



Government
of Canada

Gouvernement
du Canada

Canada

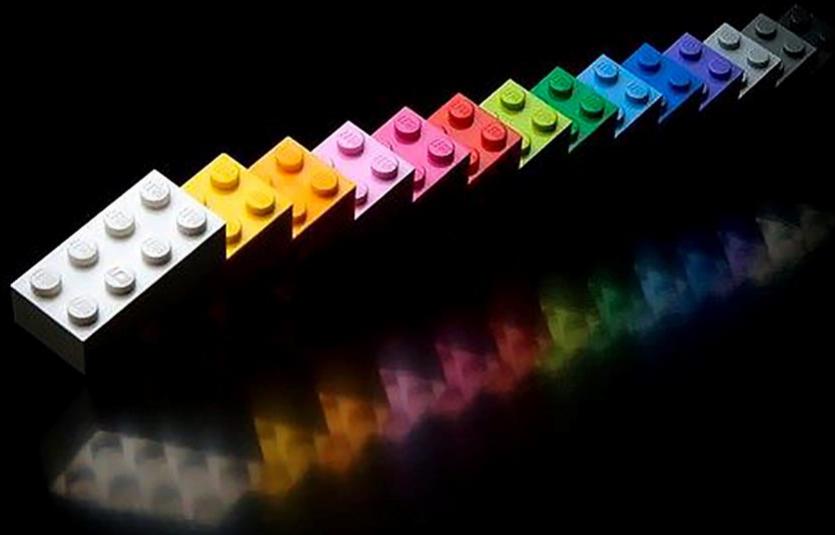
PMV de système de conception

Cas d'utilisation

[Slack de conception du GC](#)

dto.btn@tbs-sct.gc.ca

XXXX 2022

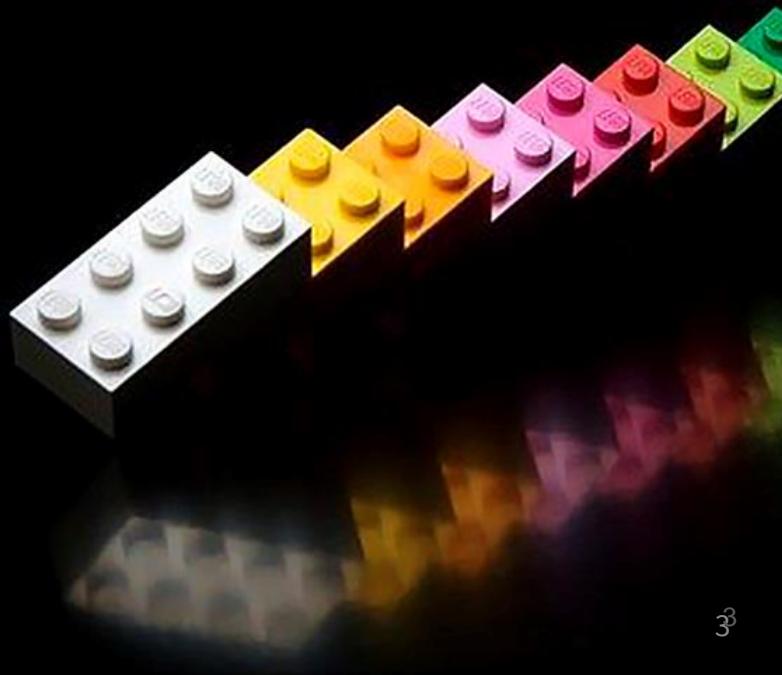


- Définition d'un système de conception
- Produits livrables et approche du PMV
- PMV du système de conception du GC
 - Installation et création faciles et rapides (npm)
 - Documents qui regroupent une orientation sur le code, la conception, le contenu, le style et les éléments stratégiques en un seul emplacement
 - Guide d'accessibilité particulier (y compris pour les besoins cognitifs)
 - Jetons de conception
- Prochaines étapes

Qu'est-ce qu'un système de conception?

Un système de conception est une collection de **conceptions souples et réutilisables, d'éléments de code et de conseils qui sont soutenus par de la recherche.**

Elle fournit aux équipes les éléments de base dont elles ont besoin pour concevoir leur propre présence en ligne de façon à soutenir la réussite des tâches et la confiance des utilisateurs.



