

Les propriétés de composant sont des aspects modifiables d'un composant. Vous pouvez définir quelles parties d'un composant peuvent être modifiées par les autres utilisateurs en les liant à des propriétés de design spécifiques. Cela vous permet de contrôler des aspects tels que :

- Cacher ou afficher des calques
- Permuter ou non une instance
- Modifier certaines chaînes de texte

Ces définitions sont consolidées par un ensemble unique de contrôles dans la barre latérale de droite. Toute personne modifiant cette instance peut utiliser ces contrôles pour comprendre ce qui peut être modifié et s'atteler à la tâche.

Cela évite d'avoir à sélectionner et écraser les calques individuels, réduit le temps de documentation et améliore la précision d'utilisation des composants et des design systems.

[Découvrez le fichier d'initiation aux propriétés de composant via la Communauté Figma →](#)

Les propriétés de composant aident également à réduire la taille de votre design system ou de votre fichier de bibliothèque grâce à l'utilisation de propriétés booléennes. [En savoir plus ↓](#)

Tout comme avec les composants normaux, vous pouvez associer des descriptions et des liens pour agrémenter la documentation. Cela permet aux personnes utilisant vos composants d'avoir les bonnes informations et le bon contexte les concernant.

## Types de propriétés

Les propriétés de composant sont liées à différentes propriétés de design. Vous pouvez créer des propriétés de composant pour tout composant principal ou ensemble de composants et les appliquer à des calques imbriqués du composant ou de la variante.

Il existe plusieurs types de propriétés de composant :

- [Propriétés booléennes ↓](#)
- [Propriétés de permutation d'instances ↓](#)

- [Propriétés de texte ↓](#)
- [Propriétés de la variante ↓](#)

## Propriété booléenne

Remarque : les propriétés booléennes ne sont actuellement disponibles que pour la visibilité des calques.

Utilisez les propriétés booléennes pour définir des valeurs true/false. Cela permet aux utilisateurs d'activer et de désactiver un attribut.

Par exemple, si un design system contient des boutons avec et sans icône, plutôt que de créer des variantes pour chaque état, appliquez une propriété booléenne à la visibilité du calque de l'icône.

C'est également une bonne manière de réduire la taille des design systems et des bibliothèques de fichiers.

Attention : avant d'appliquer des propriétés booléennes aux composants, il est important de prendre en compte tout besoin de [composants interactifs](#) ou de [connexions de prototypage](#). Les « nouilles » de prototypage doivent se connecter entre deux objets. Si vous connectez deux composants et que vous les consolidez à l'aide de propriétés booléennes, les connexions de prototypage seront perdues.

Par exemple, une propriété booléenne est appliquée à une icône dans un composant bouton afin de représenter deux états : un bouton avec une icône et un sans. Il est impossible de définir une interaction entre les deux, car vous n'utilisez qu'un seul composant. Les interactions de prototype doivent se produire entre deux objets individuels.

Si une propriété booléenne est créée pour la visibilité d'un calque et que la valeur est définie sur false, toute propriété de composant associée sera invisible. Une fois que vous réglez la valeur sur true, la propriété associée devient visible.

## Propriété de permutation d'instances

Une propriété de permutation d'instances indique quelles instances d'un composant principal ou d'une variante peuvent être permutées. Lors de la création d'une propriété de permutation d'instances, vous pouvez définir la valeur d'instance par défaut dans toutes les [bibliothèques activées](#) et locales.

Pour toute personne utilisant cette propriété, un menu de permutation d'instances est accessible pour permuter les ressources imbriquées entre eux dans un même sous-ensemble de composants, ou contre une ressource de n'importe quelle bibliothèque disponible.

Les instances imbriquées avec cette propriété de composant peuvent être permutées dans le calque de premier niveau de son composant parent. Vous ne pourrez pas la permuter contre une autre instance si vous sélectionnez l'instance imbriquée.

