

Nhóm B



A A A

Text và Styling trong Flutter



TeamMember:
Văn Tiến Lộc
Nguyễn Trung Thành

Start →

1. TextStyle Properties.
2. TextOverflow và TextAlignement.
3. RichText và TextSpan.
4. Custom Fonts và Theme

Contents



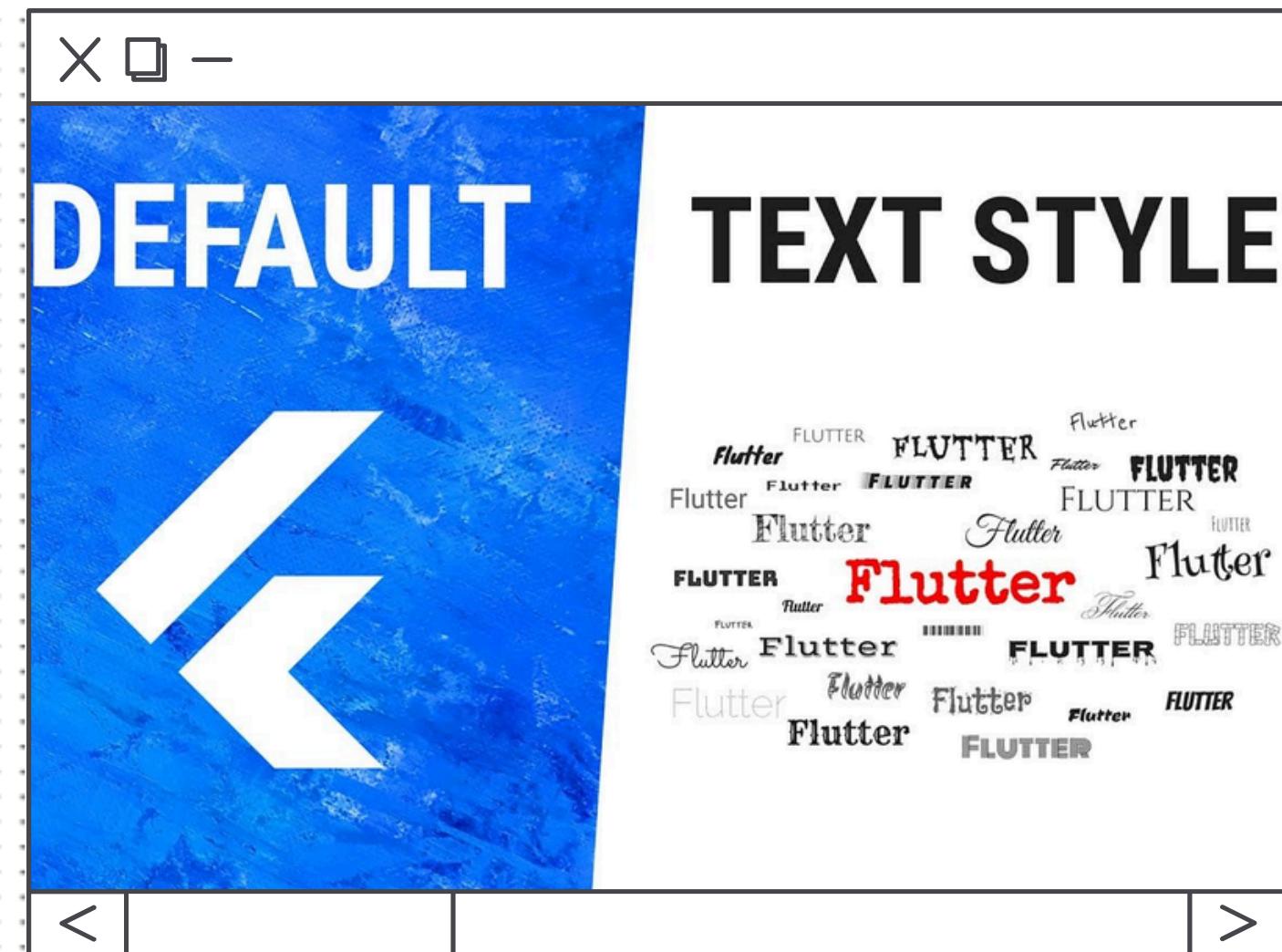
Let's go!



