

Grids horizontais, Grids flexíveis e Control Flex

GeneXus™

Quando vimos a aplicação da agência de viagens finalizada no início deste curso, pudemos ver as atrações rolando horizontalmente como se fosse um carrossel. Neste vídeo veremos como configurar um grid para que adote esse comportamento, assim como outros tipos de grids e controles flexíveis que temos disponíveis para desenvolver nossa aplicação.

Smart grid

Vamos começar com o grid padrão, no qual podemos configurar uma rolagem horizontal.

Configurando a rolagem horizontal em um grid

Layout | Rules | Events | Conditions | Variables | Documentation

Application Bar

MainTable | Grid1

Country &CountryId

Name From &AttractionNameFrom

Name To &AttractionNameTo

GRID

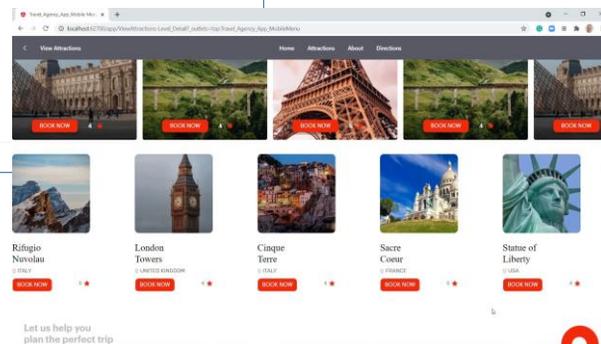
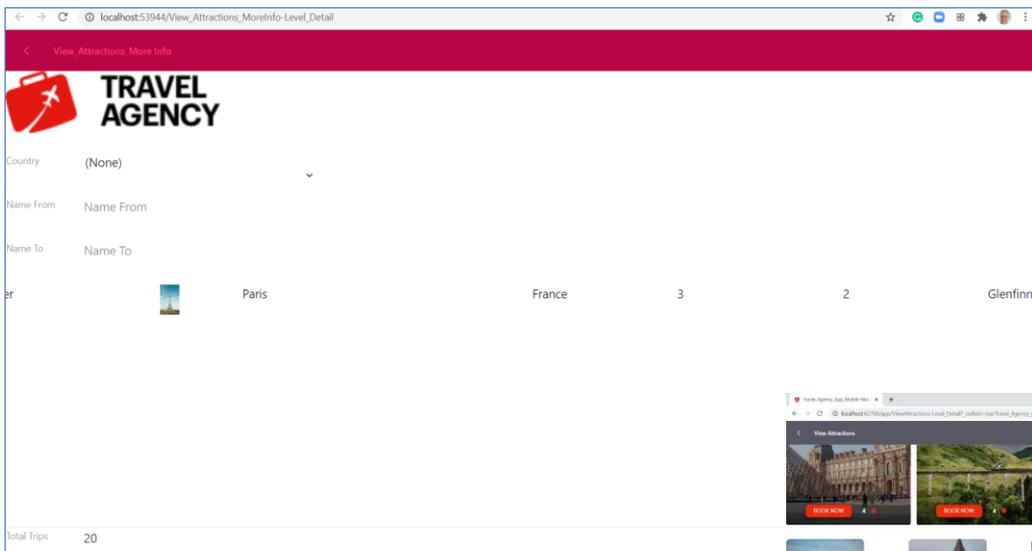
AttractionId	AttractionName		CityName	CountryId	CountryName	&Trips
--------------	----------------	---	----------	-----------	-------------	--------

Total Trips &TotalTrips

Control Info	
Control Type	Grid
Auto Grow	False
Scroll Direction	Horizontal
Snap To Grid	False
Items Layout Mode	Single
Appearance	
Class	Grid
Visible	True
Invisible Mode	Keep Space
Enabled	True
Rows	<default>
Empty Grid Background Image	(none)
Empty Grid Background Class	Image
Empty Grid Text	
Empty Grid Text Class	TextBlock

Desde o upgrade 11 da versão GeneXus 16, o controle grid passou a ser um Smart Grid, por isso temos propriedades que nos permitem alterar a aparência de como será exibida a informação, de uma forma mais flexível.

Se a única coisa que queremos é que a rolagem dos registros seja em direção horizontal, podemos definir a propriedade Scroll Direction do grid no valor Horizontal e se executamos...

