

# FIAP

## RESPONSIVE WEB DEVELOPMENT



# GRID

Prof. Alexandre Carlos

[profalexandre.jesus@fiap.com.br](mailto:profalexandre.jesus@fiap.com.br)

Prof. Luís Carlos

[lsilva@fiap.com.br](mailto:lsilva@fiap.com.br)

Prof. Wellington Cidade

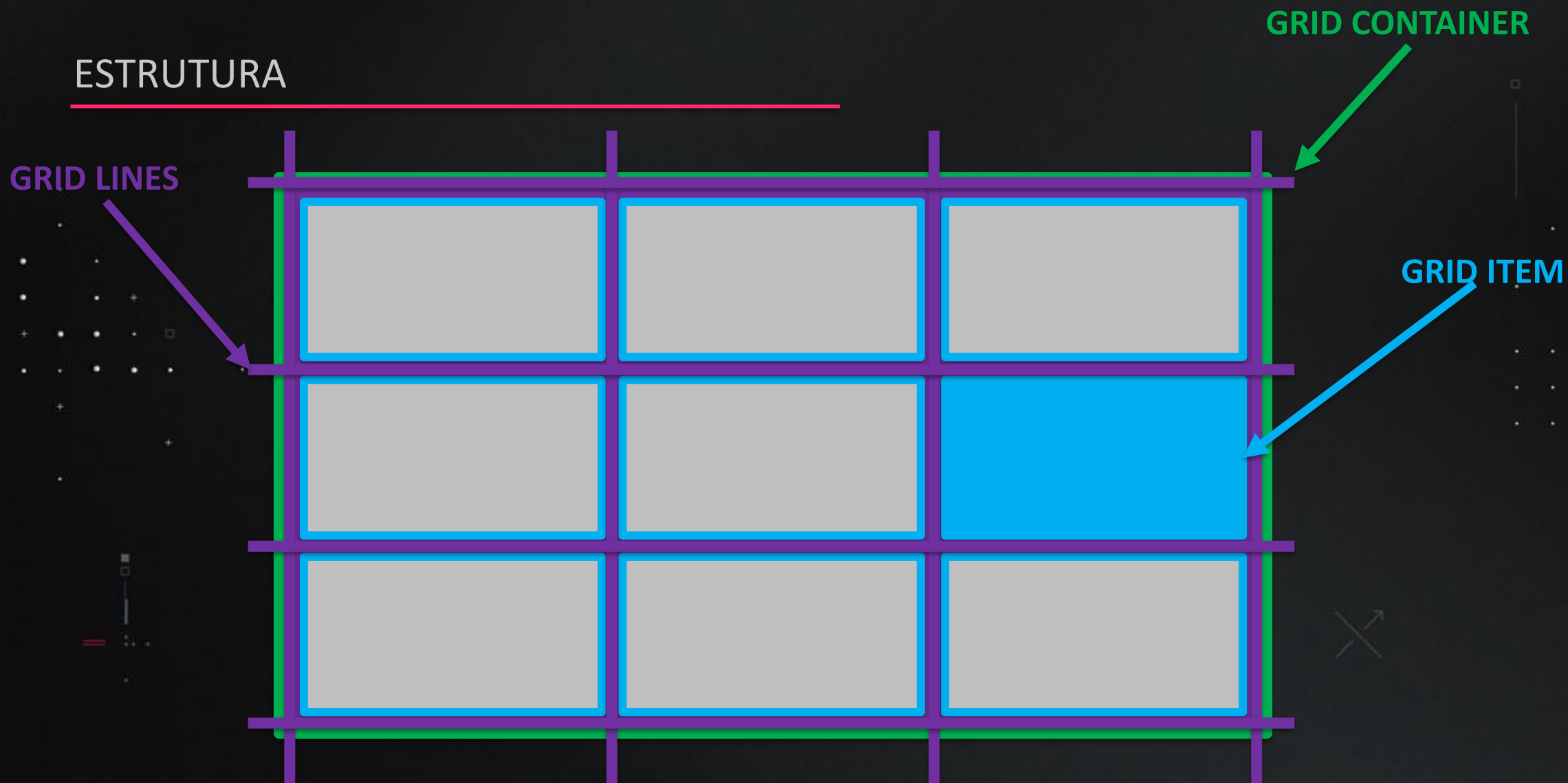
[profwellington.tenorio@fiap.com.br](mailto:profwellington.tenorio@fiap.com.br)



