これで、コンテナを使用して、テキストや画像などの単一のウィジェットをレイアウトするのが非常にうまくなりました。 そして、パディング、マージン、サイズ、背景色などを定義できます。しかし、互いに相対的にレイアウトしたいものがたくさんある場合はどうでしょうか？

その場合、1人だけではなく、たくさんの子供を連れて行くことができるものが必要です。

これを行う最も簡単な方法は、列と行を使用することです。これにより、ウィジェットを垂直方向または水平方向にレイアウトし、それらのサイズを決定することができます。

ウィジェットカタログに戻り、レイアウトウィジェットを具体的に見て、複数子レイアウトウィジェットセクションに進みます。また、複数の子ウィジェットをレイアウトするために最も一般的に使用されるものは行と列です。行は水平に、列は垂直になります。先に進み、独自の列を作成しましょう。

コンテナが1つだけでなく、たくさんの異なるコンテナがある場合はどうなりますか？だから私は先に進み、余白とパディングを取り除くつもりですので、私たちは自分自身を混乱させないで、私はそれに名前を付けます

コンテナ1。

これがコンテナ1で、白です。

そして、コンテナを折りたたむつもりです。したがって、コンテナのプロパティの設定に関連するすべてのコードがすべてこれらの括弧で囲まれていることがわかります。クリックすると、展開できます。しかし、このように圧縮することもできます。私たちの安全な地域の子供を持つ代わりに、

それで、私たちの安全な領域の中にあるのは単なるコンテナです。それを切り取ってここに貼り付けます。少し後で説明します。代わりに、子は列になります。今、非常に重要なコラムには、子がいません。

列の中にたくさんのものを入れることができるので、子があります。そして、これが実際にここに入れることができるアイテムのリストであることを伝えるために、この山かっこがあります。

これらの角括弧の間に多くのウィジェットを配置できます。そして、私たちが始めようとしているのは、今移動したコンテナです。

ここでCOMMAND + Xを押してここにカットアンドペーストし、それから折りたたんでいきます。そして、この時点で保存ボタンを押すと、複数のウィジェットがある場合にのみ実際に動作する列があるため、画面上で実際にはほとんど何も起こらないことがわかります。

最初のコンテナを離れて、列の2番目の子として別のコンテナを作成しましょう。

このコンテナの幅は100になり、高さは100 になります。すべての行を知っていると言う最後にカンマを追加し、保存をクリックすると、Dart が簡単に知ることができます。コードのレイアウト方法。

コンテナに色を付けて、実際に見えるようにします。

これを青色に変更しましょう。

保存します。画面に2つのコンテナが表示されていることを確認できます。これにテキストの一部である子を指定し、「Container 2」という単語を記述するように指示します。コンテナ1とコンテナ2があります。

そして、このコンテナをコピーして3つ目のコンテナを作成します。

そして、3番目のものは赤になります。そして、コンテナ3になります。

そのため、100 x 100のコンテナが3つあります。

すべて異なる色で、垂直方向に配置されています。これは、すべてのコンテナが列内にあるためです。これで、列は自動的に可能な限り垂直方向のスペースを占有しようとします。

Flutter Inspectorに移動して、列が実際にどのように見えるかを確認すると、このボックスがここにあることがわかります。

垂直に利用可能なスペースをすべて占有しようとしています。ただし、水平方向では、子のサイズに制限されます。

ただし、これは変更できます。

mainAxisSize を変更できます。列の主軸はもちろん垂直です。

これを最小サイズに変更しましょう。

これで、保存をクリックすると、列が実際に縮小して、3つのコンテナ（そのすべての子）を収めるのに必要なサイズになります。

現在、私たちのコラムは子を上から下に配置しているので、コンテナ1、2、3です。

では、代わりに他の方向に進みたいとしたらどうでしょうか？

さて、垂直方向を変更できます。

上に移動するか、下から上に移動するかを変更できます。

これが1、2、3です。または、上から下に移動するデフォルトのダウンとして変更することもできます。

では、方向を変えたくなくて、代わりにコンテナ間の間隔を変えたいとしたらどうでしょう。

それでは、mainAxisAlignment と呼ばれるものを使用します。デフォルトでは、すべてのコンテナが先頭に揃えられます。

したがって、mainAlignment.start です。そして、これは子を可能な限り主軸の開始点の近くに配置しようとします。

これが垂直軸であり、これが垂直軸の開始点です。

ただし、変更して終了することもできます。

私たちが持っていたのであればMainAxisAlignment.endをその後、私たちのコンテナは、すべての方に、ここでダウン束ねされます私たちの主軸の終わり。

これで、すべてを中央に移動することもできます。また、これらのさまざまなものを使用して、画面のレイアウト方法を決定できます。

さて、実際にすべての子供をこのように、最初または最後の中央にまとめて欲しくない場合はどうでしょう。それらをもう少し広げたいとしたらどうでしょう。「spaceEvenly 」と呼ばれるものを使用できます。