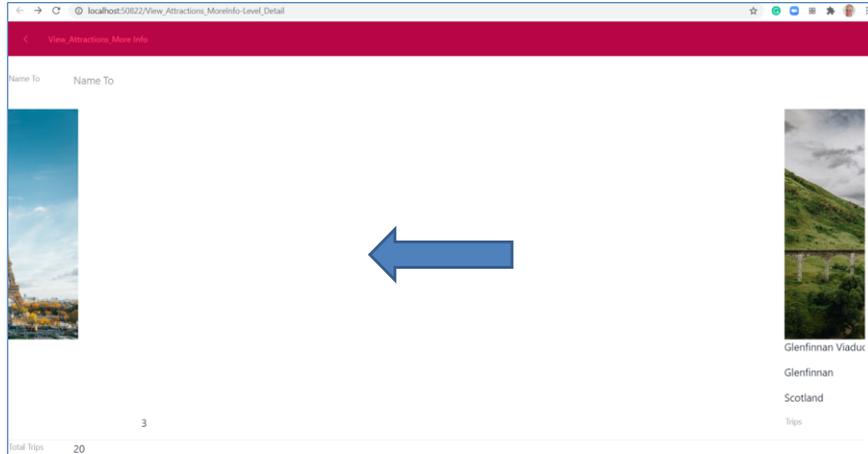
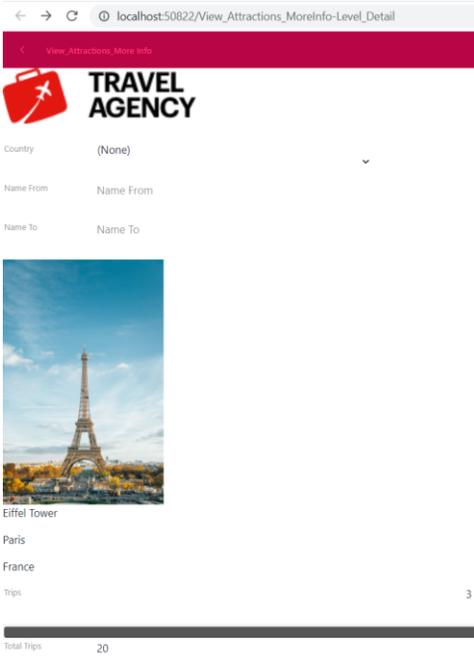


Vemos que agora o grid ocupa uma única linha e clicando com o mouse podemos mover para a direita e depois podemos retornar à esquerda.

Se voltamos para a aplicação finalizada, para a seção da tela onde são percorridas todas as atrações disponíveis, vemos que além da rolagem horizontal, podem ser vistas várias atrações ao mesmo tempo, portanto, devemos ver como fazer para configurar isto.

Mas também, para cada atração os dados da atração estão dispostos de forma diferente do que conseguimos em nossa execução.

Novo design e rolagem horizontal



Vemos que agora a apresentação dos dados está mais parecida com o que queríamos, mas ainda temos o problema de vermos apenas um registro por vez na tela.

Propriedade: Items Layout Mode

The screenshot displays the GeneXus IDE interface. On the left, a form is shown with a grid component. The grid contains the following fields: AttractionId, CountryId, AttractionName, CityName, CountryName, and Trips. The grid is currently set to 'Single' layout mode. On the right, the Properties Panel is open, showing the 'Items Layout Mode' property set to 'Single'. The 'Appearance' section is expanded, showing various options for the grid's look and feel.

Control Info	
Control Type	Grid
Auto Grow	False
Scroll Direction	Horizontal
Snap To Grid	False
Items Layout Mode	Single
Appearance	
Class	Multiple by Quantity
Visible	Staggered by Quantity
Invisible Mode	Multiple by Size
Enabled	True
Rows	<default>
Empty Grid Background Image	(none)
Empty Grid Background Class	Image
Empty Grid Text	
Empty Grid Text Class	TextBlock

Além da propriedade Scroll Direction do grid, temos a propriedade Items Layout Mode que nos permite alterar a forma como aparecem os dados do grid.

Existem vários valores disponíveis:

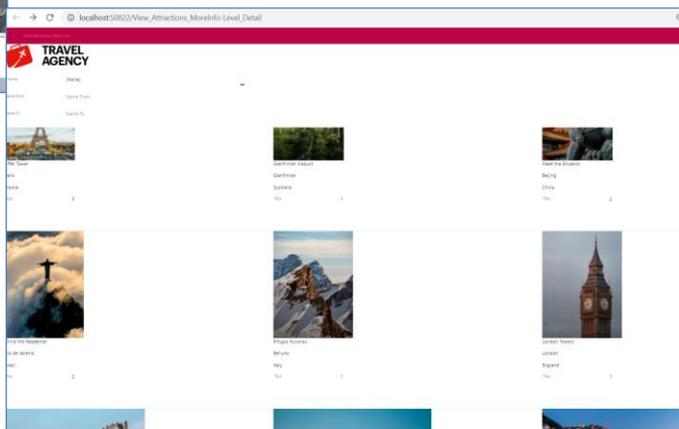
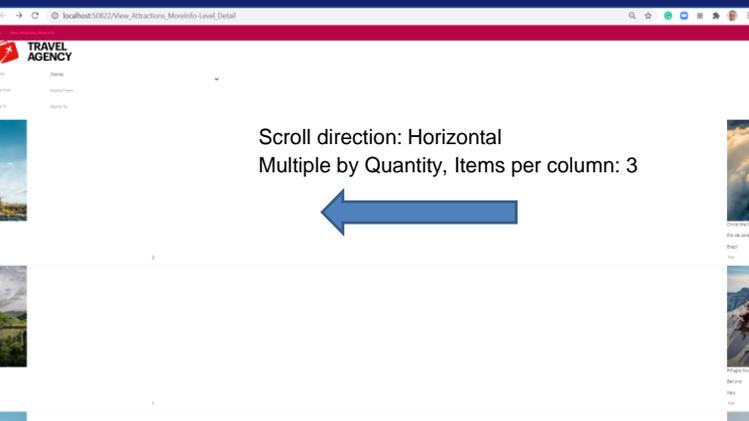
Single – É o valor padrão e faz com que o grid mostre apenas um elemento por linha.

Multiple by Quantity – Este valor indica que será exibido um número fixo de elementos em cada linha / coluna do grid. Será determinado pelas propriedades Items Per Row se a direção de scroll for vertical ou Items Per Column, se a direção de scroll for horizontal.

Staggered by Quantity – O grid é exibido com um design escalonado, onde cada elemento pode ter um tamanho diferente. A quantidade de elementos por linha ou coluna é determinado pelas propriedades Items Per Row ou Items Per Column, dependendo da direção de rolagem do grid.

Multiple by Size - O número de elementos exibidos em cada linha / coluna será determinado pelo tamanho de cada elemento. Para isso, são utilizados os valores das propriedades Maximum Width, Minimum Width se a direção de scroll for vertical e Maximum Height e Minimum Height se a direção de scroll for horizontal.

Vamos atribuir o valor Multiple by Quantity e como temos a direção de scroll como horizontal, aparece a propriedade Items Per Column. Colocamos o valor 3 e executamos.

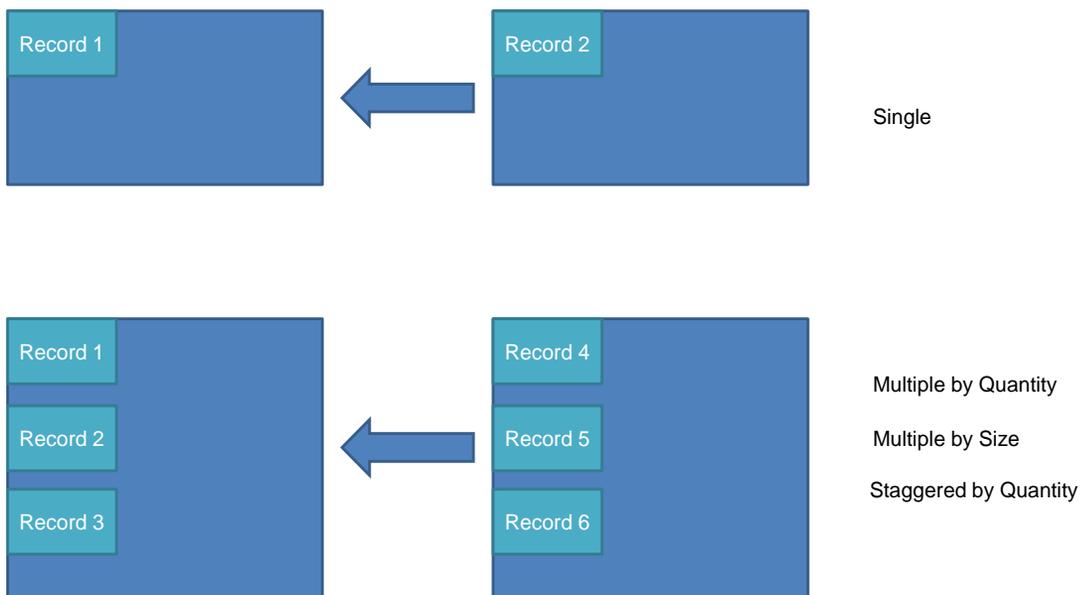


Vemos que visualizamos uma única coluna na página e em cada coluna 3 atrações que vemos se rolarmos para baixo. Se paginamos para a direita, aparecem as 3 atrações seguintes, que aparecem para baixo.

