

Les widgets & Layouts

IFNTI

1 Objectifs

- Comprendre ce qu'est un widget en Flutter.
- Faire la distinction entre `StatelessWidget` et `StatefulWidget`.
- Manipuler les widgets courants (`Text`, `Image`, `Button`, `Icon`).

2 Introduction aux Widgets

— Qu'est-ce qu'un widget ?

En flutter tout est widget. Un widget est un élément de base dans Flutter qui représente une partie de l'interface utilisateur, comme un bouton, une image, un texte ou un conteneur. Les widgets forment une hiérarchie basée sur la composition. Chaque widget s'imbrique dans son parent et peut recevoir un **contexte** de ce dernier. Les widgets sont utilisés pour construire l'arbre d'interface utilisateur d'une application Flutter.

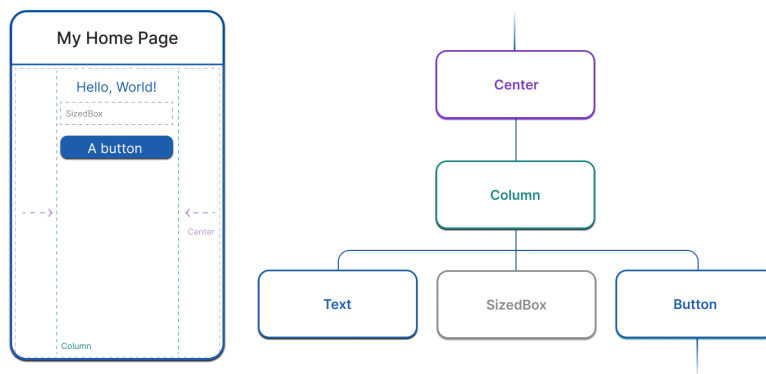


FIGURE 1 – Tiré de la documentation officielle de flutter.

— Qu'est-ce que l'état d'un widget ?

L'état d'un widget est un ensemble de données ou de variables internes qui influencent son apparence et son comportement. Un widget avec état peut changer dynamiquement en fonction des interactions ou des événements.

— Les différents type de widget :

- `StatelessWidget` :
- `StatefulWidget`

— Différence entre `StatelessWidget` et `StatefulWidget`

Un `StatelessWidget` est un widget dont l'apparence ne change jamais après sa création, car il n'a pas d'état modifiable. En revanche, un `StatefulWidget` est un widget qui peut avoir un état mutable et peut réagir aux interactions ou aux changements de données.

3 Manipulation des Widgets Courants

- Ajoutez un texte (**Text**) affichant "Bonjour, Flutter ICI c'est IFNTI!".
- Insérez une image (**Image**) à partir d'une URL ou des assets locaux.
- Ajoutez un bouton (**Button**) qui affiche un message dans la console lorsque l'utilisateur clique dessus.
- Insérez une icône (**Icon**) et modifiez sa taille et sa couleur.

4 TP 1

Créer une interface utilisateur permettant à un utilisateur de modifier la valeur d'un compteur quand il click sur un bouton.

5 Les layouts

Un Layout est un widget. Il l'arbre d'interface utilisateur construit à partir de plusieurs autres widget. C'est aussi un type particulier de widget qui définit comment disposer d'autres widgets (enfants) à l'écran, en termes de : Position, Taille, Alignement, Espacement.

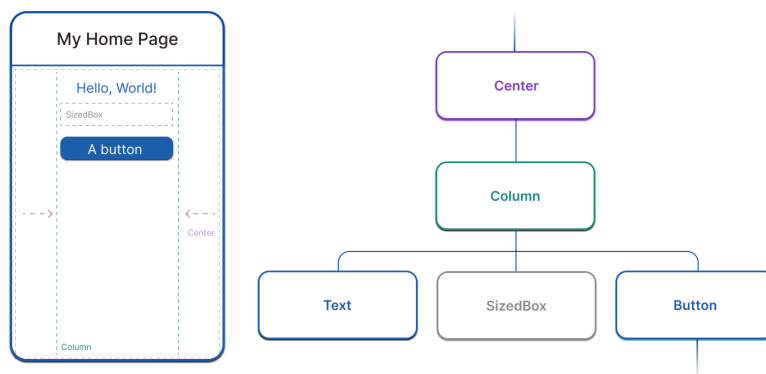


FIGURE 2 – Tiré de la documentation officielle de flutter.

Exemples de widgets Layout :

- Column : Dispose les enfants verticalement.
- Row : Dispose les enfants horizontalement.
- Stack : Superpose les enfants.
- Container : Permet de regrouper, styliser et ajuster la taille d'autres widgets.
- ListView : Dispose les enfants en défilement vertical ou horizontal.
- GridView : Crée une grille d'éléments.
- Flex : Fournit un contrôle avancé de disposition flexible.

6 TP 2

Manipulons les widgets avec la documentation de flutter.

Lien : <https://docs.flutter.dev/get-started/fundamentals/layout>