**TÌM HIỂU VỀ CUSTOM VIEW**

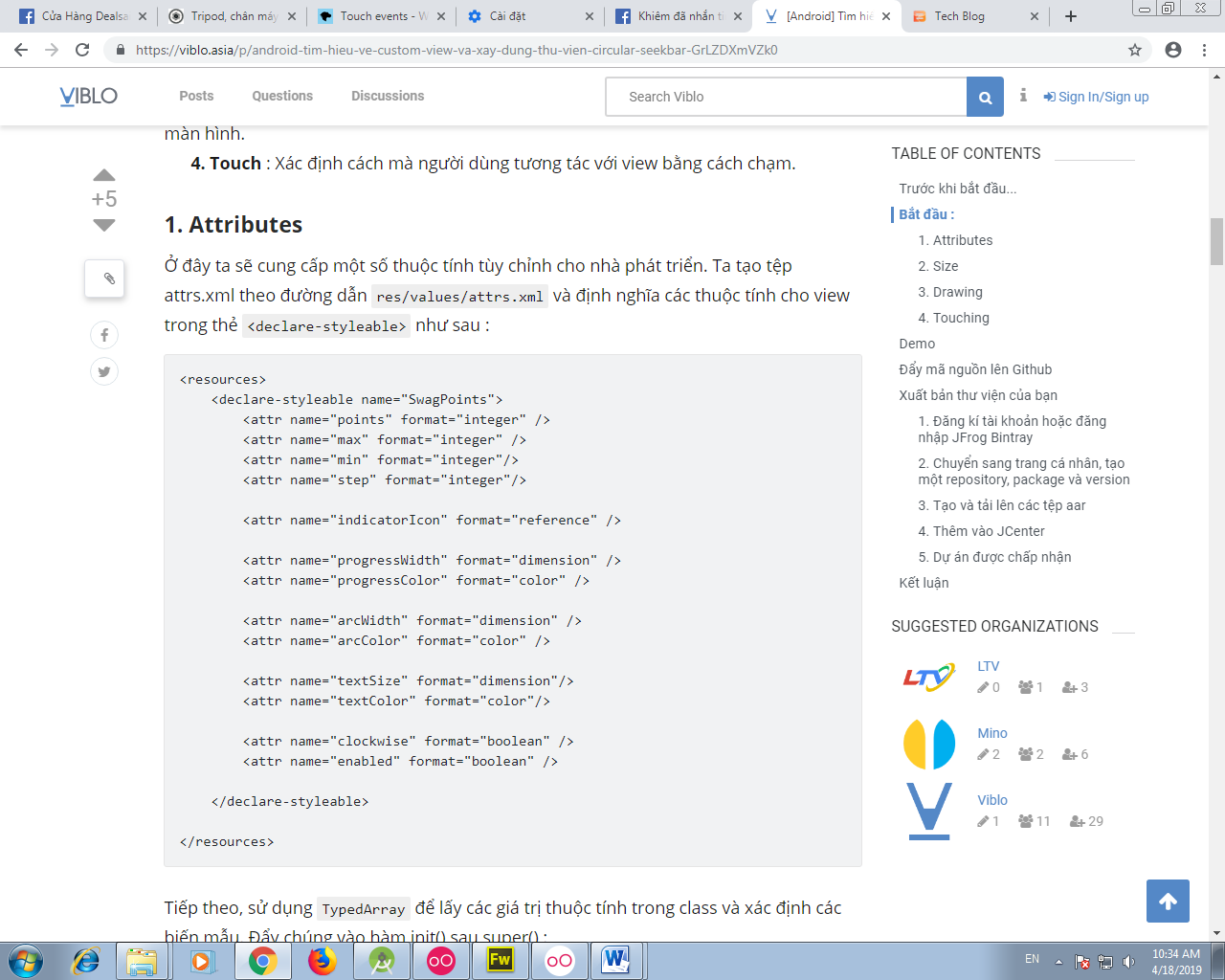
**NHÓM 8**

**Một số khái niệm ta phải quan tâm khi:**

**1. Attributes** : Xác định các thuộc tính cho phép các developer thay đổi diện mạo và hành vi trong tệp XML theo thiết kế của họ.   
**2. Size** : Xác định kích thước của View và các thành phần của nó.   
**3. Drawing** : Xác định cách mà view và các thành phần của nó được render trên màn hình.   
**4. Touch** : Xác định cách mà người dùng tương tác với view bằng cách chạm.

