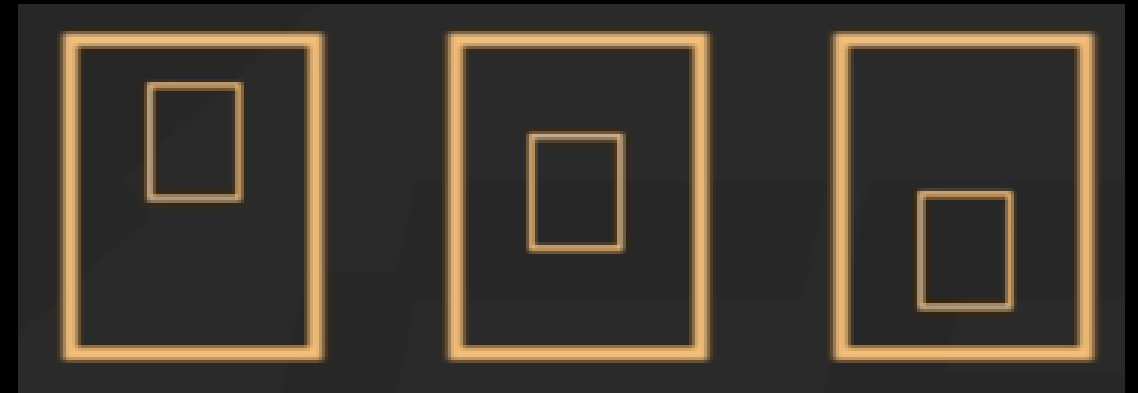


Introducción al Desarrollo de Aplicaciones en Android

- Ing. Skrauba Axel

Sistema de Posicionamiento

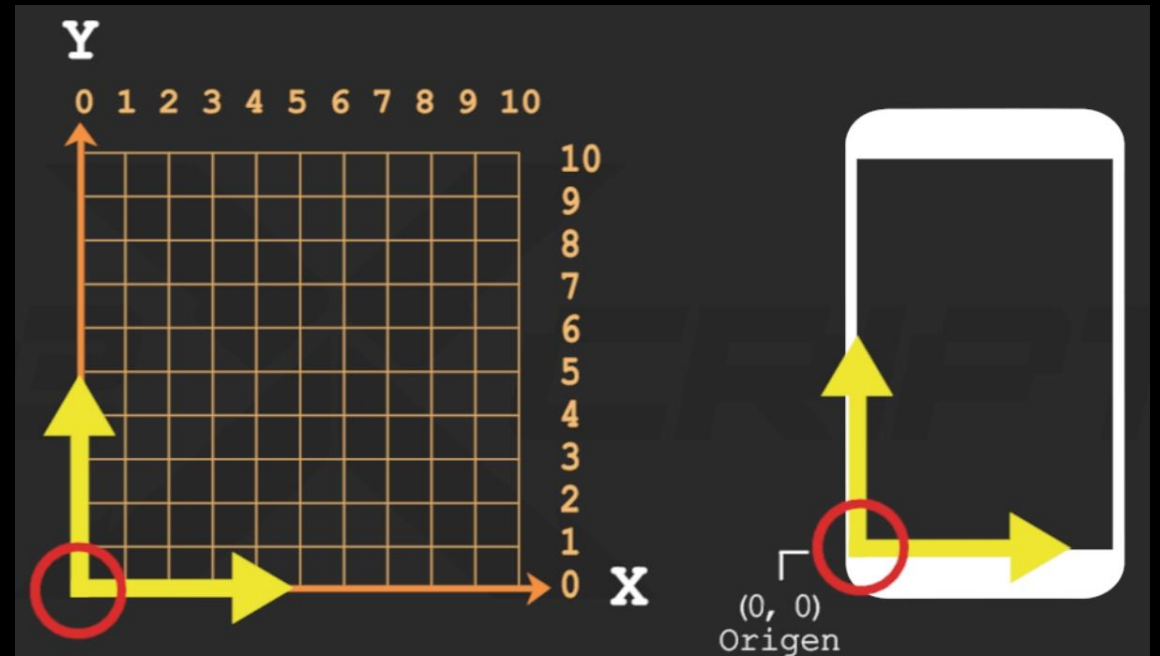
El sistema de posicionamiento en *Kivy* se basa en una combinación de coordenadas y proporciones, lo que permite que la interfaz se adapte a diferentes tamaños de pantalla y orientaciones.



