

Information Technology Policy

Pennsylvania Design Standards (PDS)

ITP Number OPD-SFT002A	Effective Date April 1, 2020
Category Software	Supersedes None
Contact RA-ITCentral@pa.gov	Scheduled Review March 2021

1. Purpose

Operations document to ITP-SFT002 *Commonwealth of PA Design Standards* that details the design requirements necessary for all Commonwealth of Pennsylvania managed websites and digital services.

2. Design Standards

2.1. Pennsylvania Design Standards (PDS) Core

Refer to <https://www.pa.gov/design-standards/index> for details and downloading of the design standard collection of tools.

2.2. Components

Components form the basic building blocks of the PDS. Each of the components include the work component, a code snippet detailing its construction, usage considerations detailing when and when not to use an item, general usability considerations, and accessibility requirements.

2.2.1. Accordions

A list of headers that hide or reveal additional content when selected.

When to Use

- If the citizen only needs a few, specific, contextual bits of information within a page.
- If you only have a small space to display a lot of content.
- If you are using an accordion menu, make sure to use on a page as part of a sidebar, not part of the modal or other nested elements.

When Not to Use

- If citizens need to see all the content on the page, do not use the accordion.
- If there is not enough content to warrant its use.
- If you are using an accordion menu, use a regular sidebar menu if you don't have any child pages that would exist in the accordion.

Usability

- If you are using the accordion menu, the chevron is the button to activate the show/hide. Do not make the link headline into the trigger for the show/hide.
- You can make the headline link to another page without activating the trigger to show/hide sub-links.

- Use the Accordion Menu Unstyled as part of the side navigation pattern when you have site navigation that includes child pages under a parent page. This would replace the list view of the side navigation links with a more robust navigation system for mobile.
- Use the Accordion Menu as part of the sidebar if you require robust sidebar navigation more than just a list of links. Each link should go to a unique page. The chevron should toggle the list of child pages only. When you open a link from the accordion menu that does have child pages, do not automatically expand the accordion.
- The FAQ Accordion is to be used in main content areas or sidebar areas when/where applicable. The FAQ Accordion is intended for question and answer type of content. You can include links, images, and other rich media when appropriate.
- When appropriate (six or more accordion sections), use the Primary Button component labeled "Expand All" `title="Expand All Accordion Sections"` after the accordion appears on the page.
- Do not add the "Expand All" button to the Accordion Menu or Accordion Menu Unstyled, use it only for Accordion Container and FAQ Accordion.
- Ensure the "Expand All" button does not create recursive patterns for citizens to get stuck in the Accordion Container and FAQ Accordion.

Accessibility

- Code headline links in the accordion as buttons.
- Using a `<button>` assures that accordions are usable with both screen readers and the keyboard.
- Use Accessible Rich Internet Applications (ARIA)-expanded on buttons to express an accordion's default state.
- Buttons should state if they are expanded by default with `aria-expanded="true"`. The `aria-expanded="false"` attributes will be added to other buttons when the accordion is initialized by the JavaScript.
- Use unique ids.
- Each button has a unique name `aria-controls="id"` that associates the control to the appropriate region by referencing the controlled element's id.
- The accordion uses JavaScript to set the aria-hidden value of its content area.
- Each content area will have its aria-hidden attribute set to either true or false by the component, depending on its corresponding button's aria-expanded attribute. To ensure that your content is accessible in the event that the JavaScript does not load or is disabled, you should not set `aria-hidden="true"` on any of your content areas.

Refer to <https://www.pa.gov/design-standards/components#accordions> for Accordions examples and code snippets.

2.2.2. Alerts

An alert keeps citizens informed of important and sometimes time-sensitive changes.

When to Use

- Emergency Messages - As a notification required to inform citizens of an emergency situation.
- System Status Messages - As a notification required to let citizens know the status of system they are engaged with, typically for outages and updates.
- Validation Messages - As a notification required to let citizens know of missing information or successful completion of a task.

When Not to Use

- Long forms - Use the inline validation for form controls instead of using an alert.

Usability

- Consider next steps. When the citizen is required to do something in response to an alert, let them know what they need to do and make that task as easy as possible. Think about how much context to provide with your message. For example, a notification of a system change may require more contextual information than a validation message. Write the message in concise, human readable language; avoid jargon and computer code.
- Be polite in error messages — don't blame the citizen.
- Alerts are an opportunity. Citizens will read a message that helps them resolve an error even if they generally won't read documentation; include some educational material in your error message.
- Don't overdo it. Too many notifications will either overwhelm or annoy the citizen and are likely to be ignored.
- Allow a citizen to dismiss a notification wherever appropriate.
- Understand the citizen's context. Don't include notifications that aren't related to the citizen's current goal.

Accessibility

- Use the proper Accessible Rich Internet Applications (ARIA) role. If the message is not interactive, use the `ARIA role="alert"` to inform assistive technologies of a time-sensitive and important message. If the message is interactive, use the `ARIA role="alertdialog"` instead.
- Don't visually hide alert messages and then make them visible when they are needed. Citizens using older assistive technologies may still be able to perceive the alert messages even if they are not currently applicable.

Refer to <https://www.pa.gov/design-standards/components/alerts> for Alerts examples and code snippets.

2.2.3. Breadcrumbs

Breadcrumbs indicate the current page's location within a navigational hierarchy.

When to Use

- Breadcrumbs are a secondary navigation element that identifies to a citizen their location on a website or application. Breadcrumbs are a visual representation of the site's hierarchy of information. Use on content pages that fall under logical information architecture.