# INTRODUÇÃO

FIAP

## ESTRUTURA

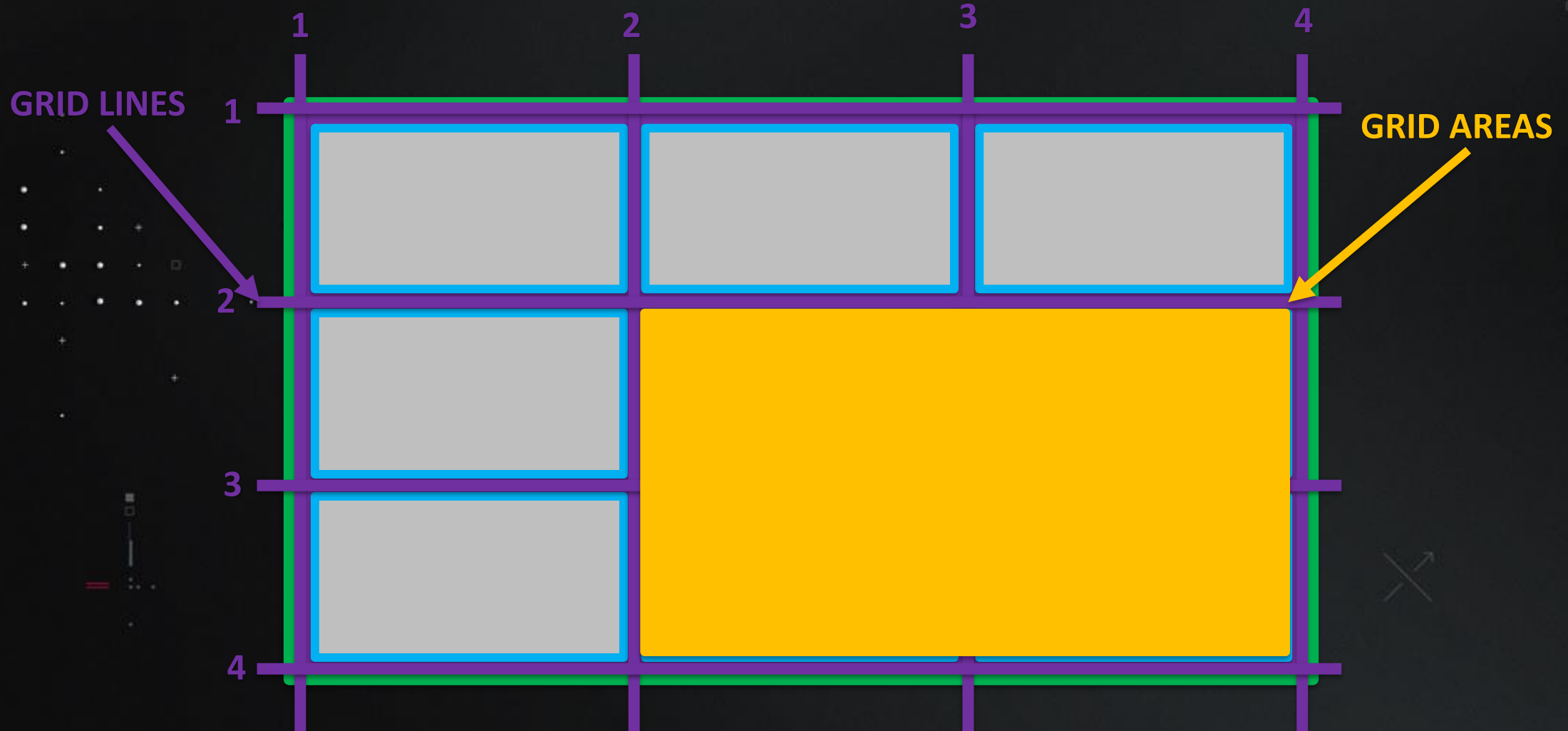




# INTRODUÇÃO

FIAP