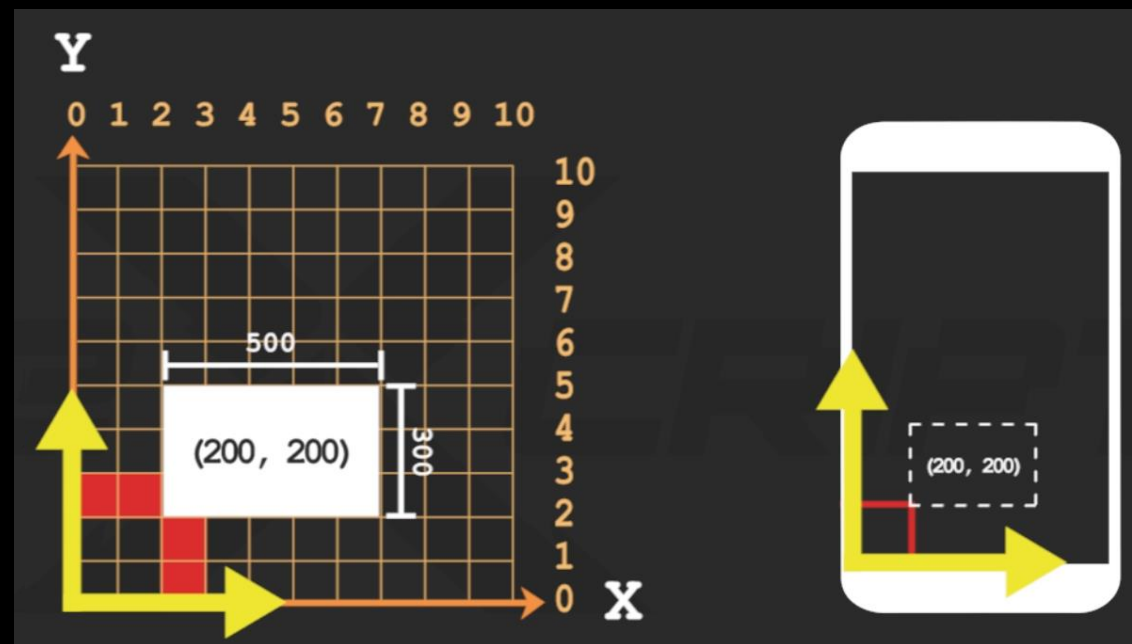
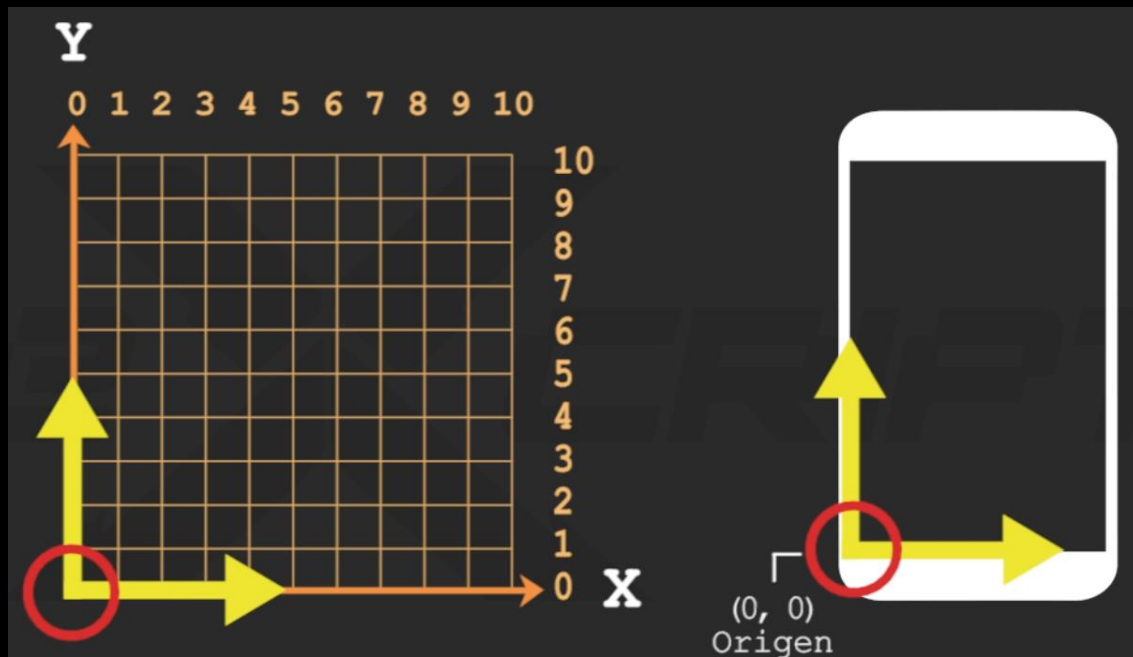
Sistema de Posicionamiento