When Not to Use

- Breadcrumbs shouldn't exist on standalone pages that aren't logically grouped within an information architecture (marketing landing pages, standalone page applications, etc.).
- They shouldn't appear on the homepage.

Usability

- The breadcrumbs in this design system are location-based, which is a representation of the site's information architecture. Location-based breadcrumbs are less confusing and more usable than path-based breadcrumbs which would represent the citizen's history navigating a web site.
- Breadcrumbs should be complementary to primary navigation and should never be used to replace overall navigation elements. Breadcrumbs are subtle and should not take visual prominence over the page. All items in the breadcrumb list should be links except for the last item.

Accessibility

- To assist in wayfinding, add the attribute `aria-label="breadcrumb"` to the `<nav>` element, and add the attribute `aria-current="page"` to the last item of the set.

Refer to <https://www.pa.gov/design-standards/components#breadcrumbs> for Breadcrumbs examples and code snippets.

2.2.4. Buttons

Buttons are visually prominent calls to action.

When to Use

- Important actions. Use buttons for the most important actions you want citizens to take on your site, such as Download, Sign up or Log out.

When Not to Use

- Linking between a site's pages. Use regular links instead.
- If the action is less popular or less important. Less popular or less important actions may be visually styled as links.

Usability

- Give an important action a distinctive style. Style the button most citizens should click in a way that distinguishes it from other buttons on the page.

- Make sure buttons look clickable. Use color variations to distinguish static, hover, and active states.
- Avoid using too many buttons on a page.
- Use sentence case for button labels.
- Keep button text short. Button text should be as short as possible with action words that clearly explain what will happen when the button is selected (for example: Download, View, or Sign up).
- Lead with a verb. Make the first word of the button's text a verb. For example, instead of Complaint filing label the button File a complaint.
- When you have a page that takes up two entire viewport heights (and more), add the back to top button and enable smooth scrolling. Have the back to top button fade in using JavaScript and have it appear after a citizen scrolls past the first viewport height (below the fold) and have it disappear when a citizen is within the first viewport height (above the fold).

How to Use

- Primary Button
 - Use on full-color backgrounds.
- Secondary Button
 - Use on photographic backgrounds.
 - Always use a secondary button for the search button.

Accessibility

- Buttons should display a visible focus state when citizens tab to them.
- Use standard markup. Avoid using `<div>` or `` tags to create buttons. Screen readers don't automatically know either is a usable button.
- Screen readers handle buttons and links differently. When styling links to look like buttons, remember that screen readers handle links slightly differently than they do buttons. Pressing the Space or Enter key triggers a button. Pressing the Enter key triggers a link.

Refer to <https://www.pa.gov/design-standards/components#buttons> for Buttons examples and code snippets.

2.2.5. Cards

Cards are containers used to organize high-level content.

When to Use

- A card should only contain a single idea which may feature a call-to-action, or the option to navigate to more detailed content. The content of a card should be concise and offer only a preview of detailed content.

When Not to Use

- Do not nest navigation within cards, use the appropriate component for navigation.
- Do not nest cards within cards; add cards to modals only when it makes logical sense.

- Do not use if you have more than one idea/call-to-action to display in a card.

Usability

Elements that can be used in a card are:

- Main Title - A short and clear label that states the card's main message.
- Meta information - Used to organize the cards. This can be a time stamp, category, etc.
- Description - A more thorough explanation of the idea and call-to-action. This text should be brief.
- Image or Rich Media - Associated with the card and text, an image or rich media should contribute to the message.
- Buttons - When a card requires a selectable action, or you want to emphasize that the card is a clickable object, you can use buttons.
- Extra information - Any additional information that enhances the message of the card.
- Not all of these elements are necessary for every card. You can choose the elements which you think best fit the purpose of the card. However, every card should have a main title. If multiple cards are placed on a page, they should be placed 16px apart on all sides.

Accessibility

- Follow the accessibility guidelines for the elements listed above as they all apply for use in a card.

Refer to <https://www.pa.gov/design-standards/components#cards> for Cards examples and code snippets.

2.2.6. Form Controls – Checkbox

Checkboxes allow citizens to select one or more options from a visible list.

When to Use

- When a citizen can select any number of choices from a setlist.
- Toggles: When a citizen needs to choose “yes” or “no” on only one option (use a stand-alone checkbox). For example, to toggle a setting on or off.
- Visibility of options. When citizens need to see all the available options at a glance.

When Not to Use

- If there are too many options to display on a mobile screen.
- Single-select only. If a citizen can only select one option from a list (use radio buttons instead).

Usability

- Make the label selectable. Citizens should be able to tap on or click on either the text label or the checkbox to select or deselect an option.
- List options vertically. Horizontal listings can make it difficult to tell which label pertains to which checkbox.

- Use positive statements. Negative language in labels can be counterintuitive. For example, use “I want to receive a promotional email” instead of “I don’t want to receive promotional email.”
- Use adequate touch targets. If you customize, make sure selections are adequately spaced for touch screens.

Accessibility

- Customize accessibly. If you customize the text inputs, ensure they continue to meet the accessibility requirements that apply to all form controls.
- Use a fieldset and legend for a checkbox group. Surround a related set of checkboxes with a `<fieldset class="form-group">`. The `<legend>` provides context for the grouping. Do not use fieldset and legend for a single check.
- These custom checkboxes are accessible. The custom checkboxes here are accessible to screen readers because the default checkboxes are moved off-screen with `position: absolute; left: -999em`.
- Use semantic ids. Each input should have a semantic id attribute, and its corresponding label should have the same value in its `for=""` attribute.
- The title attribute can replace `<label>`.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – Checkbox examples and code snippets.