TextStyle Properties

Trong **Flutter**, lớp **TextStyle** dùng để định dạng giao diện (style) của chữ trong **Text widget**.

- `fontSize`
- `fontWeight`
- `Color`
- `fontStyle`



Next →

- Khái niệm: kiểu thuộc tích double?trong lớp TextStyle.
- Đơn vị: Giá trị được chỉ định bằng logical pixels.
- Mặc định: 14 logical pixels (hoặc kích thước được định nghĩa bởi Theme)

Luôn bị ảnh hưởng bởi Tỷ lệ chữ của hệ thống do người dùng thiết lập trong cài đặt trợ năng của thiết bị.

- Tính toán kích thước hiển thị: Kích thước chữ cuối cùng được hiển thị trên màn hình sẽ là:

Kích thước hiển thị=fontSize×textScaleFactor

FontSize



```
class Home extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
          children:   
            [Text(  
              'I am the Moon',  
              textAlign: TextAlign.left,  
            ), // Text  
            SizedBox(height: 8),  
            Text(  
              'I am the Earth',  
              textAlign: TextAlign.center,  
              style: TextStyle(fontSize: 20),  
            ), // Text  
            SizedBox(height: 8),  
            Text(  
              'I am the Sun',  
              textAlign: TextAlign.right,  
              style: TextStyle(fontSize: 30),  
            ), // Text  
          ],  
        ),  
      ),  
    );  
  }  
}
```

Next →

FontWeight



- Khái niệm: dùng để xác định độ đậm của chữ.
- Kiểu dữ liệu: là 1 enum/class.

- Cách sử dụng: thường sử dụng các hằng số được định nghĩa sẵn trong FontWeight class:
 - FontWeight.normal: Độ đậm chuẩn (400).
 - FontWeight.bold: Chữ đậm.
 - FontWeight.w100 đến FontWeight.w900: Cho phép chọn độ đậm cụ thể theo thang số từ 100 đến 900.

Next →

```
  // Row
  children: const <Widget>[
    Text('Am Normal :D ',
        style: TextStyle(fontWeight: FontWeight.normal)),
    Text('Am Bold >:D ',
        style: TextStyle(fontWeight: FontWeight.bold)),
  ], // <Widget>[]
, // Row
  spacing: 8.0,
  runSpacing: 4.0,
  children: const <Widget>[
    Text('100 ',
        style: TextStyle(fontWeight: FontWeight.w100)),
    Text('200 ',
        style: TextStyle(fontWeight: FontWeight.w200)),
    Text('300 ',
        style: TextStyle(fontWeight: FontWeight.w300)),
    Text('400 ',
        style: TextStyle(fontWeight: FontWeight.w400))]
```

Color



- Khái niệm: dùng để mang màu sắc tới các ký tự.
- Kiểu dữ liệu: color?.
- Giá trị: sử dụng các hằng số từ lớp color hoặc màu tùy chỉnh.

- Để sử dụng Color:
 - Các màu có sẵn.
 - Màu custom.
 - Màu theo Theme.

ooo

+

```
text: '[(Saturn)]',
style: TextStyle(fontSize: 23, fontWeight: FontWeight.bold, color: Colors.brown[400]),
), // TextSpan
TextSpan(
  text: '[(Uranus)]',
  style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold,
    color: Theme.of(context).colorScheme.primary), // TextStyle
), // TextSpan
TextSpan(
  text: '(Neptune)',
  style: TextStyle(fontSize: 21, fontWeight: FontWeight.bold, color: Color(0x800000FF),),
```

Next



FontStyle



- Khái niệm: xác định kiểu chữ là đứng (normal) hay nghiêng (italic).
- Kiểu dữ liệu: FontStyle?

- Giá trị: Nó là một enum với 2 lựa chọn.
 - fontStyle.normal: kiểu chữ mặc định đứng thẳng.
 - dontStyle.italic: chữ in nghiêng.

```
const Text(  
  'Hi am standing here :D',  
  style: TextStyle(  
    | | | | |   fontStyle: FontStyle.normal  
  ), // Text  
const Text(  
  'Help Am Falling!!!',  
  style: TextStyle(  
    | | | | |   fontStyle: FontStyle.italic  
  ), // Text
```