Système de conception du GC

Une expérience accessible et cohérente est prévisible. Elle permet aux gens de se sentir en sécurité. Elle permet de renforcer la confiance.

Investir dans un système de conception crée une **orientation intégrée** à l'aide d'un **code moderne et uniforme**. Cela signifie que nous pouvons :

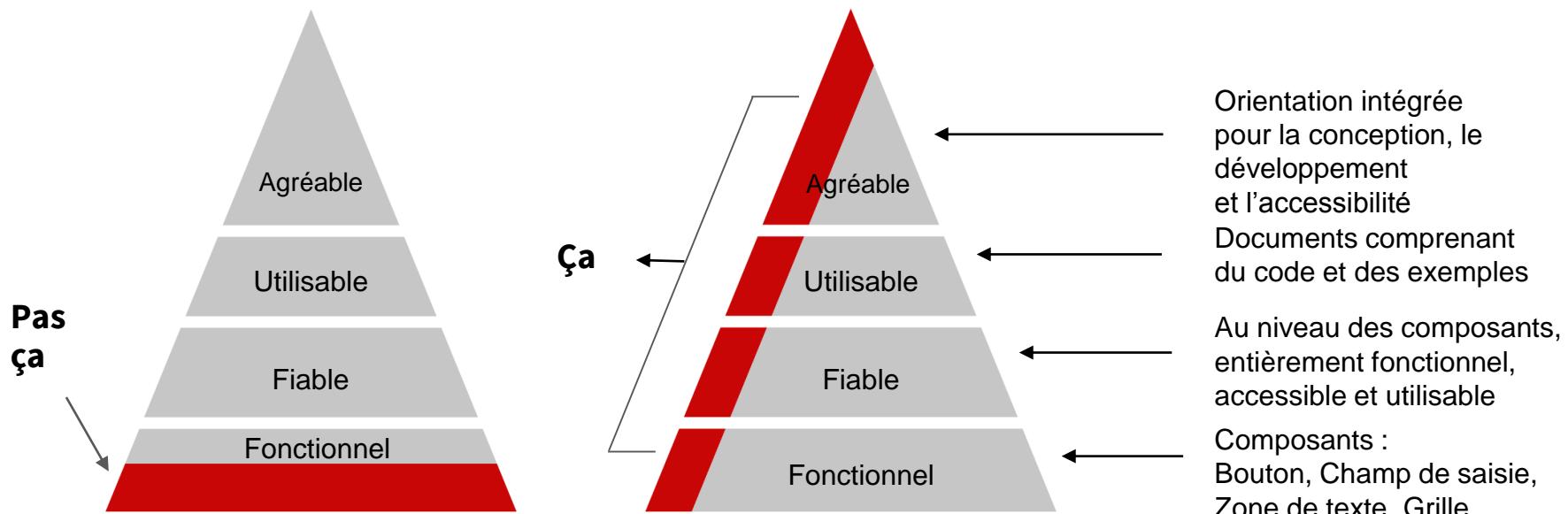
- créer de nouveaux services sans créer de dette de technologie ou de conception;
- accélérer le temps de conception et de développement pour répondre plus efficacement aux besoins changeants du public;
- améliorer l'accessibilité;

The screenshot displays the Web Experience Toolkit (WET) website, which is a collaborative open-source project led by the Government of Canada. It features a navigation bar with links for "WET project", "Implement WET", and "Contribute to WET". Below the navigation is a section titled "Web Experience Toolkit (WET)" with a brief description and a bulleted list of features. A "Resources" section follows, showing examples of the toolkit's implementation. Two main examples are shown: "GCWeb, the WET-BOEW Canada.ca theme" and the "Canada.ca design system". Both examples show a screenshot of a government website header with the Canadian flag, "Government of Canada" text, and a search bar. The GCWeb theme example includes a "Download GCWeb theme v9.4.1" button. The Canada.ca design system example includes a "Search Canada.ca" field and a "Français" link. At the bottom, there are three columns of links: "Template and design pattern library", "Canada.ca Content Style guide", and "Canada.ca Content and Information Architecture Specification"; "Continuous improvement of web content", "Canada.ca blog", and "Latest changes to the Canada.ca design system"; and "Canada.ca Content and Information Architecture Specification".