2.2.7. Form Controls – Date Input

Single date input appropriate for most dates.

When to Use

- Appropriate for most dates. Use this format for most dates, particularly memorized dates.

When Not to Use

- Date ranges. Do not use to capture date ranges.

Usability

- Semantic Markup. Follow the Date Input attributes.
- Avoid dropdowns. It may be tempting to switch all or some of these text fields to dropdowns, but these tend to be more difficult to use than text boxes.

Accessibility

- Follow input guidance. These text fields should follow the accessibility guidelines for all text inputs.
- Don’t auto-advance focus. Do not use JavaScript to auto-advance the focus from one field to the next. This makes it difficult for citizens using keyboard navigation to correct mistakes.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – Date Input examples and code snippets.

2.2.8. Form Controls – File Upload

A specialized button for attaching files.

When to Use

- When a form requires file attachment(s) to be included.

When Not to Use

- Avoid using single file upload more than once, use upload multiple files.

Usability

- Use labels to denote what file types to accept and apply appropriate error messaging for non-supported file attachments.

Accessibility

- Use the `for=""` attribute to denote what the label is used for, such as `for="custom_file"`.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – File Upload examples and code snippets.

2.2.9. Form Controls – Radio

Radio buttons allow citizens to see all available choices and select exactly one.

When to Use

- Single selection: When citizens need to select only one option out of a set of mutually exclusive choices.
- Limited choices: When the number of available options can fit onto a mobile screen.

When Not to Use

- Multiple selections: Consider checkboxes if citizens need to select more than one option or if there is only one item to select.
- Limited space: Consider a dropdown if you don't have enough space to list out all available options.
- Selecting none: If citizens should be able to select zero of the options.

Usability

- Use the label as a target. Citizens should be able to tap on or click on either the text "label" or the radio button to select or deselect an option.
- List items vertically. Options that are listed vertically are easier to read than those that are listed horizontally. Horizontal listings can make it difficult to tell which label pertains to which radio button.
- Use adequate spacing. If you customize, make sure selections are adequately spaced for touch screens.
- Set default values with caution. Setting a default value can discourage citizens from making conscious decisions, seem pushy, or alienate citizens who don't fit into your

assumptions. If you are unsure, leave nothing selected by default.

Accessibility

- Customize accessibly. If you customize the radio buttons, ensure they continue to meet the accessibility requirements that apply to all form controls.
- Use fieldset and legend. Group related radio buttons together with `<fieldset class="form-group">` and describe the group with `<legend>`.
- Use proper labels and attributes. Each radio button should have a `<label>`. Associate the two by matching the `<label>`'s for attribute to the `<input>`'s id attribute.
- The title attribute can replace `<label>`.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – Radio examples and code snippets.

2.2.10. Form Controls – Select

Dropdowns allow citizens to select one option from a temporary modal menu.

When to Use

- Use sparingly. Use the dropdown component only when a citizen needs to choose from about seven to 15 possible options, and you have limited space to display the options.

When Not to Use

- Fewer than seven options. Use radio buttons instead.
- More than 15 options. If the list of options is very long. Let citizens type the same information into a text input that suggests possible options instead.
- If you need to allow citizens to select more than one option at once. Citizens often don't understand how to select multiple items from dropdowns. Use checkboxes instead.
- Site navigation - Use the navigation components instead.

Usability

- Test dropdowns thoroughly with members of your target audience. Several usability experts suggest they should be the "UI of last resort." Many citizens find them confusing and difficult to use.
- Avoid making options in one dropdown menu change based on the input to another. Citizens often don't understand how selecting an item in one impacts another.
- When most citizens will (or should) pick a particular option, make it the default: `<option value>Default</option>`.
- Don't use JavaScript to automatically submit the form (or do anything else) when an option is selected. Offer a "submit" button at the end of the form instead. Citizens often change their choices multiple times. Auto-submission is also less accessible.

Accessibility

- If you customize the dropdown, ensure it continues to meet the accessibility requirements that apply to all form controls.
- Make sure your dropdown has a label. Don't replace it with the default menu option (for example, removing the "State" label and just having the dropdown read "Select a state" by default).
- Avoid auto-submission.
- Don't use JavaScript to automatically submit the form (or do anything else) when an option is selected. Auto-submission disrupts screen readers because they select each option as they read them.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – Select examples and code snippets.

2.2.11. Form Controls – Text Area

A large input meant for capturing qualitative information.

When to Use

- Use in forms where capturing more in-depth information is required.

When Not to Use

- Do not use in place of a text input.

Usability

- Give the text area at least three lines worth of input space with the ability to drag the bottom right corner larger. Make sure to include validation to prevent input of code.

Accessibility

- Most browsers' default rendering of placeholder text does not provide a high enough contrast ratio.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – Text Area examples and code snippets.

2.2.12. Form Controls – Text Input

Text inputs allow citizens to enter any combination of letters, numbers, or symbols. Text input boxes can span single or multiple lines.

When to Use

- Unpredictable or freeform responses. If you can't reasonably predict a citizen's answer to a prompt and there might be wide variability in citizens' answers.
- Input simplicity. When using another type of input will make answering more difficult. For example, birthdays and other known dates are easier to type in than they are to select from a calendar picker.