Next →

TextOverflow



- Khi nội dung văn bản quá dài so với không gian đã được cung cấp thì lúc đó sẽ xảy ra tình trạng Text Overflow.
- Thuộc tính được sử dụng để kiểm soát điều này là overflow trong TextStyle của widget Text. Nó nhận giá trị từ enum TextOverflow.

ooo

+

Mặt Trời là ngôi sao ở trung tâm Hệ Mặt Trời, cung cấp ánh sáng và năng lượng cho mọi hành tinh, trong đó có Trái Đất. Nhờ có Mặt Trời mà sự sống trên Trái Đất quay quanh Mặt Trời theo một quỹ đạo hình elip, đồng thời tự quay quanh trục của mình. Chính sự chuyển động này tạo ra ngày và đêm, cũng như Mặt Trăng là vệ tinh tự nhiên duy nhất của Trái Đất. Nó phản chiếu ánh sáng từ Mặt Trời, khiến bầu trời đêm trở nên sáng hơn. Khi Mặt Trời, Trái Đất và Mặt Tr...

Next →

- Các giá trị chính của textOverflow.

- TextOverflow.clip: Cắt bỏ phần văn bản bị tràn.
- TextOverflow.fade: Phần văn bản bị tràn sẽ mờ dần thành suốt ở cuối.

- TextOverflow.ellipsis: Thay thế phần văn bản bị tràn bằng (...).
- TextOverflow.visible: tiếp tục hiển thị bên ngoài giới hạn của widget.

Lưu ý: có thể gây ra lỗi tràn pixel nếu không được xử lý cẩn thận.

Next →

TextOverflow



```
ooo
```

```
text(
```

```
'Mặt Trời là ngôi sao ở trung tâm
```

```
maxLines: 1,
```

```
overflow: TextOverflow.clip,
```

```
), // Text
```

```
Text(
```

```
'Trái Đất quay quanh Mặt Trời theo
```

```
maxLines: 1,
```

```
overflow: TextOverflow.fade,
```

```
), // Text
```

```
Text(
```

```
'Mặt Trăng là vệ tinh tự nhiên duy
```

```
maxLines: 1,
```

```
overflow: TextOverflow.ellipsis,
```

```
), // Text
```

```
1
```



Text Alignment



- Xác định cách văn bản được căn chỉnh trong không gian chiều ngang của widget Text.
- Thuộc tính được sử dụng là textAlign, nhận giá trị từ enum TextAlign.

- TextAlign.left: || TextAlign.right: || TextAlign.center:
- TextAlign.justify: Dàn đều khoảng cách giữa các từ để làm cho lề trái và lề phải thẳng hàng (chủ yếu dùng cho các đoạn văn dài).
- TextAlign.start:Căn chỉnh theo hướng bắt đầu của ngôn ngữ (trái cho hầu hết các ngôn ngữ LTR như tiếng Việt/Anh, phải cho các ngôn ngữ RTL).
- TextAlign.end:Căn chỉnh theo hướng kết thúc của ngôn ngữ (phải cho LTR, trái cho RTL).

```
ooo
class MyText extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: const [
        Text(
          'I am the Moon',
          textAlign: TextAlign.left,
        ), // Text
        SizedBox(height: 8),
        Text(
          'I am the Earth',
          textAlign: TextAlign.center,
          style: TextStyle(fontSize: 20),
        ), // Text
        SizedBox(height: 8),
        Text(
          'I am the Sun',
          textAlign: TextAlign.right,
          style: TextStyle(fontSize: 30),
        ), // Text
      ],
    );
  }
}
```

Next →

RichText & TextSpan



- RichText là widget dùng để hiển thị văn bản có nhiều kiểu định dạng khác nhau trong cùng một đoạn.
- Tùy theo giới hạn bố cục, đoạn văn bản có thể xuống dòng hoặc nằm trên một dòng duy nhất.
- Văn bản trong RichText phải được chỉ định kiểu (style) một cách rõ ràng.

```
ooo
child: RichText(
  text: TextSpan(
    children: <TextSpan>[
      TextSpan(
        text: '(Mars)',
        style: TextStyle(fontSize: 19, fontWeight: FontWeight.bold, color: Colors.red),
      ), // TextSpan
      TextSpan(
        text: '(Mercury)',
        style: TextStyle(fontSize: 10, fontWeight: FontWeight.bold, color: Colors.blueGrey),
      ), // TextSpan
    ],
  ),
),
```