Coordenadas

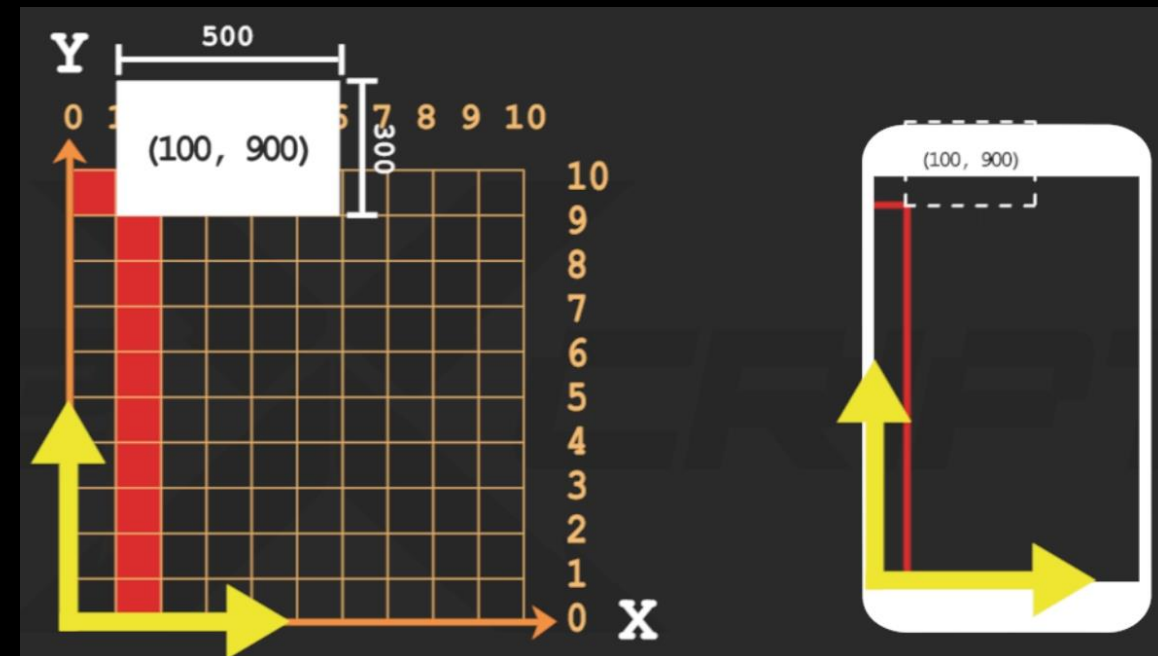
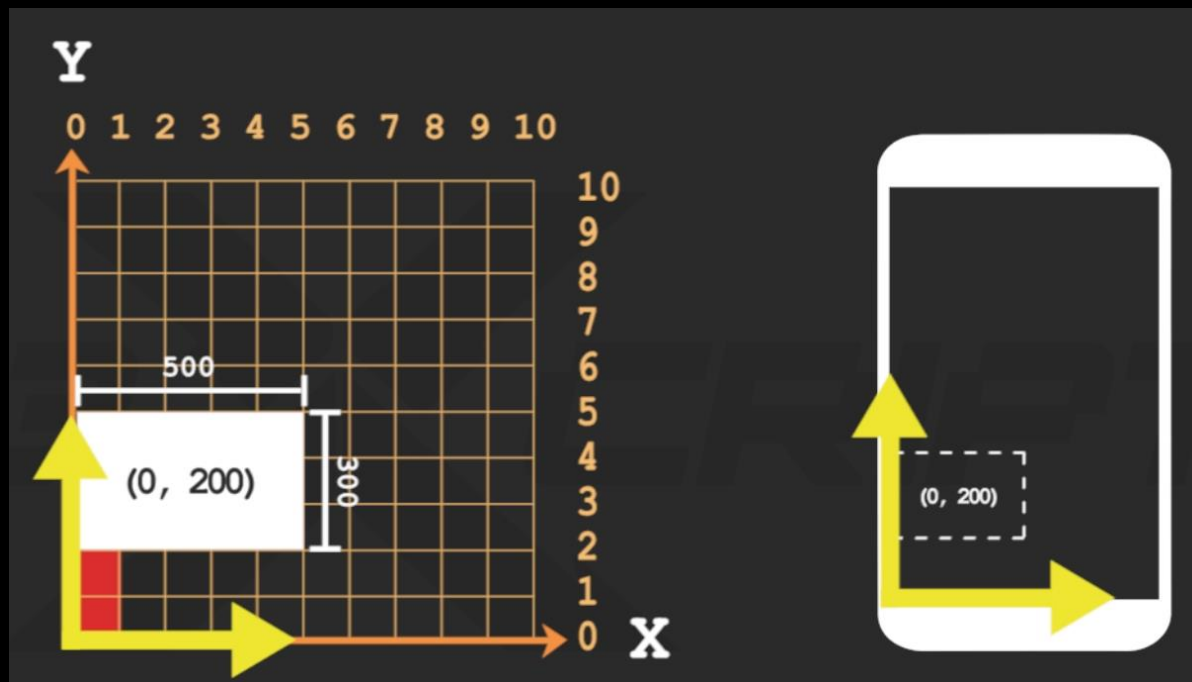
Kivy utiliza coordenadas (**x**, **y**) para posicionar widgets en la pantalla. La esquina inferior izquierda se considera el punto de origen (**0, 0**), y las coordenadas se incrementan hacia arriba y hacia la derecha. Los valores son absolutos y pueden especificarse en píxeles (**px**) o en densidad independiente de píxeles (**dp**).