Isto não é o que queremos, vamos mudar o scroll para vertical e deixamos as propriedades com os mesmos valores.

Agora vemos em cada linha 3 atrações e para ver as próximas devemos mover para baixo.

Propriedade: Items Layout Mode, Scroll direction = horizontal

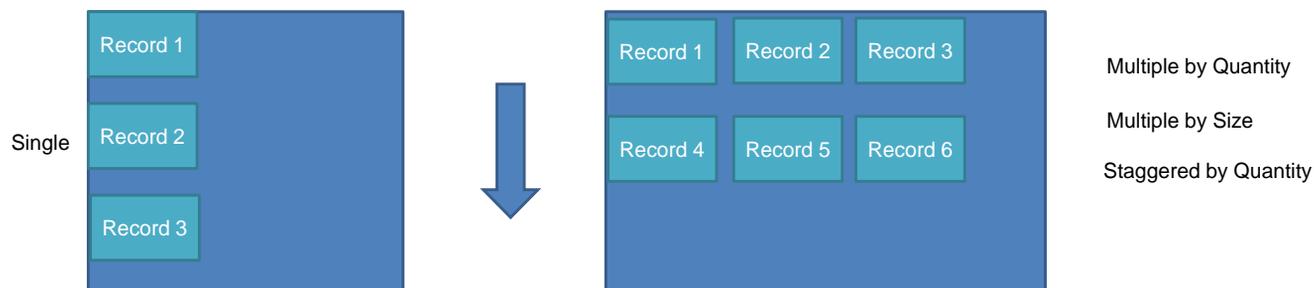


Em resumo, no grid padrão (smart grid), se a rolagem for horizontal como no nosso exemplo, podemos definir quantos registros queremos ver por coluna, ou seja, dentro da mesma coluna para baixo.

Mas sempre estaremos vendo uma coluna de cada vez.

A quantidade de registros por coluna que são vistos para baixo podemos definir como um valor fixo, ou que dependa da altura (mínima e máxima) que queremos dar a cada item. Se definirmos que o faremos com um valor fixo, temos a possibilidade de que seja visto escalonado ou não, dependendo se escolhermos Staggered by Quantity ou Multiple by Quantity, respectivamente.

Propriedade: Items Layout Mode, Scroll direction = vertical



Se a rolagem for vertical, podemos configurar quantos registros queremos ver em cada linha.

Ou seja, que a próxima linha voltará a trazer o mesmo grupo de registros. E isto também podemos definir como um valor fixo (escalonado ou não), ou que dependa da largura (mínima e máxima) que queremos dar a cada item.

Ou seja, não conseguiremos com o Smart Grid o comportamento que precisamos, de poder ver todas as atrações que cabem na largura disponível da tela e que podemos rolar para a direita se ainda houver mais que não podemos ver.

Horizontal Grid

Já vimos várias opções do grid padrão para exibir informações. Veremos agora outro controle denominado Horizontal Grid.

Horizontal grid

The screenshot shows the GeneXus IDE interface. On the left, a form layout is visible with a grid control highlighted in blue. The grid contains several text input fields: AttractionId, CountryId, AttractionName, CityName, CountryName, and Trips. Above the grid are fields for Country, Name From, and Name To. Below the grid is a Total Trips field. On the right, the Properties panel is open, showing the Control Info section. The Control Type is set to Horizontal Grid. The Columns Per Page Portrait property is set to 5, and the Rows Per Page Portrait is set to 1. The Appearance section shows the Class set to Grid and Visible set to True.

Properties	
General Class	
Control Info	
Control Type	Horizontal Grid
Auto Grow	False
Paged	True
Show Page Controller	True
Page Controller Class	PageController
Columns Per Page Portrait	5
Rows Per Page Portrait	1
Columns Per Page Landscape	1
Rows Per Page Landscape	1
Appearance	
Class	Grid
Visible	True
Invisible Mode	Keep Space
Enabled	True
Rows	<default>
Empty Grid Background Image	(none)
Empty Grid Background Class	Image
Empty Grid Text	
Empty Grid Text Class	TextBlock

Se vamos para a propriedade Control type, vemos que existem vários controles que podemos usar para que as informações sejam exibidas. Escolhemos o valor Horizontal Grid.

Ao atribuir este valor, será habilitada uma série de propriedades através das quais podemos configurar como será visto o grid. Por exemplo, a propriedade Show Page Controller determina se será visto ou não o paginador, e a propriedade Page Controller Class nos permitirá definir a classe na qual está baseada este paginador.