Approche de PMV

Notre approche de développement du produit minimum viable (PMV) visait deux objectifs :

1. Fournir la valeur rapidement
2. Favoriser la rétroaction pour la prochaine itération



Produits livrables du PMV

Éléments techniques

- Système de jeton de base
- Composantes de base de l'interface utilisateur
 - Neutres et novatrices
- Système de code conçu pour la modularité

Documents

<https://design.alpha.canada.ca/fr/>

- Bases
 - Jetons – typographie
- Pages des composants
 - Code et conseils
- Orientation :
 - Pour les développeurs et les concepteurs
 - pour l'accessibilité et la convivialité

Bases

Installation

Avantages de npm pour l'installation

- Convivialité pour l'éditeur et le développeur
 - Gratuit
 - Source libre
 - Aucune inscription ou connexion nécessaire
- Commandes uniques pour publier ou télécharger

Cadres d'application pris en charge

- JavaScript
- React
- Vue

Installation

Install from npm

```
npm install gcds-components
```

Supported frameworks

The gcds-component library works in multiple frameworks.

JavaScript

Place the following code in the <head> element of your site.

```
<script type="module">
  import { defineCustomElements } from '/node_modules/gcds-components/loader';
  defineCustomElements();
</script>
<link rel="stylesheet" href="/node_modules/gcds-components/dist/gcds/gcds.css">
```

All gcds-components should now be ready to use in your site.

React

Place the following code in the index.js file of your app.

```
import { applyPolyfills, defineCustomElements } from 'gcds-components/loader';
import 'gcds-components/dist/gcds/gcds.css';

ReactDOM.render(...);
```

Bases

Paramètres des éléments de base qui créent la marque globale

Structure :

- De quoi s'agit-il, et pourquoi est-il utile?
- À quoi cela ressemble dans le **code** et la **hiérarchie sémantique** (le cas échéant)
- **Spécifications** détaillées
- Orientation sur l'**utilisation** de l'élément et sa **personnalisation** dans des situations uniques

Typography tokens

Typography is the presentation of text. It includes fonts, sizing, and spacing as applied to the style, arrangement, and appearance of letters, numbers, and symbols.

 [GitHub](#)  Figma — coming soon

Typography design tokens

Default fonts, and styled size settings make digital products responsive, predictable, and readable across devices and platforms.

Decisions about typography values (fonts, sizes, weights) are built into typography tokens in the GC Design System for a unified design.

Font defaults

Font family tokens:

- \$gcds-font-families-heading
- \$gcds-font-families-body
- \$gcds-font-families-monospace
- \$gcds-font-families-icons

Font sizes are configured with rem units. All font sizes are relative to base size 20px.

Display	Font type	Size (rem)	Size (px)	Weight	Line height
Heading 1	Lato	2.5	40	bold	128%
Heading 2	Lato	2.25	36	bold	144%
Heading 3	Lato	2.0	32	bold	162%

Composants

Pages des composants

- Améliorer la communication entre les communicateurs, les concepteurs et les développeurs
- Accès rapide aux outils de conception et de développement
- Modèle expérimental de navigation locale
- Orientation et considérations en matière d'accessibilité intégrées
- Sections distinctes conçues pour des publics distincts :
 - Public général
 - Développeurs
 - Concepteurs
 - Apprenants

Composante en bref

« Papillon »
l'expérience de
navigation
locale « Dans
cette page »
vous permet de
rechercher ce
dont vous avez
besoin

On this page

- [The basic button component](#)
- [How to modify the button component](#)
- [What problems buttons solve](#)
- [Related components](#)
- [Where to place a button](#)
- [How to write a good button label](#)

Button

<gcds-button>

A button is an interactive object that highlights an important or common action for the person using your product.

 [GitHub](#) ⌂ Figma – coming soon

La balise HTML promeut un **langage commun** entre les utilisateurs techniques et non techniques

L'accès à un ensemble d'outils complémentaires favorise la **collaboration interdisciplinaire**

Contenu général

- Brève description :
 - Ce que c'est
 - À quoi ça sert
- Exemple visuel (pour chaque type, le cas échéant)
- Conseils « Utiliser » et « Éviter »
- Afficher et copier les options de code

The basic button component

Use buttons for user-led actions.

Buttons have built in states that change when someone interacts with that button: default, hover, focus, active, disabled.

Primary button



Primary

Use a primary button for the most important action.

Use it for critical actions in a flow or as the default button.

Avoid more than one main call to action on the same page. It reduces impact and makes it harder for a person to know what to do next.