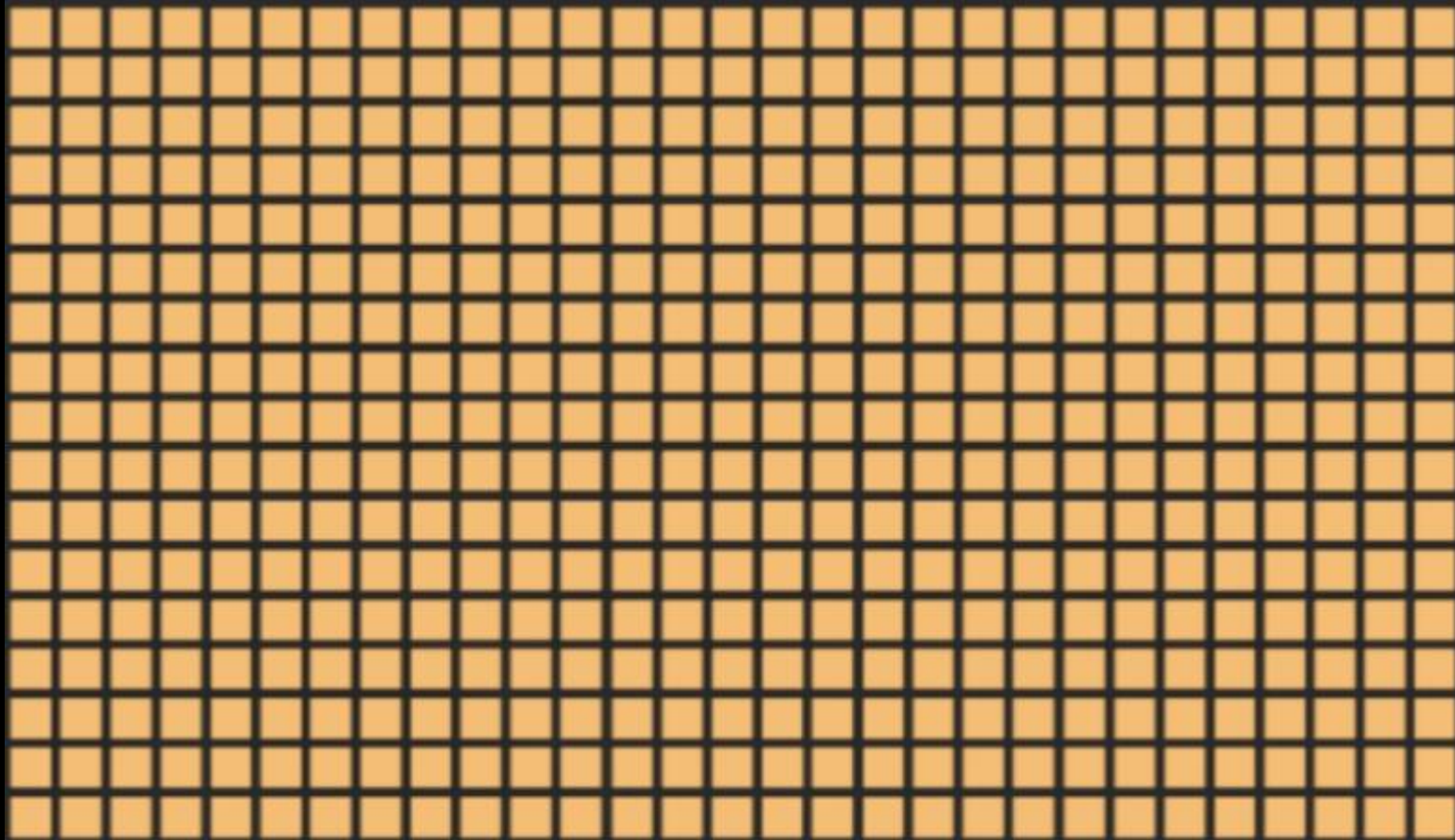
Sistema de Posicionamiento



Sistema de Posicionamiento



Sistema de Medida



Sistema de Medida

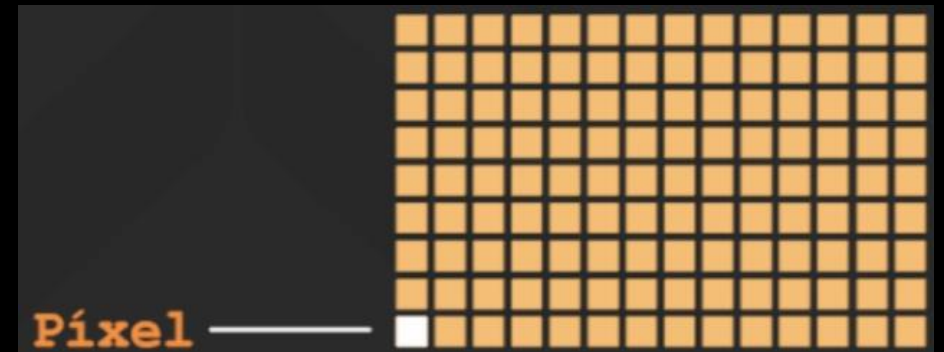
¿Qué es?

El sistema de medida en *Kivy* se refiere a cómo se definen y utilizan las unidades de longitud y otros valores en las propiedades de los *widgets* y la interfaz en general.

