 or 999-555-1111.
Address		Use an API to auto-complete and format address where possible.
Street		
City		
Province	2 letters	Drop-down list of standard values

Field label	Validation	Comment
Postal code	7 characters including space (A1A 1A1)	Allow user to enter unformatted postal code with or without space and automatically format. Auto format upper- or lower-case entries.
Country		
Unit or apt.		Optional
Date	8 digits in YYYY-MM-DD format (2023-03-15)	Use date picker calendar. Ensure the control is accessible on mobile when using a keyboard only.
Date of birth	8 digits in YYYY-MM-DD format (2023-03-15)	Better to allow direct date entry as date pickers typically do not allow easy entry of dates far in the past.
Accessibility accommodations	Text area	Optional. May be required for services delivered in-person

Mobile keyboard for input types

These are subject to change and are easily be controlled with CSS.

Field type	Keyboard	Input type (example)
Text	Regular	<input type="text" id="last_name" name="last_name" value="">
Credit card	Numeric	<input type="text" id="x_card_num" maxlength="16" name="x_card_num" pattern="[0-9]*" value="" autocomplete="cc-number">
Email	Email	<input autocomplete="email" name="email" id="email" type="email" value="">
Phone	Numeric	<input autocomplete="tel" name="phone_number" id="phone_number" type="tel" value="">
Canadian postal code	Regular	<input type="text" id="postal" maxlength="7" name="postal" pattern="[A-Za-z][0-9][A-Za-z] [0-9][A-Za-z][0-9]" value="" autocomplete="postal">
Date	Date picker (regular)	Use a date picker. Depending on the use for date, limit the years. For example, if the event is happening in the future, do not show past dates.

Error handling

Validation error messages

- Display an error message when a user enters data that does not meet the allowed values or validation criteria on that field.
- Display an error message to identify required fields that were not completed.
- Validation should be inline: that is, as soon as the user has finished filling in a field, an indicator should appear nearby if the field contains an error. Don't validate fields before input is complete (e.g. onblur form validation).
- Error messages should be displayed in red. An icon to the left of the error message or validation summary will draw attention to the error and help users who are colour blind.
- Error messages should be easy to understand and should provide simple instructions on how they can be resolved.

Common validation error messages

These should be written exactly as shown, including capitalization.

Scenario	Error message
Required field not completed	[Field label] is required.
Acknowledgement or consent checkbox not checked	You must accept the terms and conditions to continue. You must check the box to continue.
Entry is too long	This number is too long. Enter up to XX digits. This entry is too long. Enter up to XX characters.
Incorrect email format	Enter a valid email address.

Incorrect phone number	Enter a valid phone number.
Incorrect date format	Enter date in YYYY/MM/DD format.
Two field values don't match	Both [field label] should match.
Radio box not selected	Choose an option that applies to you.
Checkbox not selected	Choose at least one option that applies to you.

Submission button

There should only be one primary button in the area such as "submit" or "next." Other buttons should be styled as an alternate. See button.

Do not use a *reset* button. It can hinder users in the following ways:

- Users may click the *reset* button by mistake instead of *submit*, resulting in all their work being removed.
- Multiple buttons increase clutter to the interface and makes it harder for the user to identify their next step.

```
input[type=button], button{
  padding: 8px 10px;
  border: 1px solid transparent;
  background: #036;
  cursor: pointer;
  font-weight: 600;
  color: #FFF !important;
  display: inline-block;
  width: auto;
  margin: 0;
  transition: all ease .1s;
  min-height: 40px;
  top: 1px;
  margin-bottom: 1px;
  position: relative;
  box-sizing: border-box;
  border-radius: 5px;
  text-decoration: none !important;
}

input[type=button]:hover, input[type=button]:focus, button:hover,
button:focus{
  background: #FFF !important;
  color: #054d84 !important;
  border-color: #054d84;
}
```

Alternate button style

The following example demonstrates the second button which has a CSS selector named "hollow" attached to it:

```
input[type=submit].hollow,
input[type=reset].hollow,
input[type=button].hollow,
button.hollow{
  background: none;
  color: #054d84 !important;
}

input[type=submit].hollow:hover,
input[type=button].hollow:hover,
input[type=reset].hollow:hover,
input[type=submit].hollow:focus,
input[type=button].hollow:focus,
input[type=reset].hollow:focus,
button.hollow:hover,
button.hollow:focus {
background:#036 !important;
  color: #FFF !important;
}
```

Dropdown list items

When using list items (<select> field) preselect a black option with no value. This ensures that the user intentionally selects their option rather than one preselected by default. Be mindful of space on a mobile device and try to keep the line content short (about 350px wide).

```
<label for="happy">Are you happy</label>
<select name="happy" id="happy">
  <option selected value=""></option>
  <option value="no">No</option>
  <option value="yes">Yes</option>
</select>
```

Radio group vs checkbox

Radio buttons are used when there is a list of two or more options that are **mutually exclusive** and the user must select exactly one choice. In other words, clicking a non-selected radio button will deselect whatever other button was previously selected in the list. If there are more that about 5 items and their labels are short, consider using a drop-down list (e.g. country list).

Checkboxes are used when there are lists of options. The user may **select any number** of choices, including zero, one, or several. Each checkbox is independent of all other checkboxes in the list, so checking one box does not uncheck the others.

A stand-alone checkbox is used for a single option that the user can turn on or off.

Form components

- **Fieldset**
  Groups related information into logical blocks or sets.
- **Legend**
  Defines a caption for the <fieldset> element.
- **Label**
  The "for" attribute of <label> must be equal to the id attribute of the related element to bind them together. A label can also be bound to an element by placing the element inside the <label> element.
- **Radio**
  The field is wrapped in the label.
- **Checkbox**
  The field is wrapped in the label.
- **Drop-down lists**
  Default selection should be blank and/or a choice with no value.

References

- Accessibility in forms: Form instructions | Web Accessibility Initiative (WAI) | W3C
- Designing better error messages UX – Smashing Magazine
- How to report errors in forms: 10 design guidelines (nngroup.com)