[View code](#)[Copy code](#)

Secondary button



Secondary

Use a secondary button for supporting actions.

Public de développeurs

- Fournit aux développeurs les détails nécessaires pour **ajuster les composants** dans des **circonstances exceptionnelles**, afin de répondre au contexte de cas marginaux
- Comprend la **hiérarchie** des attributs et des propriétés
- Liens vers la section **Bases** pour les spécifications de typographie, de couleur et d'espacement

How to modify the button component

Occasionally, the default options may not meet the needs of the people using your product. You can modify the button component to create a custom button for a particular use case.

Button component type, style, and role

Button type

Set the button type based on how the button will act.

The button-type attribute, accepts the following options:

- button
- submit
- reset
- link

Button role

The buttonRole property, button-role attribute, accepts the following options:

- destructive
- primary
- secondary
- skip-to-content

Public d'apprenants

Pour les personnes qui :

- découvrent la composante
- utilisent rarement la composante
- veulent en savoir plus sur d'autres façons d'utiliser la composante
- souhaitent étudier d'autres options

What problems buttons solve

Use a button for important user-led actions like:

- saving, deleting, copying, or downloading
- clearing entered data
- sending a form or request
- giving consent or agreement
- making a binary choice
- entering or exiting a secure area login, a form, or tutorial
- skip-to-content link

In apps, buttons replace text links for things like:

- controlling movement, like going from one screen to another in a flow
- external links (in a way that lets someone know they're external)

Related components

Radio buttons or **dropdowns** when you can give someone a set of options for single selections.

Checkboxes when you can give someone a set of options for multi selections.

Public de concepteurs

Conseils relatifs aux suivants :

- la conception d'interactions;
- la conception du contenu;
- l'accessibilité.

Where to place a button

Place buttons in a regular and predictable way to help a person find the thing they want to do and take action.

Avoid making a person needlessly scroll, tab, or type by providing the path to the next action early.

Check the button border's visibility against the surface where you're placing it.

Adding skip-to-content buttons

Let a person skip a cluster of navigation links with the skip-to-content button.

To avoid covering up content, configure the button to push down content so it's not floating. For desktop, place skip-to-content button at top left of the page so it doesn't interrupt the flow.

How to write a good button label

Make buttons readable

Keep label text visible by stating the button's specific action in minimal words.

Use sentence case.

Aide à la conception de formulaires

Orientation générale avec la perspective de conception de service

- Axé sur les concepteurs de formulaires
- Explique comment faire fonctionner les composants ensemble
- Comprend les liens afficher et copier le code
- Offre une orientation de base aux développeurs qui doivent concevoir des formulaires

Provide a clear path to completion

Only select the form fields you need

Some questions to ask when you're deciding which fields to include on a form:

- Who needs the data?
- What are they using it for?
- Is it essential or just nice to have?

Not a required field? Just remove it altogether. It's probably not that important.

Choose form elements that match your use case

Choose form components that fit the use case for your data needs. Prioritize the user experience.

Consider the mental models of someone completing your form. They might be familiar with answering questions a certain way. Try to do what would be most comfortable for them.

Pages des composants de formulaire

Orientation sur l'ordre des champs et des messages, ainsi que sur les pratiques exemplaires, fondées sur la recherche

- **Les modèles de champs de formulaire** permettent d'assurer l'accessibilité
- **Conseils** pour rédiger de bonnes étiquettes et des messages d'erreur
- **Le code de composant** comprend les propriétés par défaut pour une création rapide et normalisée

On this page

[The basic text area component](#)

[How to modify the text area](#)

[What problems text areas solve](#)

[Related components](#)

[Size and placement of text area](#)

[How to write good text area labels](#)

[When to use hint text and error messages](#)

Text area

<gcds-textarea>

A text area is a space, usually on a form, for a person to enter long-form information in response to a question or instruction.

 [GitHub](#)  Figma — coming soon

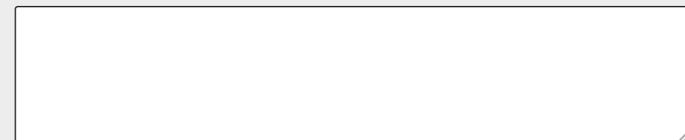
The basic text area component

Use a text area to collect multi-line information from a person.

Text areas have built in states that change when someone interacts with that field: default, hover, focus, active, disabled.