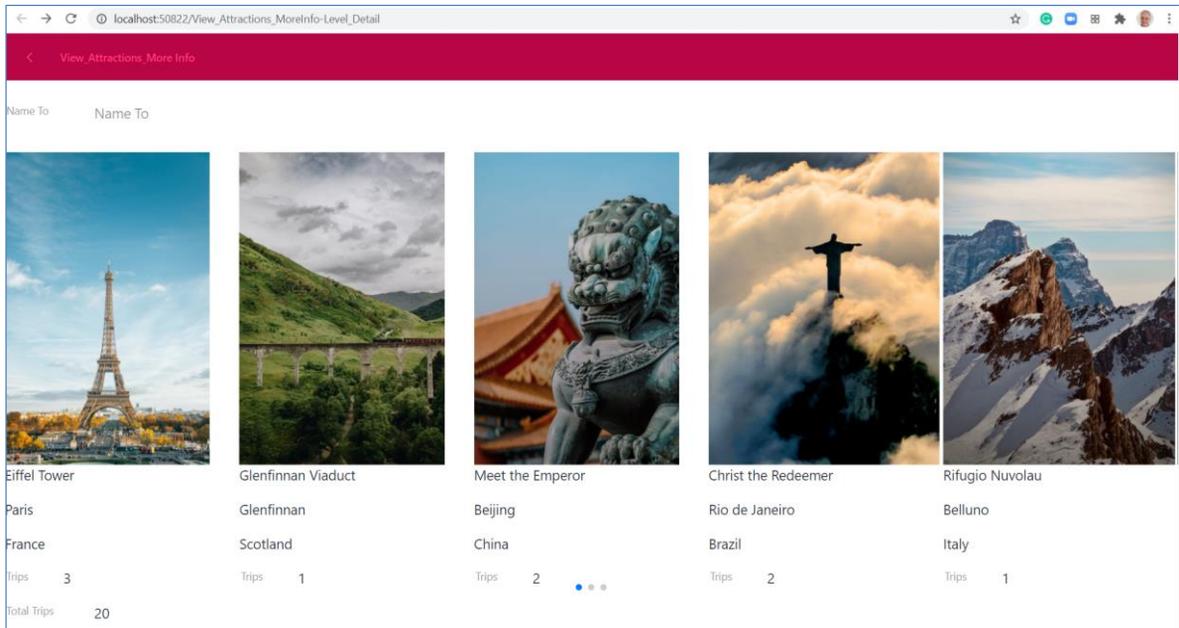
Esta classe é como um template para o qual podemos alterar propriedades e todos os controles baseados na classe herdarão esses valores.

No entanto, GeneXus 17 nos oferece uma forma muito mais flexível, escalável e poderosa de fazer isto, que é através de um Design System Object, que estudaremos mais adiante.

Voltando às propriedades que nos permitem personalizar nosso grid horizontal, vemos que podemos definir quantas colunas queremos ver e ao mesmo tempo quantas linhas queremos em cada página. Vamos definir a propriedade Columns Per Page Portrait com valor 5 e deixamos a propriedade Rows Per Page Portrait com valor 1 porque queremos ver apenas uma linha por página.

Executamos.

Usando um controle Horizontal Grid



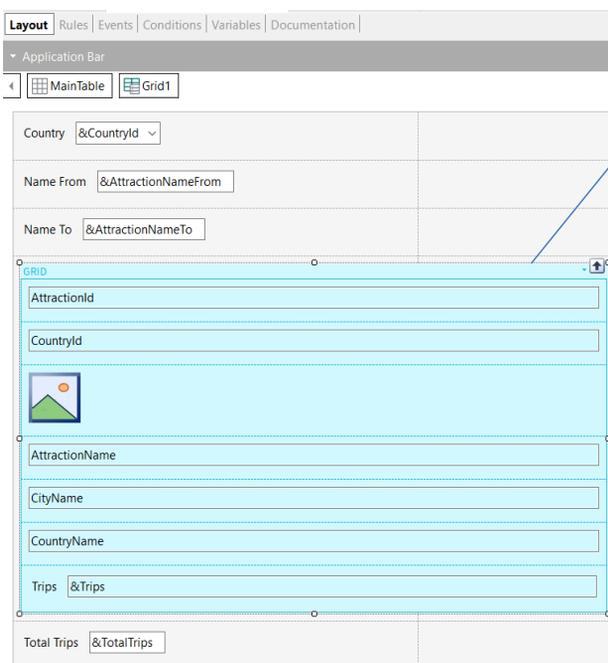
Vemos que nos aproximamos muito da forma como queríamos vê-lo. Agora podemos paginar para a direita e veremos as atrações ocupando toda a página, 5 atrações por vez.