- Pasted content. When citizens want to be able to paste in a response.

When Not to Use

- Predetermined input options. When citizens are choosing from a specific set of options.

Usability

- Use fields appropriate to the length of the input. The length of the text input provides a hint to citizens as to how much text to write. Do not require citizens to write paragraphs of text into a single-line input box; use a text area instead.
- Consider the mobile context. Text inputs are among the easiest type of input for citizens using desktops but are more difficult for citizens using mobile.
- Wait to validate. Only show error validation messages or stylings after a citizen has interacted with a particular field.
- Avoid using placeholder text that appears within a text field before a citizen starts typing. If placeholder text is no longer visible after a citizen clicks into the field, citizens will no longer have that text available when they need to review their entries. (People who have cognitive or visual disabilities have additional problems with placeholder text.)
- There are two classes provided for input messaging; success and error. Those classes are available to display the appropriate message to the citizen with the correct styling.
- Each input should be surrounded by an element with the class name of `field-group` (in this case, the `<fieldset>` element is used). When a specific error or success is needed to be displayed related to that form element, include a `` inside the parent class `field-group`, and extend this parent with a class of `-error` or `-success` for their respective visuals.

Accessibility

- If you customize the text inputs, ensure they continue to meet the accessibility requirements that apply to all form controls.
- Most browsers' default rendering of placeholder text does not provide a high enough contrast ratio.
- Avoid breaking numbers with distinct sections (such as phone numbers, Social Security Numbers, or credit card numbers) into separate input fields. For example, use one input for a phone number, not three (one for area code, one for local code, and one for number). Each field needs to be labeled for a screen-reader and the labels for fields broken into segments are often not meaningful.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – Text Input examples and code snippets.

2.2.13. Form Controls – Time Input

Single time input appropriate for most times.

When to Use

- Appropriate for most times. Use this format for most time.

When Not to Use

- Time ranges. Do not use to capture a range of time.

Usability

- Semantic Markup: follow the Time Input attributes.
- It may be tempting to switch all or some of these text fields to dropdowns, but these tend to be more difficult to use than text boxes.

Accessibility

- Follow input guidance: These text fields should follow the accessibility guidelines for all text inputs.
- Don't auto-advance focus: do not use JavaScript to auto-advance the focus from one field to the next. This makes it difficult for citizens who only use their keyboard to navigate and correct mistakes.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – Time Input examples and code snippets.

2.2.14. Form Controls – Validation

Stating validation requirements upfront, with live feedback, means citizens won't be left guessing.

When to Use

- Generally, use the validation component for account creation tasks, or as a list of errors for a form in addition to form validation errors.

When Not to Use

- They are meant to convey pertinent information about a particular task, not be to used as an addition to a card or put multiple types of call to actions in them.
- Do not put an alert in a modal or card.

Usability

- Limit the contents to the following:
 - Header
 - Paragraph
 - List
- Always include a close icon. Allow the citizen to close the validation component even if they still have pertinent tasks to complete.

Accessibility

- Follow the accessibility guidelines for the elements listed above as they all apply for use in a card.

Refer to <https://www.pa.gov/design-standards/components#form-controls> for Form Controls – Validation examples and code snippets.

2.2.15. Grid Framework

The grid system is based on Foundation’s Flex Grid. The Foundation 6 Flex Grid works very similarly to their standard float grid but includes a number of useful features only possible with flexbox, like horizontal and vertical alignment, automatic sizing, and easier source ordering. The flex grid is only supported in Chrome, Firefox, Safari 6+, IE10+, iOS 7+, and Android 4.4+. Flexbox is supported in Android 2, but not reliably enough for use with this grid.

When to Use

- Follow the 2% browser usage rule; if a browser still has 2% general usage, we will support that browser. This grid system has been chosen to support older browsers without significant polyfills.

When Not to Use

- Should always use this grid system. Deviation to another system can have adverse effects on using the design standards in your project.

Usability

- Please refer to Foundation’s Documentation on Flex Grid (<https://get.foundation/sites/docs/flex-grid.html>)

Accessibility

- When using the grid system, follow Foundation 6’s site documentation for any new developments regarding accessibility.

2.2.16. Modals

Modals are used to display content in a layer above the app/website.

When to Use

- Deliberate disrupting action. Use modals when appropriate to disrupt a task/action when the citizen wants to explore another task.
- Provide a lot of contextual information without taking up main screen space.

When Not to Use

- Choose the right component. If you're using a Modal for a system alert that the citizen must acknowledge, consider using a validation component (for forms) or an alert.
- Do not use for show/hide content. Use an accordion component.

- No pop-ups. It's very easy to make modals into timed pop-ups. Do not force citizens to engage with a modal they haven't triggered.

Usability

- They are as wide as 50% of the viewport but include a minimum and maximum width to avoid going too narrow or too wide. Modals always have an equal amount of space at the top and bottom to account for the height of the close button.

Accessibility

- Modal has `role="dialog"`.
- When the modal is open, everything behind it has HTML attribute `aria-hidden="true"`, so assistive technology won't read out the underlying page. The best way to do this is to give the modal and the page separate wrapper elements and toggle `aria-hidden="true"/aria-hidden="false"` on the main page's wrapper depending on whether or not the modal is open.
- Modal has an `aria-labelledby` attribute whose value is the id of the modal's heading.
- Follow the accessibility guidelines for the elements nested within a modal.