Text area

Text area



Use the text area for longform, multi-line, response collection.

[View code](#)

[Copy code](#)

Sous le capot

Jetons de conception

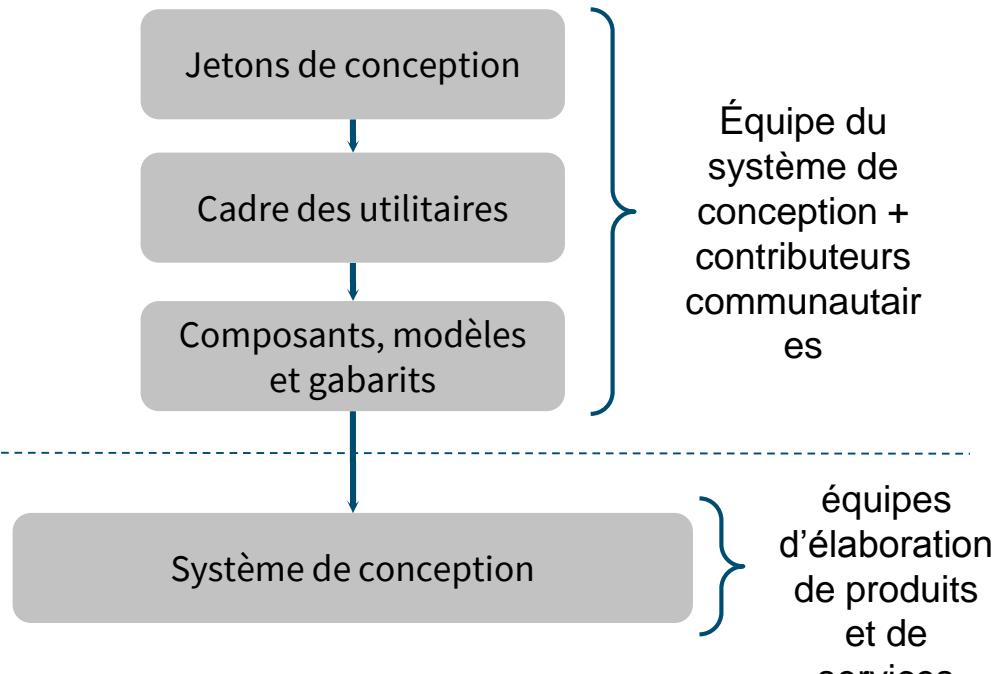
Les jetons de conception sont un moyen de créer de **petits morceaux de code réutilisable** qui centralisent les décisions concernant l'expérience visuelle – par exemple, les styles, la typographie, l'espacement.

Ils **codifient les décisions de conception** afin que les gens n'aient pas à résoudre des problèmes de conception courants comme le contraste de couleur ou l'espacement entre les composants.

Ceci améliore :

- Uniformité
- Résilience (pérennité)
- Connexion entre les humains et le code
- Souplesse

Système de conception du GC



Base de produit (cadre d'utilitaires) consomme les jetons de conception et génère un cadre de style

Les **composants** sont des éléments réutilisables contenant des jetons qui peuvent être utilisés pour élaborer des interfaces

Le **système de conception** fournit aux équipes tous les conseils, les éléments de conception et les pièces précodées dont elles ont besoin pour construire une expérience utilisateur uniforme

Prochaine étape : Essais

Recherche sur le PMV

Petites équipes
en dehors
d'Adobe
Experience
Manager (AEM)

- Comparez la convivialité avec la Boîte à outils de l'expérience Web (BOEW) et le système de conception Canada.ca
 - Améliorons-nous quoi que ce soit?
 - Résolvons-nous les besoins des utilisateurs cernés?
 - Quelle est la faisabilité de la mise en œuvre?
- Orientation sur les composants
 - Trop, pas assez?
 - Claire et complète?

Tâches

Tâches à tester
avec les
développeurs et
les concepteurs

Tous

- Est-il facile pour les utilisateurs de trouver, comprendre et utiliser le site des documents?
- Comprendent-ils ce qui peut être personnalisé?

Développeurs

- Installer le paquet avec npm

Concepteurs

- Comprendre les éléments de formulaire
- Créer un formulaire de base

Équipe de produits du système de conception du GC

[Slack de conception du GC](#) • dto.btn@tbs-sct.gc.ca