Створюючи віджет, одночасно з ним створюються три об’єкта дерева для нього:

1. Widget — Describes the configuration for an [Element] (Widget є описом деякого Element).
2. Element — An instantiation of a [Widget] at a particular location in the tree (Element це певне представлення віджету в певному місці дерева).
3. RenderObject — це об’єкт дерева візуалізації, те що бачимо в кінцевому результаті

Widget – є основним класом який використовується у Флаттер, в той же час він є описом конфігурації певного Element. Самі віджети не мають змінного стану - якщо потрібно змінити стан віджета, для цього використовується StatefulWidget, який створює об’єкт State кожного разу, коли він співставляється з Element і включається в дерево

*//Describes the configuration for an [Element](https://api.flutter.dev/flutter/widgets/Element-class.html).*

*Widgets are the central class hierarchy in the Flutter framework. A widget is an immutable description of part of a user interface. Widgets can be inflated into elements, which manage he underlying render tree.*

https://api.flutter.dev/flutter/widgets/Widget-class.html

Як було вище вказано, клас Element відповідає за розташування віджету

При кожному використанні віджету створюється свій Element який до нього співставляється та вказує розташування віджету у дереві. Тобто, другими словами, кожного разу при використанні віджету створюється не новий віджет а Element який представляє використання віджета для налаштування певного розташування в дереві – за рахунок цього (що створюється шлях до віджету а не сам віджет ) покращується продуктивність (Performance) так як не витрачається ресурс на створення віджету. Для позиціонування віджетів використовується метод BuildContext який рівнозначний Element, він називається по іншому для того щоб не було плутанини в дереві Element

Кожен Element має свій життєвий цикл:

**-Створення елементу**

**-Вбудовування та активація**

**-Оновлення та перестворення**

-**Деактивація**

**-Перевикористання**

*//An instantiation of a [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) at a particular location in the tree.*

*Widgets describe how to configure a subtree but the same widget can be used to configure multiple subtrees simultaneously because widgets are immutable. An*[*Element*](https://api.flutter.dev/flutter/widgets/Element-class.html)*represents the use of a widget to configure a specific location in the tree. Over time, the widget associated with a given element can change, for example, if the parent widget rebuilds and creates a new widget for this location.*

https://api.flutter.dev/flutter/widgets/Element-class.html

Щодо **RenderObject** то це об’єкт дерева візуалізації. У нього є батьківський об’єкт, а також поле з даними, які батьківський об’єкт використовує для зберігання специфічної інформації, яка стосується самого цього об’єкту, наприклад, його позиції. Даний об’єкт відповідає за реалізацію базових протоколів відмальовування та розташування

Всі віджети які відображаються для користувача мають RenderObject, а створюється він на етапі циклу **Вбудовування** Element - викликається метод **mount** який окрім додавання новоствореного елемента до дерева в заданий слот у даному батьківському елементі також викликикає **attachRenderObject**, за допомогою якого прикріпляються будь-які пов’язані об’єкти візуалізації до дерева візуалізації.