Kivy utiliza principalmente dos tipos de unidades de medida: **píxeles (px)** y **densidad independiente de píxeles (dp)**.

El píxel es una unidad de medida universal, utilizada por los fabricantes de hardware y lenguajes de programación.

El tamaño físico del píxel varía, cuanto menor sea este, mayor será la resolución.



Sistema de Medida

Unidades de Medida

- ***px (píxeles)***: Unidad de medida absoluta en píxeles.
- ***dp (density-independent pixels)***: Unidad de medida que ajusta el tamaño para diferentes densidades de pantalla.
- ***% (porcentaje)***: Utilizado en `size_hint` y `pos_hint` para definir proporciones relativas.

px	Píxeles
m	Pulgadas
mm	Milímetros
pt	Puntos
dp	Píxel Independiente de Densidad
sp	Escala Independiente de Píxeles

Sistema de Medida

Densidad Independiente de Píxeles

El sistema de unidades **dp** se utiliza para lograr que las dimensiones y posiciones sean consistentes en diferentes densidades de pantalla. *Kivy* ajusta automáticamente el valor de **dp** según la densidad de la pantalla, lo que ayuda a que la interfaz se vea y funcione de manera similar en diferentes dispositivos.

En términos prácticos, **dp** permite que los elementos de la interfaz se vean aproximadamente del mismo tamaño en diferentes pantallas con densidades variadas. Por ejemplo, un *widget* con un tamaño de **dp(100)** se verá más o menos del mismo tamaño en una pantalla de alta densidad y en una pantalla de baja densidad.

Sistema de Medida

Porcentual

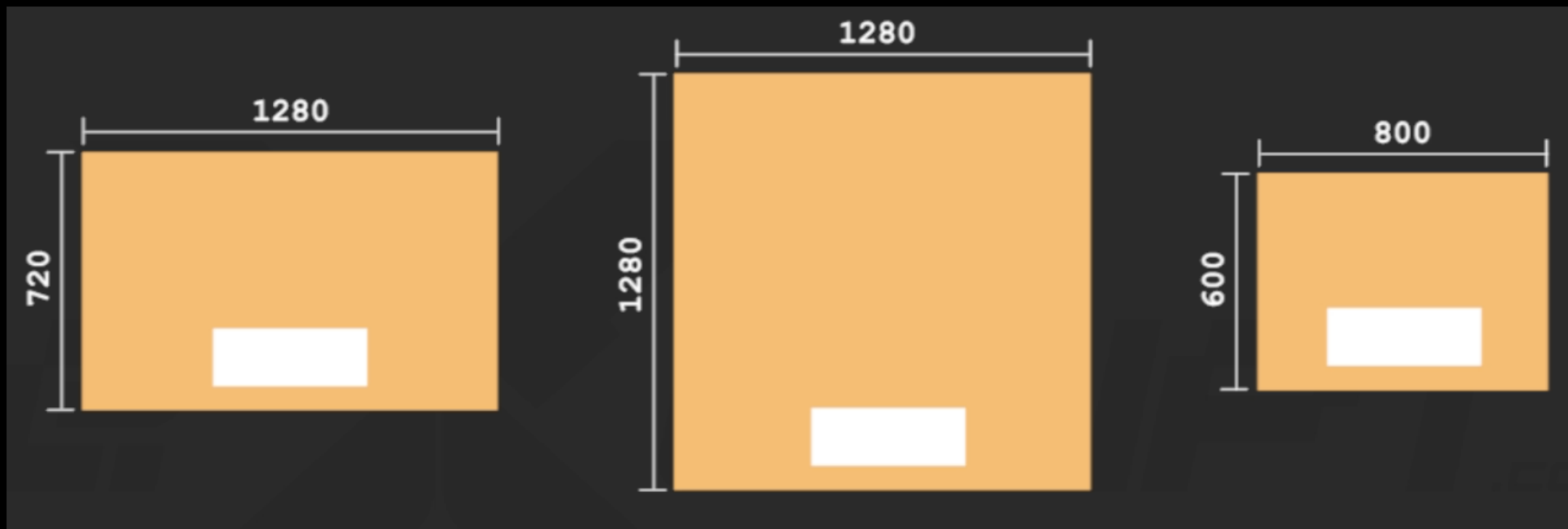
Kivy también permite usar porcentajes en ciertas propiedades, especialmente en las propiedades **size_hint** y **pos_hint**.