Next →

RichText & TextSpan



ooo

```
children: [
    TextSpan(
        text: 'Xin chào ',
        style: TextStyle(color: Colors.pink,
            fontSize: 18
        ), // TextStyle
    ), // TextSpan
    TextSpan(
        text: 'Flutter',
        style: TextStyle(color: Colors.blue,
            fontSize: 18,
            fontWeight: FontWeight.bold
        ), // TextStyle
    ), // TextSpan
    TextSpan(
        text: ' ❤️',
        style: TextStyle(color: Colors.red,
            fontSize: 18
        ), // TextStyle
    ), // TextSpan
],
```



- Widget này cần nằm trong cây con của SelectionArea hoặc SelectableRegion, và phải gán SelectionRegistrar .
- Cần cấu hình thêm selectionColor để hiển thị màu đánh dấu khi văn bản được chọn.

Next →

TextSpan



TextSpan là một đối tượng bất biến đại diện cho một đoạn văn bản có định dạng. Một TextSpan có thể chứa:

- Chỉ một chuỗi text duy nhất.
- Danh sách children là các TextSpan con, mỗi phần lại có style riêng.

- Nếu một TextSpan vừa có text vừa có children, thì phần text sẽ được xem như một TextSpan không có style.
- Khi text bị để null, TextSpan đóng vai trò như một node rỗng trong cây văn bản.

- Để vẽ TextSpan lên Canvas, sử dụng TextPainter.
- Để hiển thị trong widget, sử dụng RichText.
- Nếu chỉ dùng một kiểu chữ duy nhất, hãy dùng Text thay vì RichText.

Next →

Custom Fonts Q

```
ooo
flutter:
  uses-material-design: true
  fonts:
    - family: Roboto # Tên font bạn sẽ dùng trong TextStyle
      fonts:
        - asset: fonts/Roboto-VariableFont_wdth,wght.ttf
        - asset: fonts/Roboto-Italic-VariableFont_wdth,wght.ttf
          style: italic # Khai báo để dùng với FontStyle.italic
```

```
ooo
  fonts
    - Roboto-Italic-VariableFont_wdth,wght.ttf
    - Roboto-VariableFont_wdth,wght.ttf
  > ios
```

1. Chọn font
2. Nhập file font vào dự án.
3. Khai báo font trong pubspec.yaml
4. Sử dụng font trong widget

```
override
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Flutter Text Styling Demo',
    theme: ThemeData(
      fontFamily: 'Roboto',
      textTheme: const TextTheme(
        bodyLarge: TextStyle(fontSize: 18),
        bodyMedium: TextStyle(fontSize: 16),
      ), // TextTheme
      useMaterial3: true,
      colorScheme: ColorScheme.fromSeed(seedColor: Colors.teal),
    ), // ThemeData
    home: const TextStylingDemo(),
  ); // MaterialApp
```



Mục đích

- Dùng ThemeData và TextTheme để định nghĩa kiểu chữ, màu sắc chung, đảm bảo tính nhất quán trong toàn bộ ứng dụng.
- Dễ thay đổi style toàn cục hoặc override riêng cho vùng nhỏ nếu cần.

Cách thực hiện

- Trong MaterialApp, có thể đặt thuộc tính theme là một đối tượng ThemeData.
- ThemeData có các thuộc tính như:
 1. colorScheme: xác định bảng màu chính của ứng dụng.
 2. textTheme: xác định các kiểu chữ tiêu chuẩn (headline, body, title,...).

Next →

Theme và TextTheme

```
override  
Widget build(BuildContext context) {  
  return MaterialApp(  
    title: 'Flutter Text Styling Demo',  
    theme: ThemeData(  
      fontFamily: 'Roboto',  
      textTheme: const TextTheme(  
        bodyLarge: TextStyle(fontSize: 18),  
        bodyMedium: TextStyle(fontSize: 16),  
      ), // TextTheme  
      useMaterial3: true,  
      colorScheme: ColorScheme.fromSeed(seedColor: Colors.teal)  
    ), // ThemeData  
    home: const TextStylingDemo(),  
  ); // MaterialApp
```

Nếu muốn override theme trong phần nhỏ của ứng dụng, thì có thể làm theo 2 cách sau:

1. Tạo một Theme widget mới với ThemeData riêng.
2. Mở rộng theme hiện tại bằng copyWith(...) để thay đổi một phần nhỏ mà không thay đổi hết._

Next →

Text Styling and Theme WebDemo



Text Styling and Theming

1. Màu sắc và Kích thước chữ

(Sun) (Mercury) (Venus) (Moon) (Earth) (Mars) (Jupiter) [(Saturn)] [(Uranus)] [(Neptune)]

2. FontWeight và FontStyle

Am Normal :D Am Bold >:D

100 200 300 400 500 600 700 800 900

Hi am standing here :D

Help Am Falling!!!