これにより、列の空き容量が計算され、すべての子が列全体に均等に配置されます。

しかし、何かを最初に、何かを一番下に、次にもう1つを中央に配置する場合はどうでしょうか。

それでは、代わりにspaceBetween を使用します。

ここで、コンテナを水平軸に揃える方法を変更したい場合はどうすればよいですか？

この場合、メイン軸アライメントプロパティを使用する代わりに、クロス軸アライメントと呼ばれるものを使用します。

そして、これは主軸に交差または垂直です。

したがって、列の場合、それらは水平になります。

ここで、start CrossAxisAlignment 、およびend CrossAxisAlignmentにもアクセスできます。

しかし、あなたが気づくでしょう、私が先に進んでこれを入れて、ホットリロードするためにセーブを押した場合、水平軸には何も実際には起こりません。

この理由は、私の軸のアライメントが実際に私の子供に合わせようとしているからです。

つまり、もし私がはるかに幅の広い子供がいるなら、たとえば300 人の子供がいるとすれば、すべての子供はお互いの右端に揃うことになります。

しかし、それらがすべて同じサイズである場合、それらはすでに端に揃えられているため、右側に飛び出すことはありません。

この場合。

ここで、3つのコンテナーすべてを右に移動したい場合、これを行う1つの方法は、単純に見えないコンテナーを作成し、無限の幅を与え、無限の幅を与えることです。 無限これが何をするかは、ここにコンテナを作成することです。高さがないことがわかります。

そのため、幅があるだけで、画面で許容される幅です。

それで、私があなたがより簡単にそれを見ることができるようにそれを小さい高さに与えるならば、この場合、それはちょうどこの最後のビットです。

色がありません。

そのため、完全に見えませんが、すべての子を右にプッシュします。水平軸の列の最後にすべて揃えられ、コンテナの1つが画面の許容範囲内に収まっているためです。

これを行う代わりに、実際にすべてのコンテナを引き伸ばして、幅全体を埋めたいとしたらどうでしょうか？

これで、各コンテナの幅をdouble.infinity に設定できます

それでは、幅をdouble.infinity に設定します。

そのため、今ではすべてのコンテナが可能な幅全体に伸びています。しかし、実際にこれを行うには、列に付属するプロパティを使用する方がはるかに簡単です。そして、CrossAxisAlignment.endを使用する代わりに、我々はCrossAxisAlignment.stretchを使用することができる。そして、これは子供たちを引き伸ばすので、彼らは自分たちの最高のバージョンになることができます。この場合、実際に画面全体に子供たちを引き伸ばします。

したがって、リロードしても、すべてがまったく同じに見えることがわかります。

また、コンテナが親である列によって引き伸ばされているため、コンテナの幅プロパティを指定する必要がなくなりました。

コンテナ間に少し間隔を空けたい場合はどうでしょう。現時点では、彼らはお互いのビジネスに非常に多く、間にスペースがない状態で隣り合っているからです。

コンテナ間に必要なスペースが正確にわかっている場合は、サイズ設定ボックスを使用できます。そして、サイズの付いた箱はもちろん高さと幅を持つことができます。

ただし、垂直に配置された列内に配置する場合は、実際に高さを指定するだけです。したがって、高さを20に設定し、widthプロパティをゼロのままにしておくことができます。

ご覧のように、これはコンテナ1とコンテナ2の間にあり、これら2つのものの間に少しの間隔があります。

そして、それは私たちが子供たちの間に少しのスペースが欲しいときに私たちを助けます。これで、列について説明したことはすべて行にも適用されます。

実際、列を行に変更するだけで、水平軸でもまったく同じことがわかります。

そのため、この場合、交差軸のアライメントと言っています。

したがって、行の場合、主軸はもちろん水平ですが、交差軸は整列します。

そのため、すべてのコンテナが垂直方向に引き伸ばされるように、垂直方向の配置を引き伸ばす必要があります。

また、すべての子の高さは、伸びているため実際には考慮されませんが、たとえば幅を指定することもできます。

最初の幅を30にするとします。

私のコンテナは水平軸上に配置されているため、影響を受けることがわかります。

これで、高さを指定しているのに、サイズが設定されたボックスが消えたことがわかります。ただし、幅がない場合は、実際には問題ではありません。

だから、行で、それにスペースを与えるために、我々はしているつもり与えるために幅の大きさの箱を使用してそれを間隔を少し。

そのため、列、行、およびコンテナを知るための最良の方法は、実際に試してみることです。

そのため、次のレッスンでは、あなたに挑戦します。

コンテナ、列、行、およびこれまでに学んだすべてを使用して複製してほしいレイアウトがあります。

今、あなたがこれをいじっている間、人間として、私たちはかなり視覚的である傾向があります。そして通常、実際にコードを書き出す前に、何かがどうなるかを確認する方が簡単です。

そのため、Tomek によって作成されたこのFlutterレイアウトの虎の巻をプルアップすることをお勧めします。コースのすべてで、コースリソースにリンクされています。そして、彼がやったことは、列や行などを取り上げ、コードを書く前にどのように見えるかを見せようとしたことです。

したがって、この場合、列でメインのAlignment.start を使用すると、次のようになります。メインアライメントを使用する場合、行から開始しないでください。そして、少なくとも列と行について、私たちが話してきたすべてのことを彼が入れていることがわかり、コードを書くことにコミットする前に視覚的に識別することができます。課題を完了するには、次のレッスンに進んでレイアウトを確認してください

達成してほしい。そして、先に進んで自分でコードを記述し、それを機能させてまったく同じように見えるかどうかを確認します。

それでは、次のレッスンでお会いしましょう。