Se tivéssemos alterado também a propriedade Rows Per Page Portrait, poderíamos ver várias linhas de atrações, mas não nos interessa em nosso caso.

Observemos abaixo da página no controle de paginação, que nos informa em qual página estamos.

Flex Grid e Flex Control

Já alcançamos um resultado aceitável para nossa aplicação. Veremos agora outro grid ainda mais flexível e um container flexível que permite organizar conteúdo ainda desconhecendo seu tamanho.



Control Info

Control Type	Flex Grid
Auto Grow	False
Flex Direction	Row
Flex Wrap	No Wrap
Justify Content	Flex Start
Align Items	Stretch
Align Content	Stretch

Appearance

Class	Grid
Visible	True
Invisible Mode	Keep Space
Enabled	True
Rows	<default>
Empty Grid Background Image	(none)
Empty Grid Background Class	Image
Empty Grid Text	
Empty Grid Text Class	TextBlock

Para este exemplo, primeiro fazemos um Save As do painel View_Attractions_MoreInfo e o nomeamos View_Attractions_MoreInfo 2.

Para usar o grid flexível, vamos até a propriedade Control Type do grid e escolhemos Flex Grid.

Flex Grid é um tipo de grid FreeStyle, onde podemos carregar quantos elementos quisermos, dinamicamente.

A forma como aparecem as informações depende do que definimos nas cinco propriedades que vemos na tela: Flex Direction, Flex Wrap, Justify Content, Align Items e Align Content.

Control Info

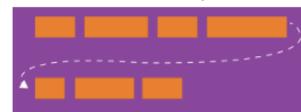
Control Type	Flex Grid
Auto Grow	False
Flex Direction	Row
Flex Wrap	No Wrap
Justify Content	Flex Start
Align Items	Stretch
Align Content	Stretch

Flex Direction



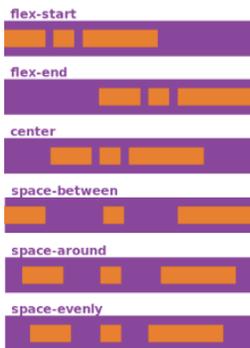
Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Flex Wrap



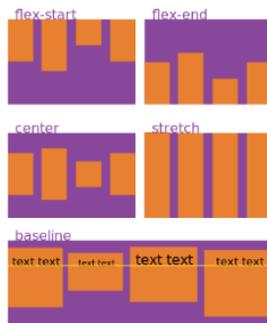
Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Justify Content



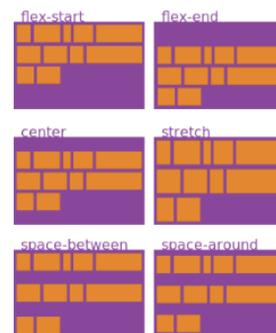
Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Align Items



Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Align Content



Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Flex Direction: Especifica a direção dos elementos, que podem ser colocados horizontalmente (Rows ou Inverse Rows) ou verticalmente (Column ou Inverse Column)

Flex Wrap: Define como serão organizados os elementos quando ocupam mais do que o comprimento de uma linha e não podem ser exibidos todos. O valor Wrap os acomoda na próxima linha.

Justify Content: Define o alinhamento dos elementos nas linhas (alinhamento horizontal se Flex Direction for especificada como Row e alinhamento vertical se Flex Direction for especificada como Column). Se o alinhamento for horizontal, permite estabelecer como será alinhado o conteúdo, se da esquerda para a direita, da direita para a esquerda, centralizado, que seja ajustada a separação para ocupar toda a largura disponível ou se são adicionados espaços entre os elementos.

Align Items: Define o alinhamento dos elementos nas linhas (alinhamento vertical se Flex Direction for especificada como uma linha e alinhamento horizontal se Flex Direction for especificada como uma coluna)

Align Content: Nos permite alinhar a linha quando há espaço adicional no container.

Estas propriedades também se aplicam ao Flex Control, que é um container flexível que veremos em um momento.

Exemplo de uso do Flex grid

FlexGrid
Flex Direction = Row
Flex Wrap = Wrap



Control Info	
Control Type	Flex Grid
Auto Grow	True
Flex Direction	Row
Flex Wrap	Wrap
Justify Content	Flex Start
Align Items	Stretch
Align Content	Stretch

Layout | Rules | Events | Conditions | Variables | Documentation

Application Bar

MainTable | Grid1

Country: &CountryId

Name From: &AttractionNameFrom

Name To: &AttractionNameTo

GRID

AttractionId

CountryId



AttractionName

CityName

CountryName

Trips: &Trips

Total Trips: &TotalTrips

Suponhamos que queremos ver uma galeria de fotos das atrações, com os dados de cada atração abaixo da foto. Queremos que as fotos fluam horizontalmente e que automaticamente passem para uma nova linha para evitar a rolagem horizontal.