3. Text Alignment (TextAlign.center)

I am the Moon

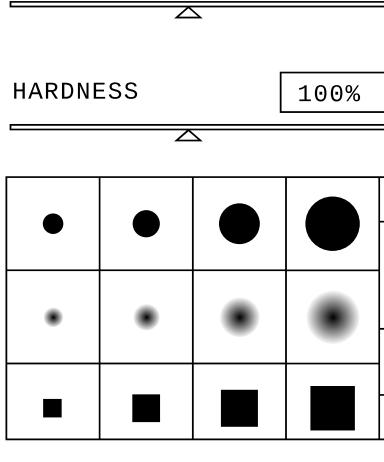
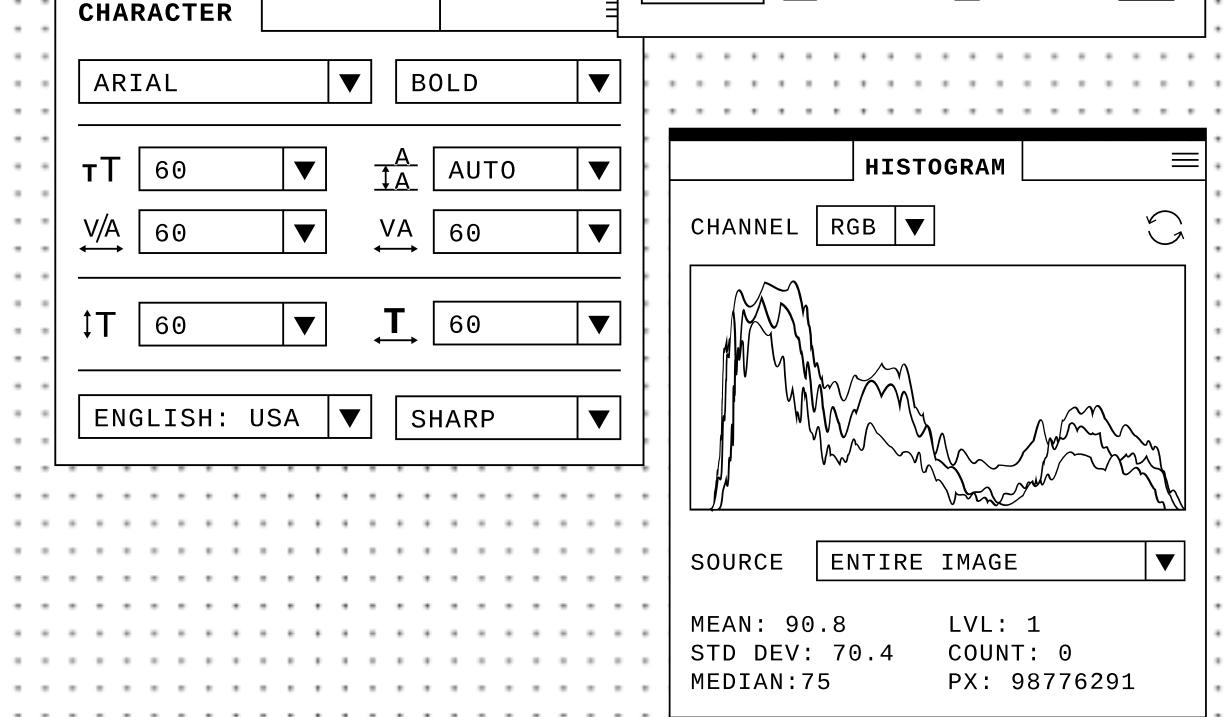
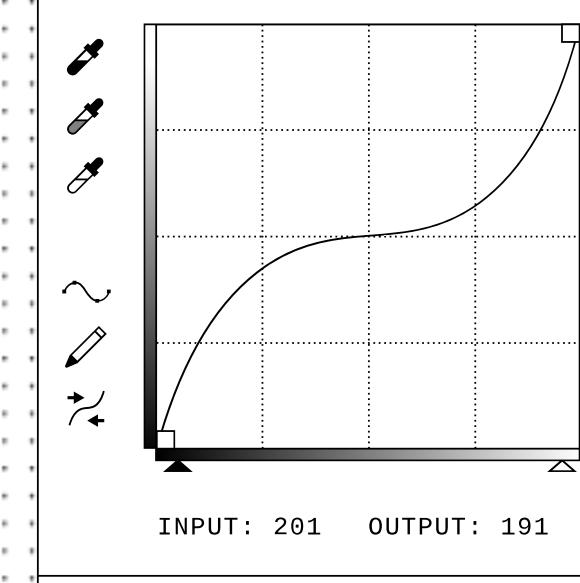
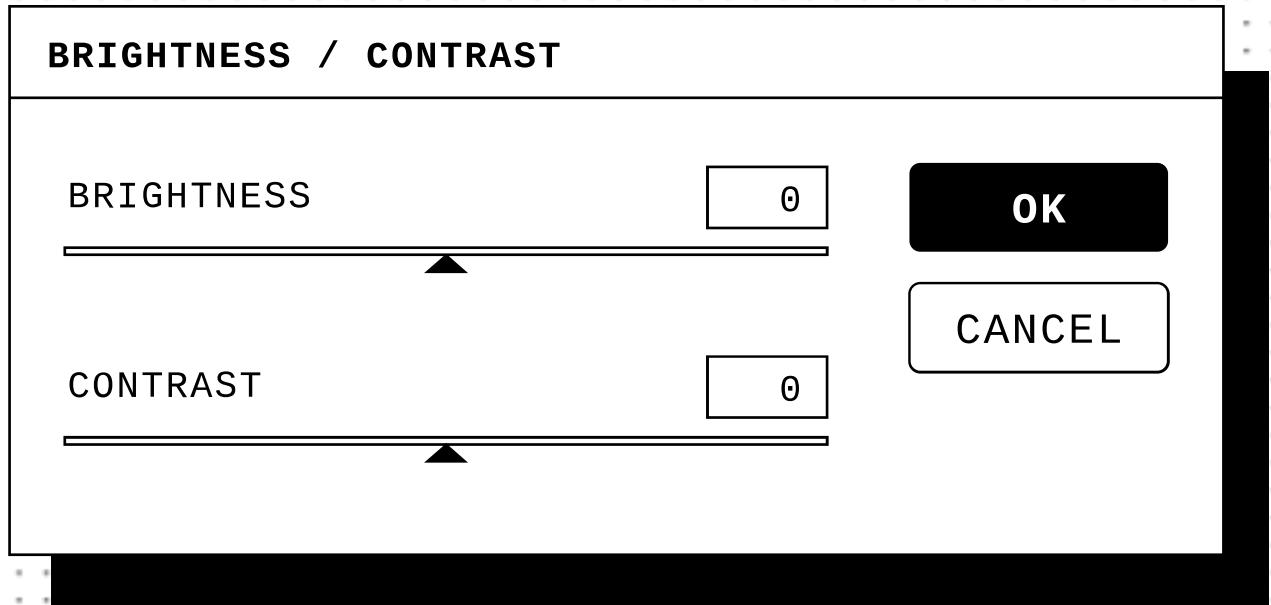
I am the Earth

I am the Sun

4. Text Overflow (TextOverflow.ellipsis)

Mặt Trời là ngôi sao ở trung tâm Hệ Mặt Trời, cung cấp ánh sáng và năng lượng cho mọi hành tinh, trong đó có Trái Đất. Nhờ có Mặt Trời mà sự sống trên Trái Đất mới tồn tại và Trái Đất quay quanh Mặt Trời theo một quỹ đạo hình elip, đồng thời tự quay quanh trục của mình. Chính sự chuyển động này tạo ra ngày và đêm, cũng như các mùa trong năm. Mặt Trăng là vệ tinh tự nhiên duy nhất của Trái Đất. Nó phản chiếu ánh sáng từ Mặt Trời, khiến bầu trời đêm trở nên sáng hơn. Khi Mặt Trời, Trái Đất và Mặt Trăng thẳng hàng, t...

DEBUG



Cảm ơn đã lắng nghe!!!