Los porcentajes se utilizan para indicar proporciones relativas en relación con el contenedor o el padre. Por ejemplo, **size_hint: 0.5, 0.5** significa que el widget ocupa la mitad del ancho y la mitad de la altura del contenedor.

size_hint: Define la relación entre el tamaño del widget y el tamaño del contenedor o padre. Puede ser un valor flotante o una tupla (ancho, alto).

pos_hint: Define la relación entre la posición del widget y la posición del contenedor o padre. Puede ser un diccionario que contiene valores para 'x' (horizontal) y 'y' (vertical).

Sistema de Medida



KVLANG

Kvlang (Kivy Language) es un lenguaje de marcado específico diseñado para construir interfaces de usuario en aplicaciones desarrolladas con el *framework Kivy*.

Proporciona una forma declarativa de describir la estructura y el diseño de la interfaz gráfica, separando la lógica de la interfaz de manera efectiva

```
root = MyRootWidget()
box = BoxLayout()
box.add_widget(Button())
box.add_widget(Button())
root.add_widget(box)
```

```
MyRootWidget:
    BoxLayout:
        Button:
        Button:
```

KVLANG

A medida que su aplicación se vuelve más compleja, es común que la construcción de árboles de widgets y la declaración explícita de enlaces se vuelvan detalladas y difíciles de mantener. El lenguaje KV es un intento de superar estas deficiencias.

<https://kivy.org/doc/stable/guide/lang.html>

Permite prototipos muy rápidos y cambios ágiles en su interfaz de usuario. También facilita la separación de la lógica de su aplicación y su interfaz de usuario.

```
<MyWidget>:
    label_widget: label_widget
    Button:
        text: 'Add Button'
        on_press: root.add_widget(label_widget)
    Button:
        text: 'Remove Button'
        on_press: root.remove_widget(label_widget)
    Label:
        id: label_widget
        text: 'widget'
```

KVLANG: Beneficios

Facilita el Diseño: Permite a los diseñadores crear interfaces visuales sin necesidad de conocer detalles de programación, lo que acelera el proceso de diseño y desarrollo.

Mantenibilidad: La separación entre lógica e interfaz facilita la modificación y actualización de la interfaz sin afectar la funcionalidad subyacente.

Rápido Prototipado: Permite construir prototipos rápidamente, lo que ayuda a visualizar la interfaz y ajustarla según las necesidades del usuario.

Adaptabilidad: Las propiedades `size_hint` y `pos_hint` permiten crear interfaces que se ajustan a diferentes tamaños y orientaciones de pantalla.

Menos Código: Reduce la cantidad de código Python necesario para definir la interfaz, lo que simplifica el desarrollo y mejora la legibilidad.

Colaboración Eficaz: Los diseñadores y desarrolladores pueden trabajar en paralelo, ya que proporciona un punto de integración claro entre la lógica y la interfaz.

Diseño Declarativo: La naturaleza declarativa de *Kvlang* facilita la identificación y resolución de problemas en la interfaz.

KVLANG: uso

Por convención de nombres:

Kivy busca un archivo **Kv** con el mismo nombre que su clase **App** en minúsculas, menos "App" si termina en "App", por ejemplo:

MyApp -> my.kv

Las reglas usan sangría para la delimitación, como Python. La sangría debe ser de cuatro espacios por nivel, como recomienda la guía de estilo de Python.

Hay tres palabras clave específicas del lenguaje Kv:

app: siempre se refiere a la instancia de tu aplicación.

root: se refiere al widget/plantilla contenedor actual.

self: siempre hace referencia al widget actual.

Preguntas...

- Dudas
- Sugerencias
- Cuestiones
- Vamos al código...