[Définissez des instances préférées afin que les autres utilisateurs sachent quelles instances permuter](#) ↓

## Propriété de texte

Utilisez des propriétés de texte pour indiquer quels calques de texte doivent être modifiés.

Que vous gériez la valeur par défaut de la propriété de texte ou que vous utilisiez une instance qui la contient, la chaîne de texte peut être modifiée à partir de la barre latérale de droite ou depuis le canevas. La valeur se met à jour dans les deux cas.

Les calques dépourvus de cette propriété peuvent quand même être mis à jour.

Remarque : les propriétés de composant de texte ne prennent pas encore en charge les textes riches — tels que les styles de liste, les exposants et d'autres [paramètres de type](#). Vous pouvez quand même appliquer ces paramètres au calque de texte,

mais leurs formats ne seront pas reproduits dans le panneau des propriétés de composant dans la barre latérale de droite.

Si le calque de texte contient un [style de liste numérotée ou à puces](#), modifier la chaîne de texte dans la barre latérale de droite supprimera le style de liste. Afin de le conserver, mettez plutôt à jour la chaîne depuis le canevas.

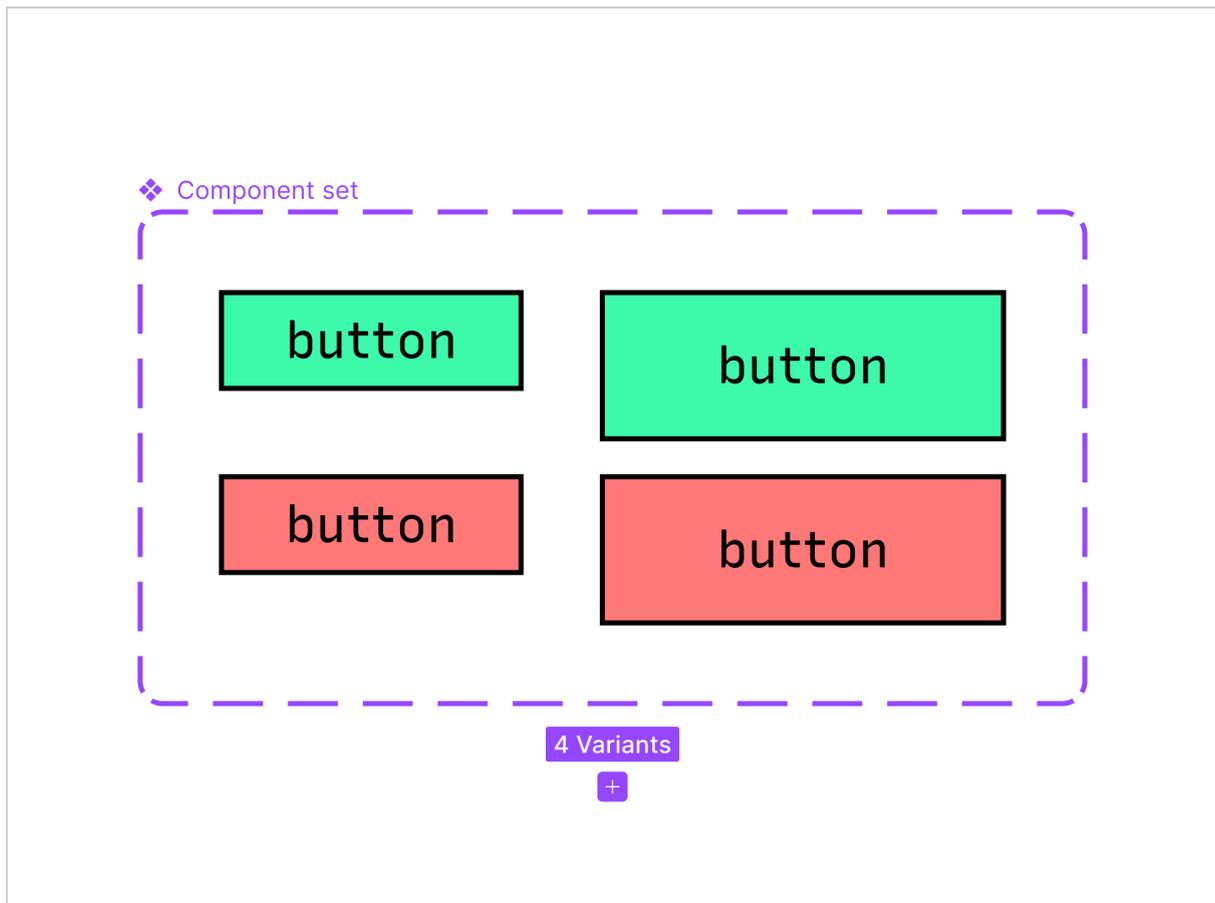