Como vimos anteriormente, algumas fotos estão no formato paisagem e outras no formato retrato, portanto suas dimensões são desconhecidas no momento do design.

Para conseguir isto, usaremos o Flex Grid, com a propriedade Flex Direction definida como "Row" e Flex Wrap definido como "Wrap". Deixamos o restante das propriedades padrão e vamos executar.

Exemplo de uso do Flex Grid

The screenshot shows a web application interface for a travel agency. The page title is "View Attractions_MoreInfo_Level_Detail". The header features the "TRAVEL AGENCY" logo. Below the header, there are form fields for "Country" (set to "(None)"), "Name From", and "Name To". A table of attractions is displayed, with a "Total Trips" of 20. The interface is overlaid with a control info panel for a "Table: Grid1Table".

Table: Grid1Table	
Control Name	Grid1Table
Layout Name	
Appearance	
Columns Style	400dip
Rows Style	pd;pd;400dip;pd;pd;pd;pd

Control Info	
Control Type	Flex Grid
Auto Grow	True
Flex Direction	Row
Flex Wrap	Wrap
Justify Content	Space Around
Align Items	Stretch
Align Content	Stretch

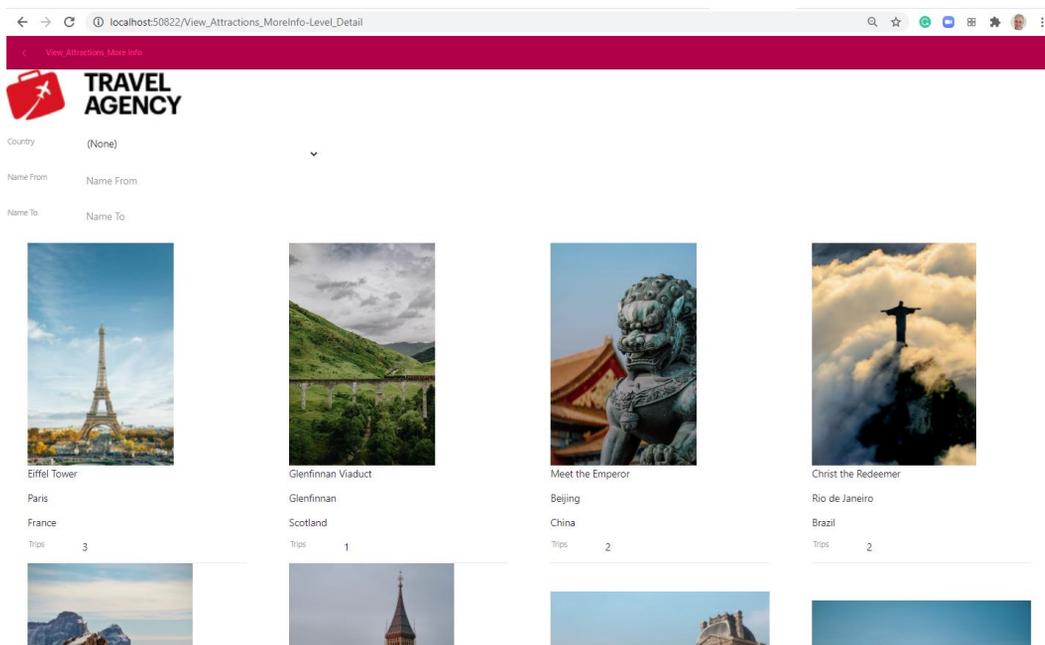
Vemos que tentou fazer um wrap, mas como havia espaço suficiente, não o fez. Também vemos que não foi deixado espaço suficiente para ver os dados das atrações claramente.

Para corrigir isto, vamos dar um valor de 400 dips para as colunas da tabela do grid, a Grid1Table, de forma que ocupe mais espaço e o Flex Grid deva fazer o wrap.

Também alteramos a propriedade Justify Content do Flex grid para o valor Space Around para que adicione espaços entre as atrações e não fiquem coladas umas nas outras.

Executamos.

Exemplo de uso do Flex Grid



Vemos que agora cada coluna ocupa mais espaço e é feito o wrap, mostrando as atrações abaixo automaticamente assim que é ocupado o espaço disponível na linha.

Também vemos que a separação entre atrações é a adequada e que as informações são bem apresentadas.

Suponhamos que quiséssemos que a informação do nome da atração, cidade, país e total da viagens apareçam à direita da foto e não abaixo.