Để bắt đầu, đầu tiên ta phải tạo một class extends View và tạo 2 hàm tạo mặc định cho phép tạo view trong chương trình hoặc trong XML.

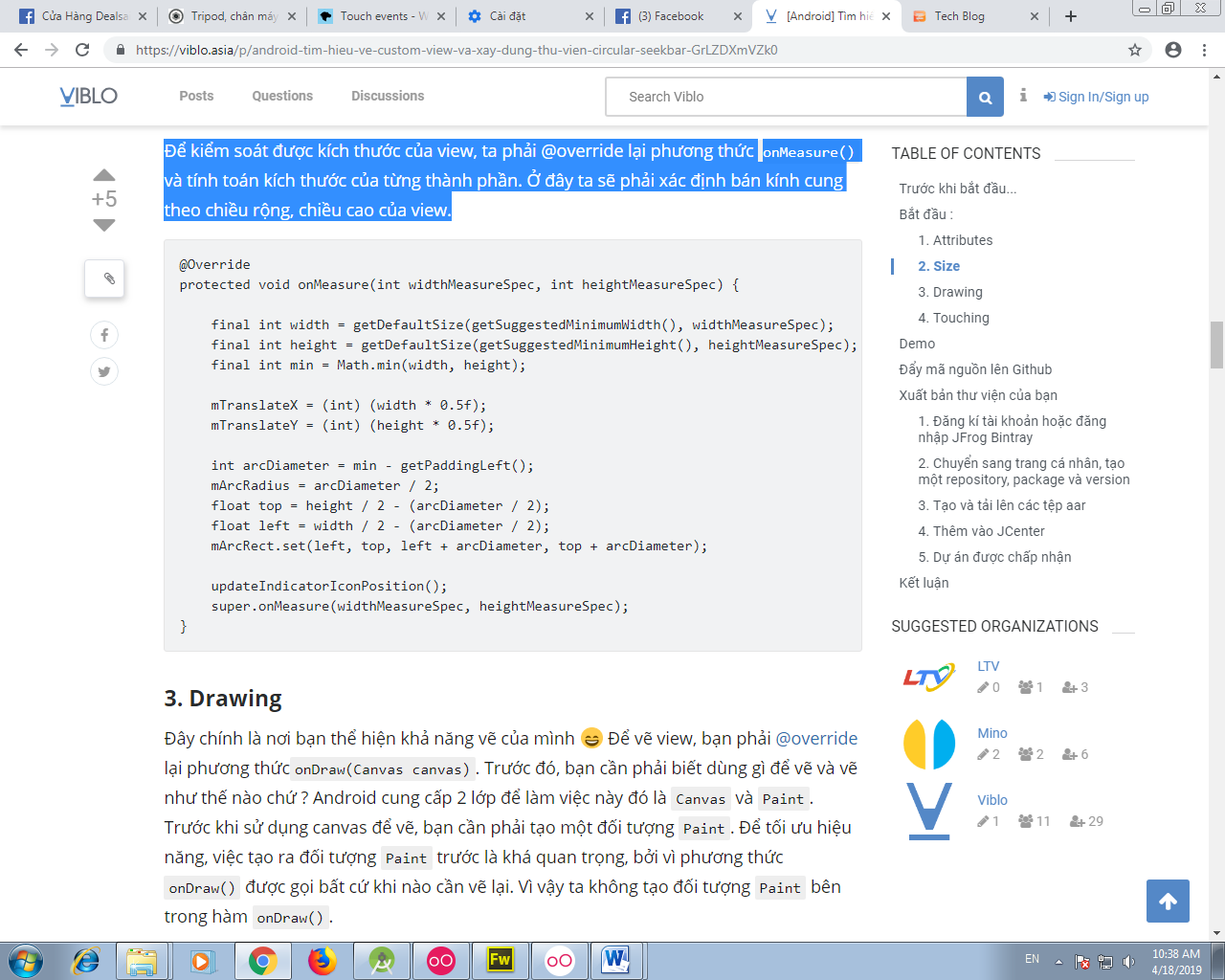
**1. Attributes**

Ở đây ta sẽ cung cấp một số thuộc tính tùy chỉnh cho nhà phát triển. Ta tạo tệp attrs.xml theo đường dẫn res/values/attrs.xml và định nghĩa các thuộc tính cho view trong thẻ <declare-styleable> như sau :

Tiếp theo, sử dụng TypedArray để lấy các giá trị thuộc tính trong class và xác định các biến mẫu. Đẩy chúng vào hàm init() sau super() :

### 2. Size

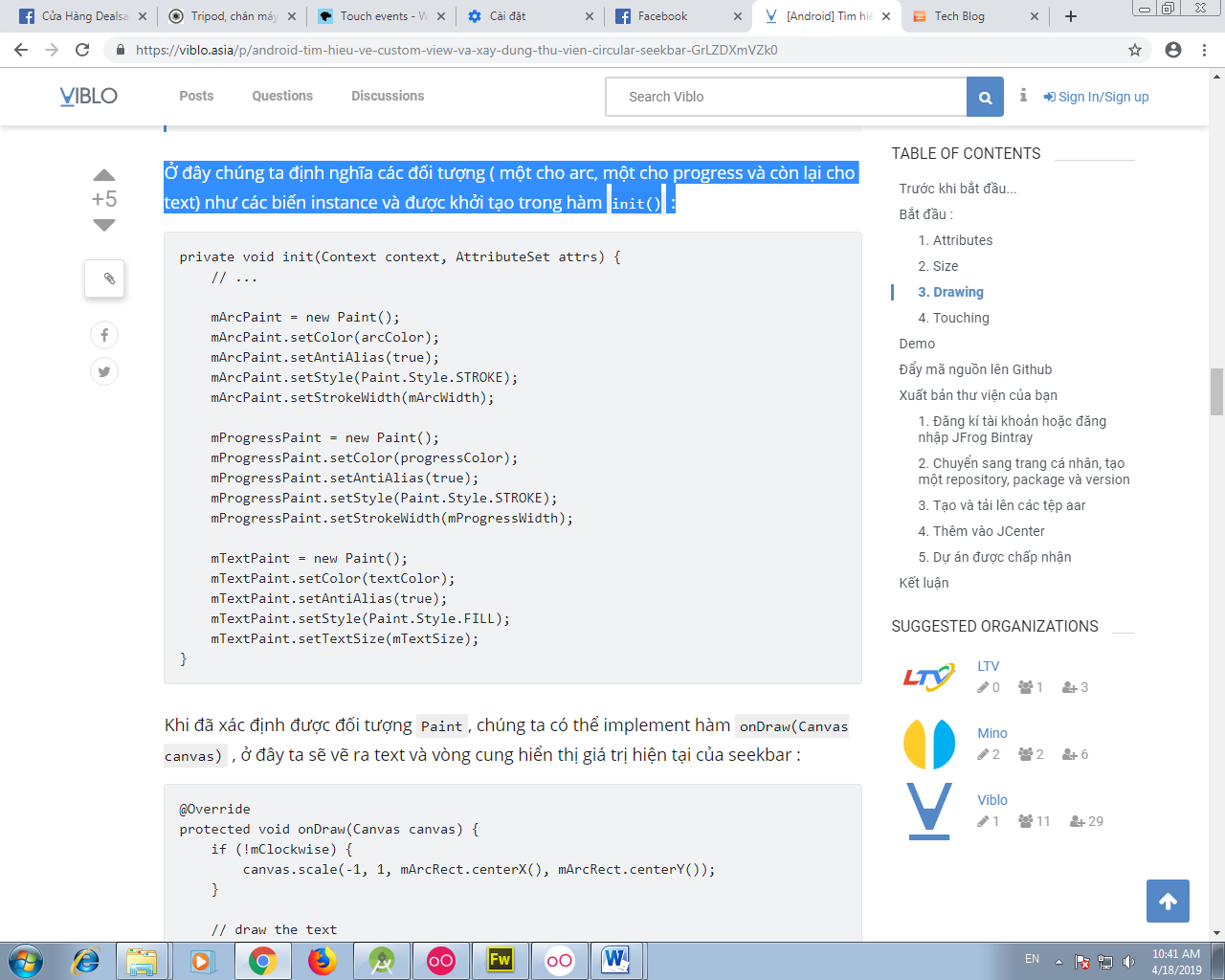
Để kiểm soát được kích thước của view, ta phải [@override](https://viblo.asia/u/override) lại phương thức onMeasure()và tính toán kích thước của từng thành phần. Ở đây ta sẽ phải xác định bán kính cung theo chiều rộng, chiều cao của view.

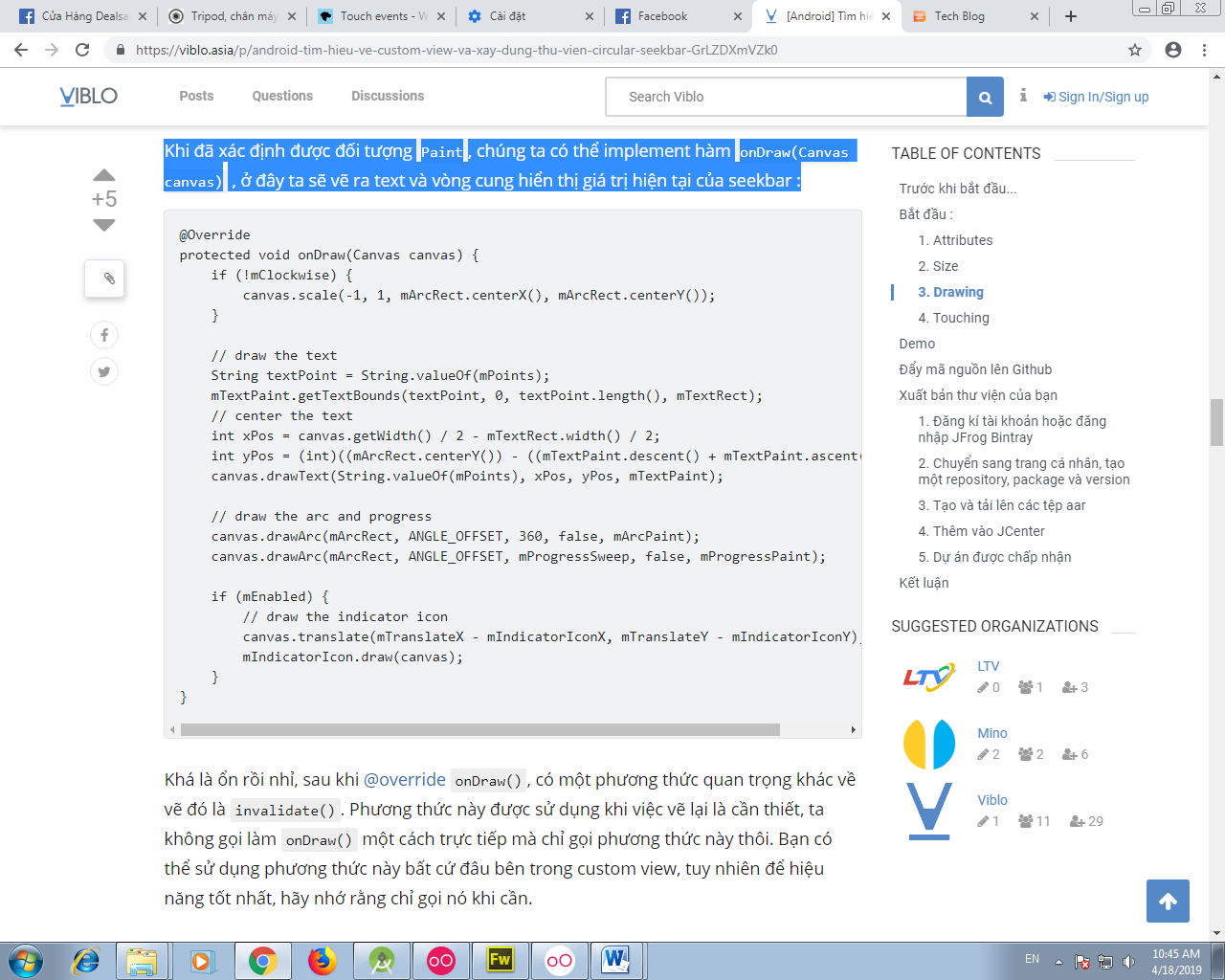


### 3. Drawing

Đây chính là nơi bạn thể hiện khả năng vẽ của mình. Để vẽ view, bạn phải [@override](https://viblo.asia/u/override)lại phương thứconDraw(Canvas canvas). Trước đó, bạn cần phải biết dùng gì để vẽ và vẽ như thế nào chứ ? Android cung cấp 2 lớp để làm việc này đó là Canvas và Paint. Trước khi sử dụng canvas để vẽ, bạn cần phải tạo một đối tượng Paint. Để tối ưu hiệu năng, việc tạo ra đối tượng Paint trước là khá quan trọng, bởi vì phương thức onDraw() được gọi bất cứ khi nào cần vẽ lại. Vì vậy ta không tạo đối tượng Paint bên trong hàm onDraw().

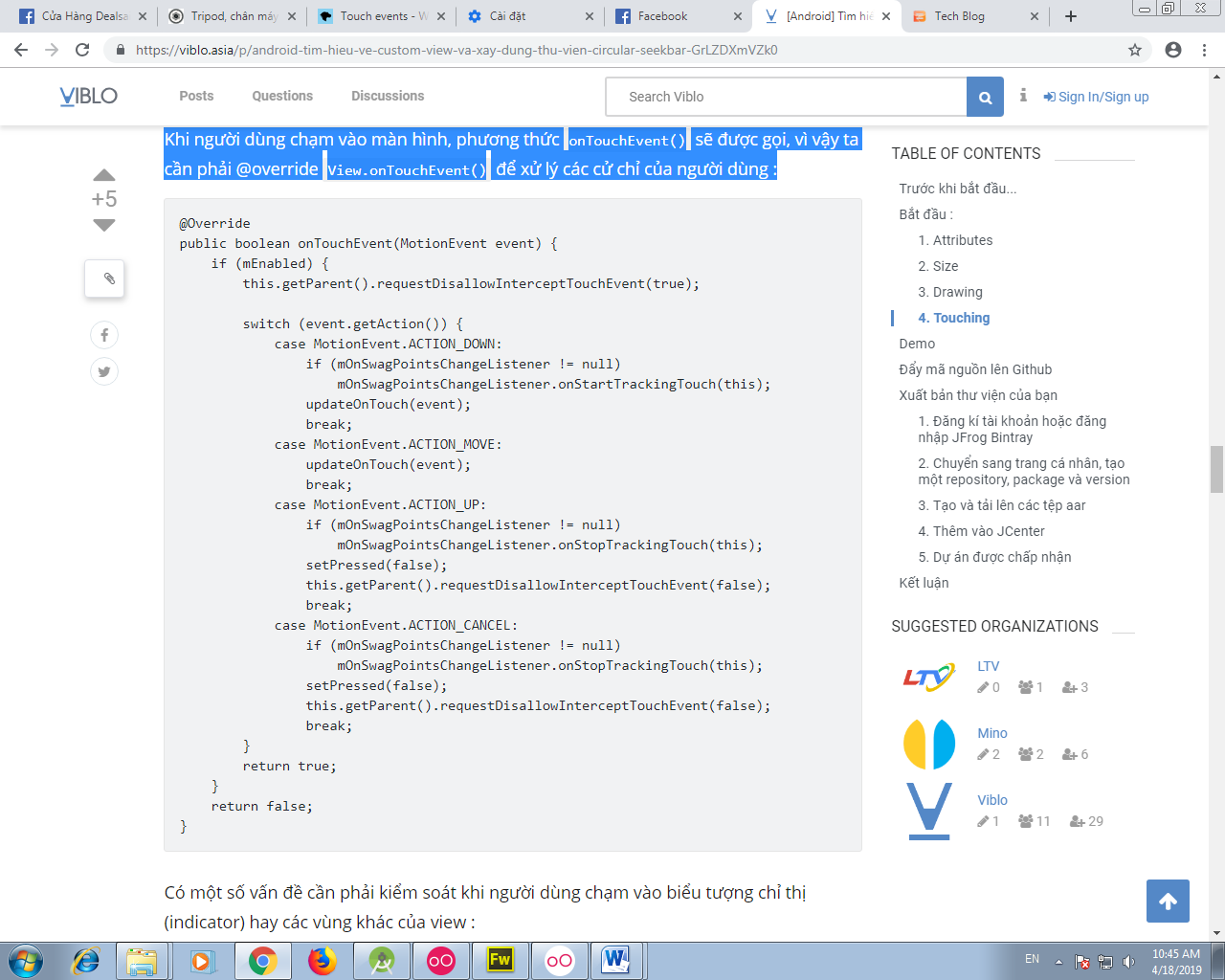
Ở đây chúng ta định nghĩa các đối tượng ( một cho arc, một cho progress và còn lại cho text) như các biến instance và được khởi tạo trong hàm init() :



Khi đã xác định được đối tượng Paint, chúng ta có thể implement hàm onDraw(Canvas canvas) , ở đây ta sẽ vẽ ra text và vòng cung hiển thị giá trị hiện tại của seekbar :

Khá là ổn rồi nhỉ, sau khi [@override](https://viblo.asia/u/override) onDraw(), có một phương thức quan trọng khác về vẽ đó là invalidate(). Phương thức này được sử dụng khi việc vẽ lại là cần thiết, ta không gọi làm onDraw() một cách trực tiếp mà chỉ gọi phương thức này thôi. Bạn có thể sử dụng phương thức này bất cứ đâu bên trong custom view, tuy nhiên để hiệu năng tốt nhất, hãy nhớ rằng chỉ gọi nó khi cần.

### 4. Touching

Khi người dùng chạm vào màn hình, phương thức onTouchEvent() sẽ được gọi, vì vậy ta cần phải [@override](https://viblo.asia/u/override) View.onTouchEvent() để xử lý các cử chỉ của người dùng :

Có một số vấn đề cần phải kiểm soát khi người dùng chạm vào biểu tượng chỉ thị (indicator) hay các vùng khác của view :

* Cập nhật lại vị trí indicator và progress text
* Vẽ progress hiện tại trên cung
* Dừng lại khi đạt max/min.
* Để biết vị trí của indicator icon và progress hiện tại để vẽ, ta cần phải chuyển đổi tọa độ chạm trên màn hình thành góc cung. Trong custom view này, ta sẽ chọn trung tâm của vòng cung làm gốc (0,0) và sử dụng các hàm lượng giác để biến tọa độ chạm thành góc của cung tròn trong khoảng (0,360), và map nó với giá trị hiện tại.

Về cơ bản là như sau :