## Propriété de variante

Une variante est un composant individuel dans un ensemble de composants.

Une propriété de variante vous permet de définir des attributs pour vos variantes, tels que l'état, la couleur et la taille.

Un ensemble de composants contient plusieurs variantes et ces dernières sont dotées d'attributs qui peuvent être définis par des propriétés de variante. Les propriétés de variante sont réservées aux variantes et aux ensembles de composants et ne peuvent pas être créées ou appliquées aux composants principaux. [Découvrez comment créer des variantes et des ensembles de composants](#) →

Dans l'exemple ci-dessous se trouve un ensemble de composants bouton avec quatre variantes. Il contient deux propriétés de variante, la taille et la couleur. Les valeurs de taille sont petite et grande, tandis que les valeurs de couleur sont verte et rouge.



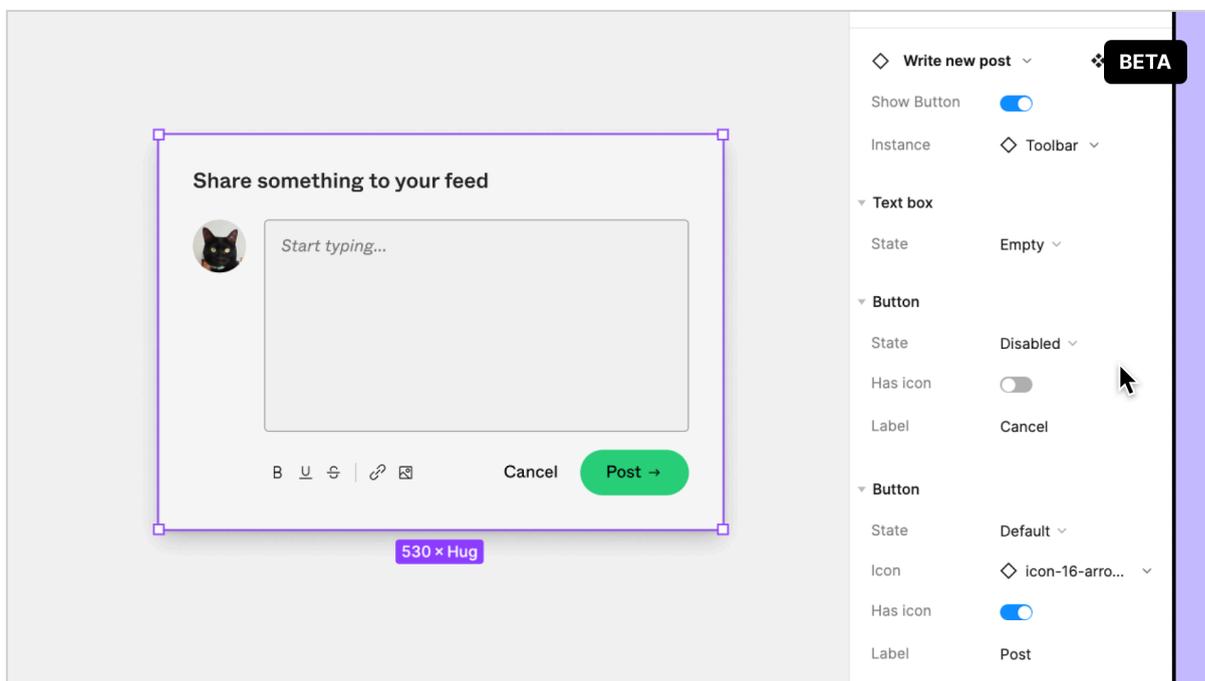
Remarque : nous vous conseillons d'examiner votre design system existant avant l'adoption de propriétés de composant. De cette manière, vous pouvez choisir quels aspects vous pouvez répercuter en tant que propriétés de composant et quels aspects seront des variantes.