Se quisermos alterar o layout de cada item do grid, devemos usar um Flex Control.

Flex Control

Flex: Table1

Control Name	Table1
Appearance	
Class	Table
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
Scroll Behavior	
Scroll Factor	1
Zoom Factor	0
Scroll Attachment	Parent
Layout Behavior	
Expand Bounds	Background Only
Expand Bounds Direction	Top, Left, Bottom, Right
Flex Direction	Column
Flex Wrap	No Wrap
Justify Content	Flex Start
Align Items	Stretch
Adjust Container Size	False

Control Info

Control Type	Flex Grid
Auto Grow	True
Flex Direction	Row
Flex Wrap	Wrap
Justify Content	Space Around
Align Items	Stretch
Align Content	Stretch

Table: Grid1Table

Control Name	Grid1Table
Layout Name	
Appearance	
Columns Style	400dip;100%
Rows Style	400dip
Width	100%
Height	400dip
Auto Grow	True

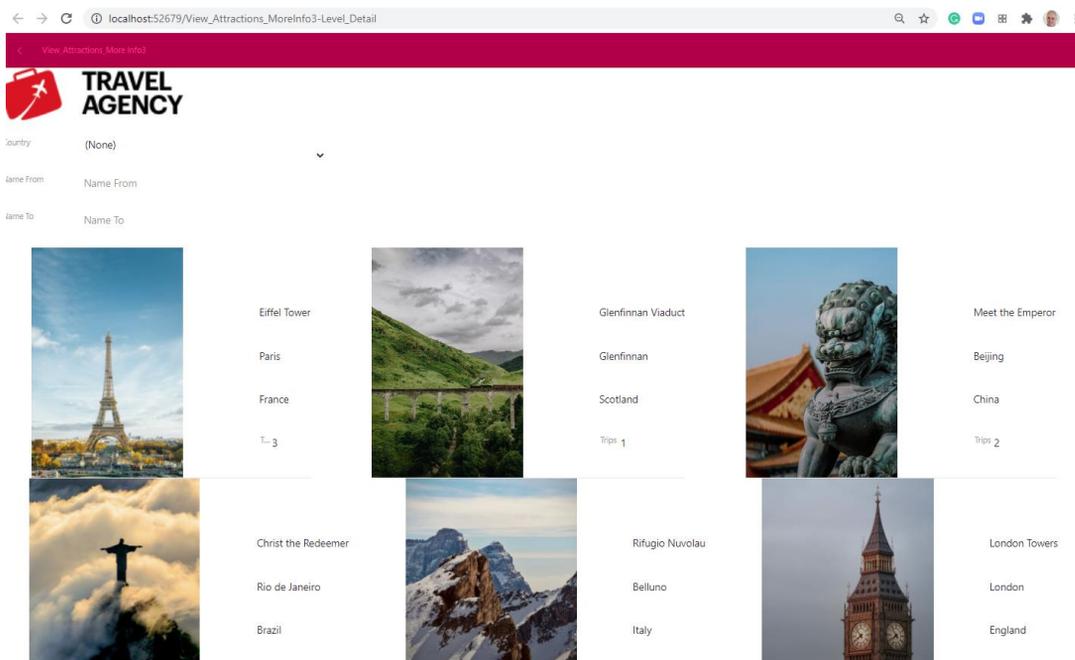
O Flex Control (também chamado de Flex Table) é um container de controles, como as Tabelas ou as Tabelas Responsivas. Seu objetivo é oferecer novas maneiras de organizar os controles na interface de usuário, mesmo que sejam desconhecidos os tamanhos dos controles contidos.

Para continuar com o exemplo que propusemos, salvamos o panel View_Attractions_MoreInfo2 com o nome View_Attractions_MoreInfo3. E arrastamos um Flex Control dentro do grid, à direita dos outros controles. Vemos que fica com o nome Table1 e se formos às suas propriedades, reconhecemos as mesmas propriedades que tínhamos para o Flex Grid.

Definimos a propriedade Flex Direction para Column e movemos todos os dados da atração, exceto a foto, para dentro da Flex Table. O Flex Grid ainda possui as mesmas propriedades do exemplo anterior, e para a tabela do grid, alteramos o valor da linha de pd para 400 dips.

E executamos.

Exemplo de uso do Flex Control



Agora vemos as informações da atração à direita. Então se quisermos mais tarde podemos ajustar os valores das propriedades do Flex Control para melhorar a forma como vemos os dados.

Neste vídeo vimos como podemos utilizar diferentes tipos de grids e configurá-los para visualizar as informações de diversas formas.

Nos vídeos a seguir, continuaremos melhorando a experiência de usuário de nossa aplicação.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications