Visual Format Language

1. Giới thiệu

Auto Layout’s Visual Format Language (VFL) là một ngôn ngữ khai báo để xác định Auto Layout constraints cho views. Nó cho phép ta xác định ràng buộc giữa các đối tượng bằng cách sử dụng chuỗi định dạng ASCII.

Cú pháp của ngôn ngữ này rất đơn giản và dễ hiểu.

Với một đoạn mã bạn có thể điều chỉnh cóntraints theo cả chiều ngang hoặc chiều dọc. Điều này giúp tiết kiệm rất nhiều mã so với việc tạo ra các ràng buộc cùng một lúc.

Chúng ta cần phải xây dựng:

+ Cấu trúc constranints theo chiều nang và chiều dọc.

+ Định nghĩa view bên trong chuỗi VFL

+ Sử dụng ma trận constraints trong chuỗi VFL.

+ Sử dụng các layout options để bố trí các đối tượng này với đối tượng khác.

VFL rất hữu ích khi giao điện không khởi tạo trực tiếp mà phải init trong code. VFL rất hữ ích khi ta tạo Auto Layout constraints trong code. Thay vì chúng ta tạo từng constraint một VFL cho phép chúng ta tạo rất nhiều constraints trong 1 dòng code. Trước khi Autolayout được tạo ra việc resize views hết sức phức tạp, từ khi có Auto Layout và VFL việc tạo giao diện với các device có kích thước khác nhau đã trở nên dễ hơn bao giờ hết.

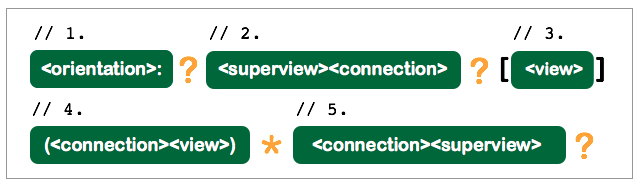
Chú ý:

Khi add auto layout bằng code thì phải thay đổi thuộc tính translatesAutoresizingMaskIntoConstraints, thuộc tính này có giá trị mặc định là true, điều này có nghĩa là Auto Layout constraints sẽ được tạo ra dựa theo autoresizing mask của các view. Do đó để ko bị conflict chúng ta sẽ set giá trị này thành false

Trước khi add constraints cho các đối tượng thì chúng cần được add vào superview nếu ko sẽ bị runtime exception.

1. Visual Formart Language format string

Các chuỗi định dạng có thể được chia thành các thành phần sau



Đầu tiên chúng ta xác định rằng constraint này sẽ được định nghĩa cho chiều ngang (horizontal) bằng cách bắt đầu string bằng chữ H, các constraints sẽ được định nghĩa sau dấu : này.  
Ký hiệu | tượng trưng cho superview, để add space giữa các thành phần ký tự - được sử dụng, chúng ta có thể đặt các giá trị số cố định để định nghĩa ra giá trị spacing này. Các views được truy cập thông qua key định nghĩa ở dictionary truyền vào constraintsWithVisualFormat, mỗi view sẽ được bọc bởi []  
Nhìn vào sting này ta có thể hiểu như sau: Theo chiều ngang Horizontally, view màu đỏ này sẽ dãn toàn bộ chiều rộng của nó so với superview ko có padding.

* 1. Orientation – Không bắt buộc

Nếu không định nghĩa thì mặc định là theo chiều ngang (H)

H: đang định nghĩa cấu trúc theo chiều ngang

V: đang định nghĩa cấu trúc theo chiều dọc

* 1. <superview><connection> - Không bắt buộc

Kết nối bên trái với view cha.

V: Neo top với view cha

H: Neo trái với view cha

* 1. <view> - bắt buộc

Ít nhất là phải xác định tên của view đang xử lý. Các views được truy cập thông qua key định nghĩa ở dictionary truyền vào constraintsWithVisualFormat, mỗi view sẽ được bọc bởi []

* 1. <connection><view> - Không bắt buộc

Neo với các view khác nếu cần thiết

Có thể neo với 1 hoặc nhiều view

* 1. <connection><superview> - Không bắt buộc

Kết nối bên phải với view cha

V: Neo bottom với view cha

H: Neo phải với view cha

**Kí tự sẵn có:**

* **| : superview**
* **- : Khoảng cách tiêu chuẩn ( thường là 8 pixel, giá trị có thể được thay đổi nếu nó là khoảng cách đến các cạnh của một superView)**
* **== : bằng với**
* **-20- : định nghĩa khoảng cách không phải là mặc định (là 20 pixel)**
* **<= : nhỏ hơn hoặc bằng với**
* **>= : lớn hoăn hoặc bằng với**
* **@250 : độ ưu tiên của constraint, giá trị trong khoảng 0 – 1000**

**250 – độ ưu tiên thấp**

**750 – độ ưu tiên cao**

**1000 – độ ưu tiên cao nhất, bắt buộc**

**Ví dụ:** H:|-[redView(==blueView)]-20-[ greenView (120@250)]-20@750-[blueView(>=50)]-|

H : theo chiều ngang.

|-[redView: redView cách mép trái view cha 8 pixels

redView(==blueView)] : chiều ngang của redView = blueView

]-20-[greenView : mép phải redview cách mép trái greenview 20 pixels.

greenView (120@250) : chiều ngang greenview là 120, độ ưu tiên thấp. Auto Layout có thể phá vỡ constraint này nếu có xảy ra xung đột.

-20@750- mép trái blueView cách mép phải greenView 20 pixel. Độ ưu tiên cao, Auto Layout sẽ không phá vỡ constraint này nếu có xung đột.

[blueView(>=50)] : chiều ngang của blueView lớn hoặc bằng 50 pixels.

-| : blueView cách mép phải view cha 8 pixels.

1. Layout Option

### Vertical Constraint Options

* NSLayoutFormatAlignAllLeft – Căn trái tất cả các view
* NSLayoutFormatAlignAllRight – Căn phải tất cả các view
* NSLayoutFormatAlignAllLeading – This aligns all of the views on the edge that is the start of text in the current locale (English: left, Hebrew: right).
* NSLayoutFormatAlignAllTrailing – This aligns all of the views on the edge that is the end of text in the current locale (English: right, Hebrew: left).
* NSLayoutFormatAlignAllCenterX – Căn giữa theo chiều X tất cả các view

### Horizontal Constraint Options

* NSLayoutFormatAlignAllTop – Căn top tất cả các view
* NSLayoutFormatAlignAllBottom – Căn bottom tất cả các view
* NSLayoutFormatAlignAllCenterY – Căn giữa theo chiều Y tất cả các view
* NSLayoutFormatAlignAllBaseline

1. Metric

Sử dụng khi muốn để các giá trị kích thước nằm ngoài chuỗi định dạng , được định nghĩa trong viewDictionary

1. Layout Guides