## Exposer les instances imbriquées

Exposez des instances imbriquées spécifiques pour révéler leurs propriétés de composant ainsi que celles de l'instance de premier niveau. Cela permet aux utilisateurs de design system de découvrir les instances imbriquées et leurs propriétés de composant sans parcourir en profondeur les calques pour les trouver.

*Par exemple, vous créez un composant de carte de réseaux sociaux avec plusieurs composants imbriqués : un avatar, un nom, des boutons, etc. Vous pouvez modifier l'icône du bouton par défaut pour une icône « envoyer » et modifier le texte. Vous pouvez également modifier l'état du bouton d'abandon pour le passer à l'état désactivé.*

*Au lieu de parcourir en profondeur chaque calque pour trouver les contrôles de leur propriété de composant, exposer les instances imbriquées vous permet de simplement cliquer sur le composant premier niveau, ici, la carte de réseaux sociaux, et de modifier votre instance en un seul endroit.*



Lorsque vous sélectionnez une instance de premier niveau avec des instances imbriquées exposées, une liste des propriétés de composant pour les instances de premier niveau et imbriquées apparaît dans la barre latérale de droite.

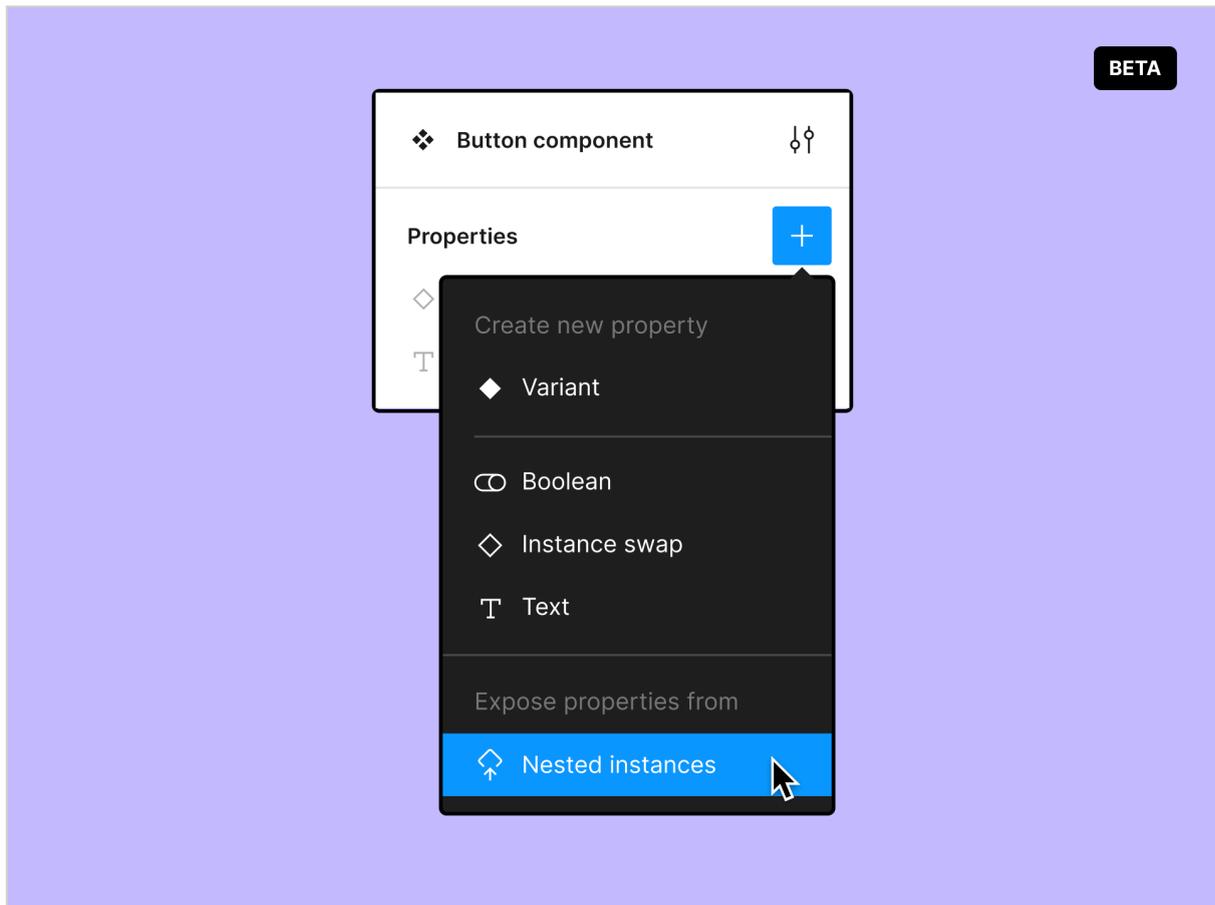
Lorsque vous passez votre souris sur une ligne de propriétés, une légère surbrillance violette apparaît autour de l'objet correspondant sur le canevas. Ainsi, vous savez ce que vous êtes en train de modifier.

Remarque : si vous avez une instance imbriquée exposée avec le paramètre de visibilité défini sur caché dans une propriété booléenne, toutes les propriétés de composant associées à l'instance sont aussi cachées.

## Choisir les instances imbriquées à exposer

Lorsqu'un composant principal ou un ensemble de composants est sélectionné, cliquez sur dans la section Propriétés dans la barre latérale de droite et sélectionnez

Instances imbriquées sous Exposer les propriétés de.

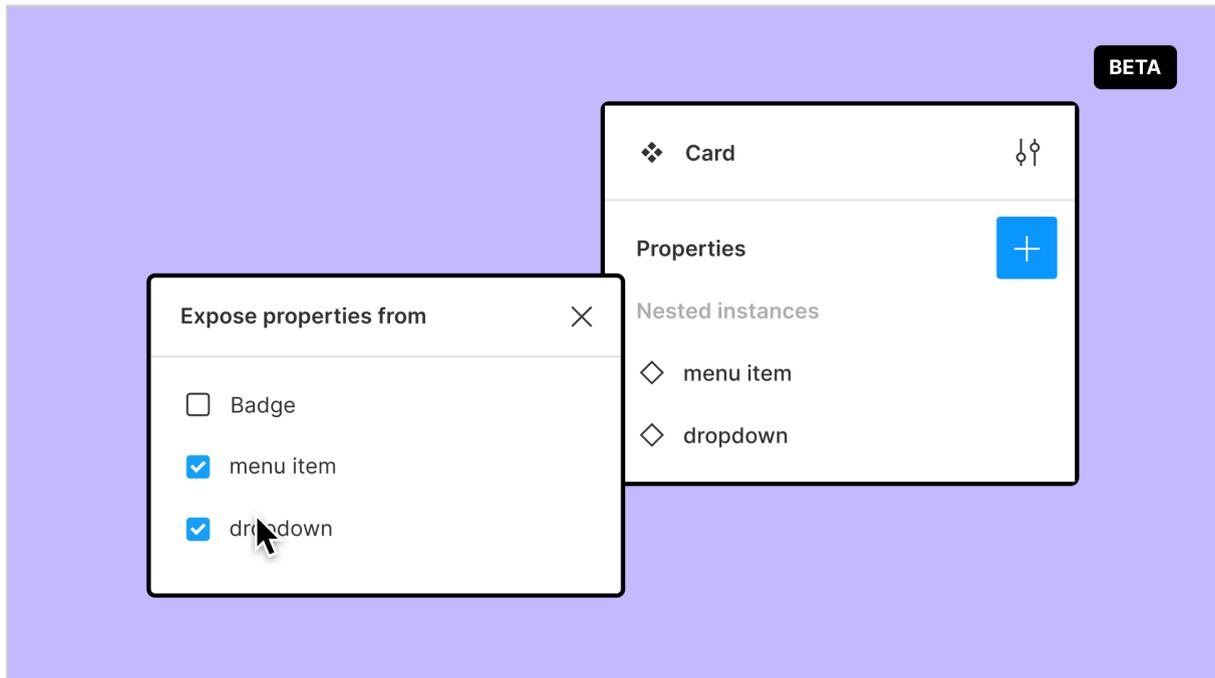


Remarque : l'option d'exposition des instances imbriquées n'est disponible que si un composant principal :

- possède déjà une instance imbriquée exposée, ou ;
- contient des instances imbriquées avec des propriétés de composant appliquées.

Si l'option Instances imbriquée n'est pas visible, essayez d'abord d'appliquer une propriété de composant à un composant imbriqué.

Depuis la fenêtre modale Exposer les instances imbriquées, vérifiez les cases des instances que vous souhaitez exposer au premier niveau.



Une liste des instances imbriquées sélectionnées apparaît dans la barre latérale de droite.

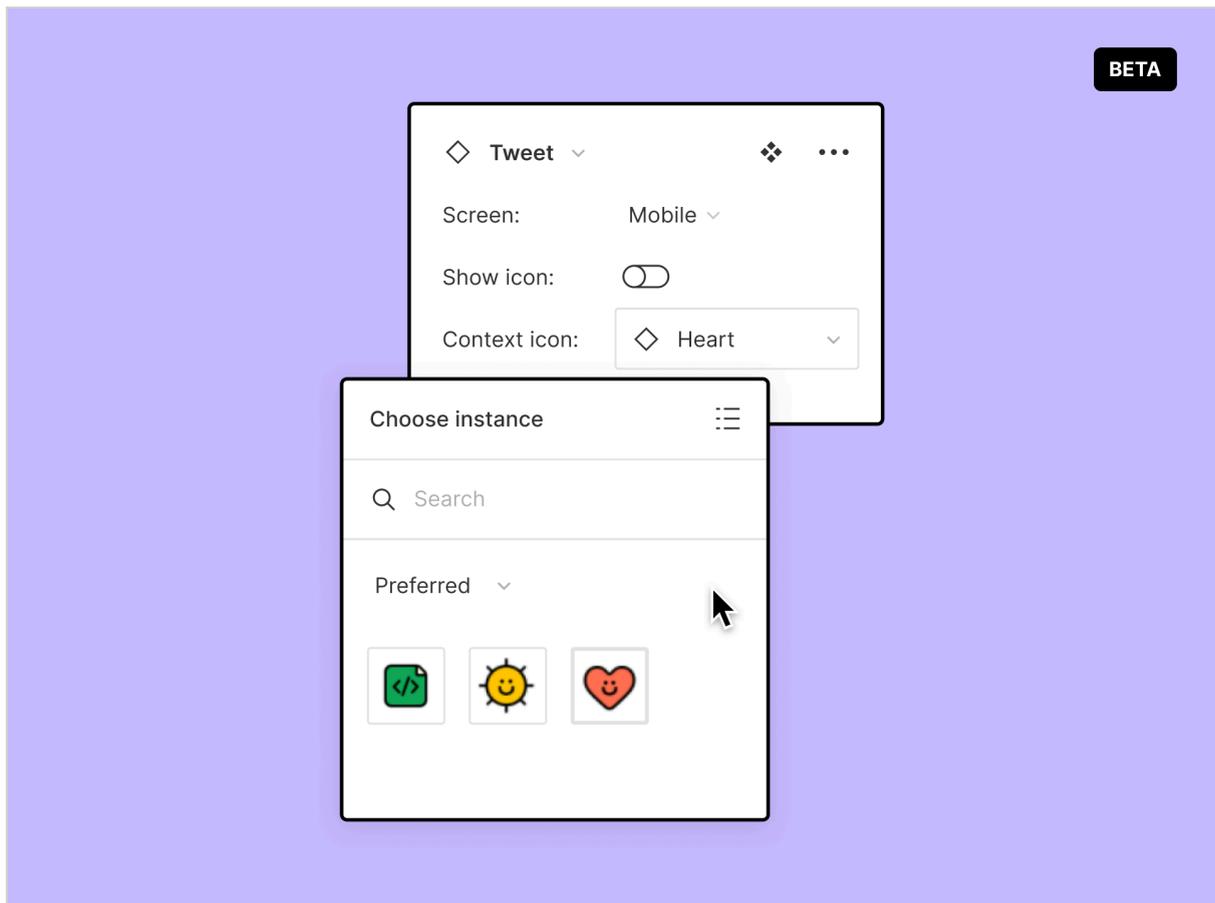
Pour supprimer des instances imbriquées de la liste des instances exposées, survolez le nom de l'instance dans la liste et cliquez sur .

## Valeurs préférées

Les valeurs préférées vous permettent de créer un ensemble intégré de composants dans lequel choisir les instances à permuter (via la propriété de permutation d'instances). Elles réduisent le temps de réflexion en communiquant quels composants spécifiques peuvent remplacer un composant existant.

*Par exemple, vous créez un bouton d'icône avec une propriété de permutation d'instances pour indiquer que l'icône peut être permutée. Cependant, votre bibliothèque d'icônes contient plus de 100 icônes, mais seulement 8 d'entre elles doivent être utilisées à cet effet.*

*Afin de faciliter la recherche de ces 8 icônes et pour savoir laquelle peut être utilisée, vous les ajoutez en tant que valeurs préférées. Dorénavant, lorsque des concepteurs voudront permuter l'icône, ils auront une liste intégrée de ressources à utiliser.*



Lorsque vous utilisez un composant avec une valeur préférée, une liste de valeurs préférées apparaît par défaut lorsque vous accédez au menu de propriété de permutation d'instances.

Remarque : si vous ne voulez pas utiliser une valeur préférée, cliquez sur la liste déroulante ou utilisez la barre de recherche au-dessus de la liste afin de trouver d'autres composants disponibles.

## Ajouter et supprimer

- Pour ajouter des valeurs préférées à une instance, commencez par modifier sa propriété de permutation d'instances ou créez-en une nouvelle. Depuis les fenêtres modales respectives, cliquez sur dans la section Valeurs préférées et vérifiez les cases en regard des instances que vous voulez désigner comme valeurs préférées.
- Supprimez une valeur préférée à tout moment en cliquant sur en regard de la valeur dans la section Valeurs préférées de la fenêtre modale.

## Instances simplifiées

Simplifier une instance aide à réduire l'encombrement du panneau des calques et des propriétés en cachant les calques sans les propriétés de composant appliquées. Figma part du principe qu'un calque sans propriété de composant est un calque qui ne devrait pas être modifié, et qui donc, peut être caché.

Alt : simplifiez les instances pour réduire l'encombrement du panneau de calques (et de propriétés ?) en cachant les noms des calques dépourvus de propriétés de composant.

Remarque : les instances simplifiées cachent certains noms de calques, mais toute personne possédant des autorisations pour éditer le fichier peut quand même modifier les calques.

Pour simplifier une instance, sélectionnez un composant principal ou un ensemble de composants et cliquez sur dans la barre latérale de droite. Depuis la fenêtre contextuelle, vérifiez la case Simplifier toutes les instances.

Lorsque vous utilisez une instance d'un composant simplifié, les calques trop nombreux sont rangés dans la liste déroulante Voir tous les calques. Cliquez sur Voir tous les calques pour dérouler la liste et afficher tous les calques. Vous pouvez la réduire de nouveau en cliquant à l'extérieur du panneau des calques ou en sélectionnant un autre calque.

<https://help.figma.com/hc/fr/articles/5579474826519-D%C3%A9couvrir-les-propri%C3%A9t%C3%A9s-de-composant>

les min width

Dans Figma, les propriétés **Max Width** (largeur maximale) et **Min Width** (largeur minimale) sont utilisées pour contrôler la largeur des éléments de design de manière flexible, en particulier lorsqu'il s'agit de créer des composants ou des éléments responsives. Ces propriétés sont particulièrement utiles lorsque vous travaillez avec des designs qui doivent s'adapter à différentes tailles d'écran ou de conteneurs.

### 1. Min Width (Largeur Minimale)

La **Min Width** définit la largeur minimale qu'un élément peut avoir. Peu importe le contenu ou les contraintes de son parent, l'élément ne deviendra jamais plus petit que la valeur définie pour la largeur minimale.

### Exemple :

Imaginons que vous créez un bouton dans Figma. Vous voulez que ce bouton puisse s'étirer pour contenir du texte dynamique, mais qu'il ne devienne jamais plus petit que 100 pixels pour assurer sa lisibilité.

- Si vous définissez la **Min Width** à 100 pixels, même si le texte du bouton est court (comme "OK"), le bouton ne sera jamais plus petit que 100 pixels de large.

## 2. Max Width (Largeur Maximale)

La **Max Width** définit la largeur maximale qu'un élément peut avoir. Cela signifie que même si le contenu ou les contraintes extérieures voudraient agrandir l'élément, celui-ci ne dépassera pas la largeur maximale définie.

### Exemple :

Supposons que vous concevez une barre de recherche qui doit occuper tout l'espace disponible dans un conteneur, mais vous ne voulez pas qu'elle dépasse une certaine largeur pour éviter qu'elle ne devienne trop étendue sur les grands écrans.

- Si vous définissez la **Max Width** à 600 pixels, la barre de recherche ne s'étendra jamais au-delà de cette largeur, même si l'espace disponible dans le conteneur est plus grand.

## Combinaison de Min Width et Max Width :

Ces deux propriétés peuvent être combinées pour créer des éléments flexibles qui s'adaptent parfaitement aux différents scénarios de design.

### Exemple combiné :

Imaginez que vous concevez une carte d'information (comme une carte produit) dans une interface.

- Vous définissez la **Min Width** de la carte à 200 pixels pour s'assurer que la carte ne devienne jamais trop petite.
- Vous définissez également la **Max Width** à 400 pixels pour éviter que la carte ne devienne trop large et perde son esthétique sur les grands écrans.

Dans ce cas, la carte pourra s'étendre et se réduire entre 200 et 400 pixels en fonction de l'espace disponible et du contenu, garantissant ainsi un design harmonieux.

## Conclusion :

Les propriétés **Min Width** et **Max Width** dans Figma sont des outils essentiels pour contrôler la réactivité et la flexibilité des éléments de design. Elles permettent de s'assurer que vos éléments restent dans des dimensions acceptables, peu importe les conditions, tout en s'adaptant aux différentes tailles d'écran ou aux contraintes de conteneur.