Refer to <https://www.pa.gov/design-standards/components#modals> for Modals examples and code snippets.

2.2.17. **Pagination**

Pagination splits content into multiple discrete pages.

When to Use

- When you have a lot of tabular data or a stepped process with cards that would benefit from using pagination navigation. Breaking the content/data list into multiple pages stops the citizen from feeling overwhelmed. A citizen can see the size of the data set, so they know how much more content there is to investigate. Pagination also allows the citizen to feel more in control of the content they are viewing as they can decide whether or not to click to the next page. Every click should lead the citizen closer to achieving their goal.

When Not to Use

- Do not use as a navigation replacement to go to different pages altogether, keep the content and data logically organized.

Usability

- Through pagination, a citizen should be able understand the amount of content there is; you should include a link to the last page of the content in the pagination bar, unless the data set fluctuates in size. Also include a link to the first page on every page.
- Each page should contain the same amount of objects. You may include an option for the citizen to change how many objects are shown on each page.
- The citizen should be able to tell clearly on which page they are on. Place this number in the middle of the pagination bar and give it the heaviest visual weight.

The citizen should also be able to see what pages they have already viewed so they do not have to revisit content that they have already seen. You can show this by changing the color, typically a secondary color, of already visited pages versus pages the citizen has yet to visit.

- Use the Pagination Simple component when you have less than five sections of content to navigate through.
- If you have more than five sections of content that you will have the need for pagination, use the Pagination Advanced component and utilize its ability to direct links to sections of content using the numbered buttons.
- When you are using the Pagination Advanced Component and a citizen is in the middle of paging through the sections, always show the current page you are on and have numbered buttons to the previous three sections, the next three sections, and include a direct link to the first section and the very last section.

Accessibility

- Place the pager in a `<nav>` element when you can.
- If you do not use a `<nav>` element, you need to add `role="navigation"` to the markup. Note: this role is implied when you use the `<nav>` element so it is a bit redundant to use both at the same time.
- The markup includes an `aria-label="pagination"` to describe the type of navigation.
- Add `aria-current="page"` to the link that points to the current page. This will tell Assistive Technology that the focused link is pointing to the current page.
- Add `aria-disabled="true"` to the link when the link is disabled.

Refer to <https://www.pa.gov/design-standards/components#pagination> for Pagination examples and code snippets.

2.2.18. Tables

A table shows tabular data in columns and rows.

When to Use

- Tabular data. When you need tabular information, such as statistical data.

When Not to Use

- Non-tabular data. Depending on the type of content, consider using other presentation formats such as definition lists or hierarchical lists.

Usability

- Keep it simple. Tables are great at displaying tabular data. Minimal visual styling helps surface this information more easily.

Accessibility

- Simple tables can have two levels of headers. Each header cell should have `scope="col"` or `scope="row"`.

- Complex tables are tables with more than two levels of headers. Each header should be given a unique id and each data cell should have a headers attribute with each related header cell's id listed.
- When adding a title to a table, include it in a `<caption>` tag inside of the `<table>` element.

Refer to <https://www.pa.gov/design-standards/components#tables> for Tables examples and code snippets.

2.2.19. Tags

Tags can focus attention on important content or content that might otherwise be missed.

When to Use

- To indicate what information being displayed is associated with a certain category (example: "Policy" would be a tag that would appear at the top of the search results page when a search query for "government home inspection policy" would be returned).

When Not to Use

- Confusion with buttons. Avoid tags if they might appear in the same area of the page as buttons.
- New or updated content. To call attention to new or updated content, consider changing the background color of the object itself or experiment with changing the font-weight.
- When citizens already expect content to be updated frequently, for example on a site. For example, on a site dedicated to breaking news. In this case, placing the new content at the top may be enough.

Usability

- Citizens frequently confuse tags as buttons. Always conduct usability testing to make sure your particular implementation is not causing frustration.
- If your tags aren't interactive, disable hover, focus, and active styles.
- Don't mix interactive and static tags. Once you establish a pattern for how tags behave on your site, citizens will expect that behavior every time.
- Don't overdo it. If everything on a page is called out as important, nothing commands unique attention.

Accessibility

- Use Accessible Rich Internet Applications (ARIA) live regions to highlight dynamically loaded content. When tags are used to call out new content that is dynamically loaded onto a page, be sure to use ARIA live regions to alert screen readers of the change.

Refer to <https://www.pa.gov/design-standards/components#tags> for Tags examples and code snippets.

2.3. Patterns

Patterns are collections of commonly used components that form familiar sections for page layouts. These patterns were created to introduce consistency across the Commonwealth of Pennsylvania government websites. New patterns can be created with the current list of components.

2.3.1. Footers

A consistent footer containing links to key services within the Commonwealth.

When to Use

- The enterprise footer is used on all public facing websites. This pattern is comprised of distinct links, the Commonwealth seal, and a subfooter with additional links.

When Not to Use

- If you are building a website for internal usage, the enterprise footer may not be appropriate. Please review your project requirements.

Usability

- Follow the code snippet as a guide to build the enterprise footer. Text links are styled differently in the footer, so take note in your styling. Test the footer in mobile to ensure proper spacing and reflow. Do not add any FontAwesome Icons unless directed in this documentation.
- The subfooter bar links are set to 12px instead of the default 18px/1rem.

Accessibility

- Semantic HTML is required to build the footer. Review the code snippet for further details.

Refer to <https://www.pa.gov/design-standards/patterns#footer> for Footer examples and code snippets.

2.3.2. Hero Patterns

A typical first pattern familiar to citizens coming to a website.

When to Use

- Always use some form of hero banner. Your project may detail a particular layout for the central background hero section, but always include the official top header, main nav, and its associated mobile styling.
- Typically, the hero banner is unique on the homepage and uniform on interior pages.

When Not to Use

- Unless specifically mandated, do not reuse the homepage hero banner on interior pages.

Usability

- Follow the usability guidelines for each individual component to ensure conformance.
- Do not turn the hero banner into a carousel.
- Only use the Secondary Button in a hero banner.

Accessibility

- Follow the accessibility guidelines for each individual component to ensure conformance.

Refer to <https://www.pa.gov/design-standards/patterns#hero-patterns> for Hero Patterns examples and code snippets.

2.3.3. Interior Patterns

When to Use

- Typically, the interior hero banner is uniform and can be used for a main page of associated content. It's important to work with an information architect to determine when and where this pattern should be used in a site hierarchy.

When Not to Use

- Do not use in place of Hero Patterns.

Usability

- Follow the usability guidelines for each individual component to ensure conformance.
- Do not turn the interior hero banner into a carousel.
- Only use the Secondary Button in an interior hero banner.

Accessibility

- Follow the accessibility guidelines for each individual component to ensure conformance.

Refer to <https://www.pa.gov/design-standards/patterns#interior-patterns> for Interior Patterns examples and code snippets.

2.3.4. Navigation – Main Navigation

Hierarchical, horizontal wayfinding.

When to Use

- Always use this pattern for public websites. It is the definitive pattern for creating top navigation.

When Not to Use

- In rare instances, this pattern shouldn't be used for unique services or landing pages.

Usability

- If a top link has a dropdown list of child links, the top link is activated on hover, not click.
- If a top link has a dropdown list of child links, include a chevron icon to visually indicate it has child links.
- If a top link has a dropdown list of child links, it may also link to a new page if clicked on.
- Do not duplicate the top link in the list of child links.
- It is preferable to make the focus states for the main navigation the same as a hover states.
- Always include the Official Top Header pattern about the Main Navigation when creating an official PA.gov site or service.

Accessibility

- Follow accessibility guidelines for creating links and search inputs.

Refer to <https://www.pa.gov/design-standards/patterns#navigation> for Navigation - Main Navigation examples and code snippets.

2.3.5. Navigation – Main Navigation Mobile

Top navigation reflowed for smaller devices.

When to Use

- Utilized when a site/service screen resolution is 900px wide and smaller.

When Not to Use

- At full-screen size where all of the navigation items can comfortably be displayed on the page.

Usability

- Only use for mobile navigation.
- Top links that have child links underneath it, are indented, and expanded open to show the entire navigation.

Accessibility

- Follow the accessibility guidelines for links.

Refer to <https://www.pa.gov/design-standards/patterns#navigation> for Navigation - Main Navigation Mobile examples and code snippets.

2.3.6. Search

A binary pattern of input and results.

When to Use

- Include the search input in the navigation as detailed in the Navigation pattern. Pages denoted for search results should use the search results component.

When Not to Use

- Always include the search input and search results page unless you are creating a one page site.

Usability

- Unless otherwise directed, do not create a page with a dedicated search input, it is redundant from the main navigation.
- Only use the search results component for search results.

Accessibility

- Follow the accessibility guidelines for text inputs and buttons to ensure conformance with those components.

Refer to <https://www.pa.gov/design-standards/patterns#search> for Search examples and code snippets.

2.4. Styles

The PDS uses a type system to maximize readability, accessibility, and page layout for a multitude of devices.

2.4.1. Address

Addresses are used to list physical locations.

When to Use

- Address elements are used specifically for mailing/shipping/billing addresses or the physical location of a building/lot/land.

When Not to Use

- Do not use the address element to denote anything else but a physical address.

Usability

- Use logically within the content.
- Work with a content strategist to help set address placement and standards.

Accessibility

- Use the `<address>` element instead of a stylized `<p>` element.

Refer to <https://www.pa.gov/design-standards/styles#address> for Address examples and code snippets.

2.4.2. Blockquote

A verbatim statement described differently from other content.

When to Use

- Interviews and news content, blockquotes make the most sense in these types of content.
- Dramatic content breaks. Blockquotes are used to help break up content on the page in conjunction with other page elements such as images, videos, and other components.

When Not to Use

- No quotable content. Do not use the blockquote for content other than a quote.

Usability

- Don't overuse them. Add them for effect, not to utilize for every quote.
- A blockquote includes verbatim content in quotes. Outside of the quote is a dash with the quote person/entity with an optional next line to denote their title or other unique identifiers.

Accessibility

- Use the `<blockquote>` element. When citing an entity or using a quote from another location, add `cite="add url"` in the `<blockquote>`.

Refer to <https://www.pa.gov/design-standards/styles#address> for Address examples and code snippets.

2.4.3. Color Palette

When to Use

- Always use the PDS color palette for official pa.gov sites and services.

When Not to Use

- Default is to use the color palette unless specifically directed otherwise for your project.

Usability

- `.base-offwhite` is for backgrounds with dark text (`.base-darkgray`).
- `.base-darkgray` is for backgrounds with light text (`.base-offwhite`).
- `.alert-green` and `.alert-red` must use `.base-white` for text.
- `.alert-gold` must use `.base-black` for text.
- Any `.alert color` is only used for alerts.

- `.link-focus` is the color used for focus states.
- `.base-lightgray` is used in navigation with `.brand-blue` being the color of links in the navbar. `.brand-blue` is also the hover state color of the nav links with the link text turning to `.base-white`.

Accessibility

- WCAG 2.1 AA and AAA Color Combinations.

Refer to <https://www.pa.gov/design-standards/styles#color-palette> for Color Palette examples and code snippets.

2.4.4. Dividers

Use dividers to define thematic breaks between paragraphs.

When to Use

- Dividers are used to separate content on a page, and are used primarily to break up long pages of text, or to separate titles from other content.

When Not to Use

- Table substitution. Do not use dividers as a way to create tables.

Usability

- Judiciously use on long content pages to help break up the content section.

Accessibility

- Do not add any additional ARIA or additional attributes to the `<hr>`.

Refer to <https://www.pa.gov/design-standards/styles#dividers> for Dividers examples and code snippets.

2.4.5. Headers

Section titles that structure content.

When to Use

- Page titles. The page title will always be an `<h1>` element.
- Page header. After the use `<h1>` page title is used, the next level element will be `<h2>` for in content page headers.
- Subheads. After the `<h2>` element is used for page headers, use an `<h3>` for subheads.
- Additional headers. For especially long content structures, you may use the additional `<h4>`, `<h5>`, and `<h6>` elements, though they should be rarely utilized.

When Not to Use

- You must use the `<h1>`, `<h2>`, and occasionally the `<h3>` for establishing page structure.

Usability

- Improper order and nesting. Your page should always have an `<h1>` starting off the page content. Do not put any other header elements before it. Do not mix up the logical use of headers, they should always descend on the page.
- Multiple `<h2>` can exist with children headers. You can have several `<h2>` elements on the page, just make sure that any subhead `<h3>` elements are properly nested under them. The existence of other `<h2>` elements on the page denotes multiple sections of separate content blocks with their own subheads.
- The default font is Montserrat. Do not use Fungo unless specifically directed so in your requirements.

Accessibility

- Write clear header titles.
- Do not add any other Accessible Rich Internet Applications (ARIA) attributes to the header elements.

Refer to <https://www.pa.gov/design-standards/styles#headers> for Headers examples and code snippets.

2.4.6. Iconography

Icons are used in conjunction with content to communicate meaning.

PDS uses a subset of Font Awesome icons on the project (<https://fontawesome.com/>). They are available for use in `src/img`. If you need additional icons, use Font Awesome in your project. There are some mandatory icons that are part of select components that are packaged in the source files as well.

When to Use

- When and where custom icons are detailed in the usage of a certain component.
- Use of icons is limited in content and only when appropriate to your project.
- How to Use: use PDS icons available in `src/img` or add Font Awesome to your project. PDS uses icons in two ways:
 - SVG in HTML `img` elements.
 - SVG in CSS backgrounds.

When Not to Use

- Do not use icons to replace content that can be written out.

Usability

- Refer to Icon Usability from Nielsen Norman Group for more details: (<https://www.nngroup.com/articles/icon-usability/>)

Accessibility

- Provide descriptive text when an icon is meaningful or has an action associated with it. Do not provide additional text when the icon is decorative.
- See Images - Accessibility for Teams (<https://accessibility.digital.gov/front-end/images/>) and Accessibility - Font Awesome (<https://fontawesome.com/how-to-use/on-the-web/other-topics/accessibility>).

Refer to <https://www.pa.gov/design-standards/styles#iconography> for Iconography examples and code snippets.

2.4.7. Image Caption

Descriptive text that appears under photos and illustrations.

When to Use

- Contextual caption. Add an image caption to an image/photo/illustration when it makes sense for your content strategy.

When Not to Use

- External interactive content. Do not use with `<iframes>` to external content.

Usability

- Make sure to add descriptive alt text to your image and ensure your caption aligns to your alt text. You can use a caption to add more content that wouldn't be appropriate for an alt text description.
- Use both. You can give your image good alternative descriptions and extended captions.

Accessibility

- Add a caption by wrapping an `` or `<canvas>` element with a `<figure>` element and adding a `<figcaption>`.
- Write good `alt=""` alternative description for images. Refer to Moz's article on alt text for more information (<https://moz.com/learn/seo/alt-text>).
- Use the following documentation for images:
 - Be precise: The alt text should describe the content in the image accurately.
 - Keep it concise: A short phrase or sentence will be enough for many images.
 - Imagine you closed your eyes and you couldn't see the image. What description is necessary for the reader to understand the information presented in the image?
 - If an image is simply decorative, then the alternative text can just be the word "null" or "decorative."
 - If you have infographics or images of text (e.g., charts and tables), the alt text should state all of the information provided. Alternatively, you can provide a text version of the information.
 - Avoid using "image of" at the beginning of the description. It's not necessary since the screen reader will already announce that it is an image.

- Refer to <https://www.pa.gov/design-standards/styles#image-caption> for Image Caption examples and code snippets.

2.4.8. Links

Links connect citizens to a different page or further information.

When to Use

- In context. When and where it makes sense to link to another resource.

When Not to Use

- Buttons and Link styling. Do not mix link styling with button styling.

Usability

- Visual Indication. Always underline your links.
- External Links. Add the external link icon at the end of your link if it goes to a resource outside of pa.gov. Open those links in new tabs.
- Internal Links. Open internal links in the same tab.
- Contextual Link Labels. Don't make links labeled as "Click Here" or "Click Me", give contextual link labels.
- When linking to an internal document (such as MS Word Document or a PDF), at the end of the link text add parentheses with the applicable file extension for download.
- When linking to an external document that is not part of the Commonwealth of Pennsylvania's document repository, also add parentheses with the applicable file extension after the link text and before the external link icon.
- Acceptable formats for Microsoft Office documents: (docx), (pptx), (xlsx), and other MS Office output files.
- Acceptable format for PDF files: (pdf).
- When linking to specific and unique output files for download, consult the content developers of those files for the applicable file extension to use and denote in hyperlinks.

Accessibility

- Citizens should be able to tab to navigate between links.
- Citizens should be able to activate a link when pressing 'Enter' on their keyboard.
- Citizens should be able to identify links without relying on color alone.
- Citizens should be able to activate hover and focus states with both a mouse and a keyboard.

Refer to <https://www.pa.gov/design-standards/styles#links> for Links examples and code snippets.

2.4.9. Lists

Lists organize information into discrete sequential sections.

When to Use

- Ordered list: Use an ordered list when you need to display text in some ranking, hierarchy, or series of steps.
- Unordered list: Use unordered lists to display text in no specific order.

When Not to Use

- Narratives. If you need to communicate long lists of narrative text.

Usability

- Use sentence case and begin lists with a capital letter.
- Use punctuation appropriate to the text. Do not leave sentences without periods.

Accessibility

- Do not add any additional Accessible Rich Internet Applications (ARIA) attributes to list elements.
- Add appropriate semantic markup and/or ARIA attributes for items in the ``.

Refer to <https://www.pa.gov/design-standards/styles#lists> for Lists examples and code snippets.

2.4.10. Paragraph

A block of text.

When to Use

- Narrative content. If you need to create text on the page, use a paragraph.

When Not to Use

- If it's supporting content structures. Do not style paragraphs to mimic any of the other components in this system (tables, FAQ accordions, lists, etc.).

Usability

- Use paragraph blocks, not one paragraph. Be sure to use both `<p>` and `</p>` to encapsulate a paragraph. Do not use `
` elements to create paragraphs.
- Headers are not paragraphs. Do not wrap a paragraph with a header element instead of an `<p>` element.

Accessibility

- Do not add any additional Accessible Rich Internet Applications (ARIA) attributes to the `<p>` element.

Refer to <https://www.pa.gov/design-standards/styles#paragraph> for Paragraph examples and code snippets.

2.4.11. Scripts – Subscript / Superscript

Subscripts appear below the text around it.

Superscripts appear elevated from the text around it.

When to Use

- Formulas. Typically used in mathematical and scientific formulas.
- Citations. Using a superscript wrapped [] with a link can create citation links.

When Not to Use

- Limited use. Do not use unless your content requires them.

Usability

- Use their respective `<sup>` and `<sub>` elements to create them.
- When used as a citation, follow the same usability and accessibility guidelines for links.

Accessibility

- Do not add any additional Accessible Rich Internet Applications (ARIA) attributes to the superscript and subscript elements.
- When used as a citation link, follow the same accessibility guidelines for links.

Refer to <https://www.pa.gov/design-standards/styles#scripts> for Scripts examples and code snippets.

2.4.12. Typography

Type formatting for websites.

When to Use

- There are specific rules and use cases for font weights and styles on the elements they are used in.

When Not to Use

- The only deviation from using Montserrat and the associated rules would be to use the typeface Fungo in specific, directed use as outlined in your project.

Usability

- Montserrat Light Italic - Use in the Blockquote element (the quote).
- Montserrat Medium - Use in the Blockquote element (the source).
- Montserrat Regular - Use in body copy, see paragraph element.
- Montserrat Regular Italic - Use in body copy and the Image Caption element.

- Montserrat Bold - Use in body copy and in headlines, see header elements.
- Our REM system is based on 18px instead of 16px. In order to correlate a specific pixel size to REM value, use this formula: (X/18). X is the desired pixel size divided by 18. The resulting number is your REM value.

Accessibility

- Do not add additional Accessible Rich Internet Applications (ARIA) attributes to the `<p>`, ``, and `` html elements.
- The base body copy text is 18px, which forms the new basis for our REM measurements. This aids in the readability of the text.
- Unless directly specified in a style element or pattern, all body copy and type setting is set to 18px/1rem.

Refer to <https://www.pa.gov/design-standards/styles#scripts> for Scripts examples and code snippets.

2.5. Website Accessibility Checklist

To maintain compliance with Federal and State regulations, agencies are to comply with all Accessibility requirements. The use of an accessibility checklist is a useful tool for agencies to utilize to ensure compliance.

Refer to <https://www.pa.gov/design-standards/accessibility-checklist> for details on Accessibility Checklists.

3. Publication Version Control

It is the user's responsibility to ensure they have the latest version of this publication, which appears on <https://itcentral.pa.gov> for Commonwealth personnel and on the Office of Administration public portal: <http://www.oa.pa.gov/Policies/Pages/default.aspx>. Questions regarding this publication are to be directed to RA-ITCentral@pa.gov.

This chart contains a history of this publication's revisions. Redline documents detail the revisions and are available to CWOPA users only.

Version	Date	Purpose of Revision	Redline Link
Original	04/01/2020	Base document	N/A