## ESTRUTURA

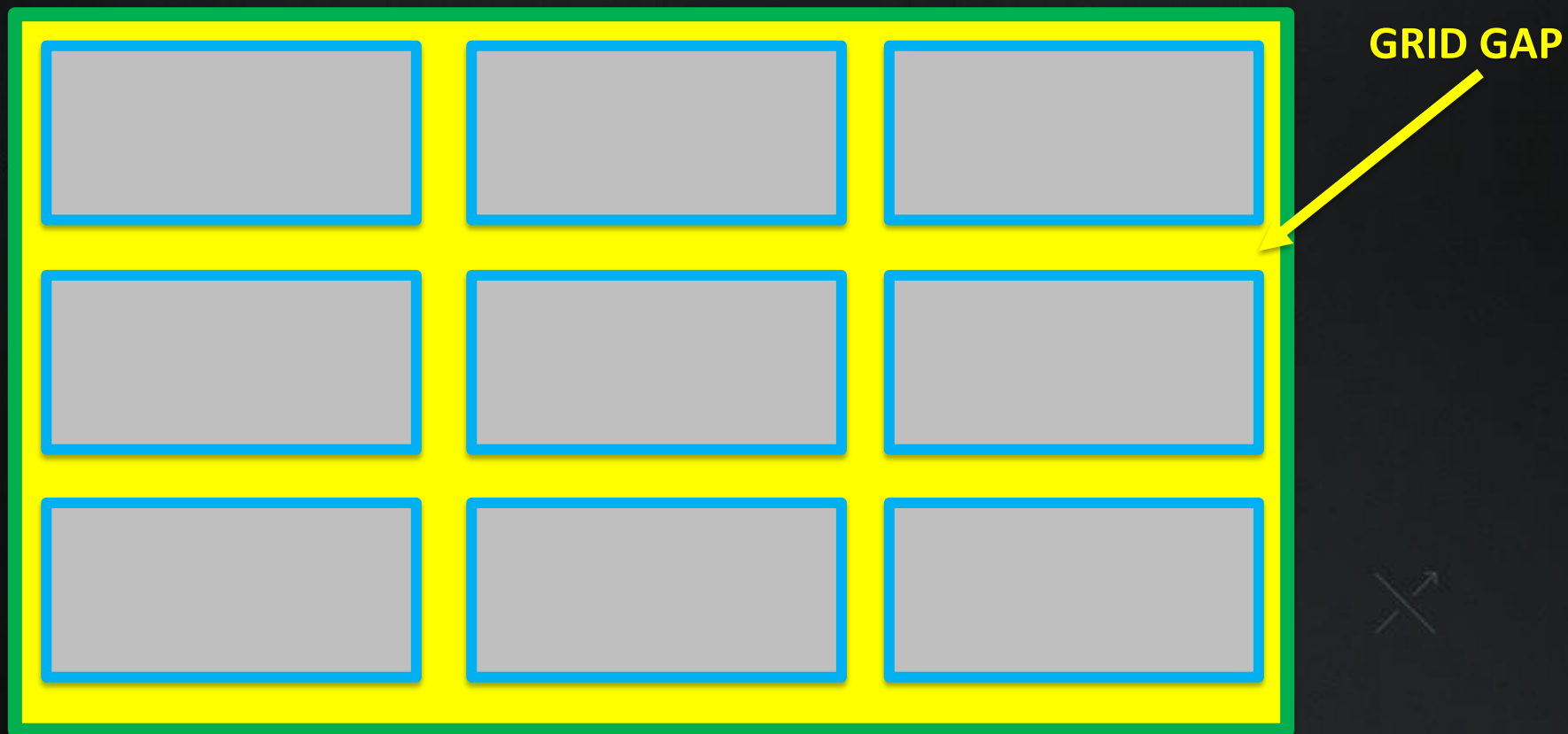




# INTRODUÇÃO

FIAP

## ESTRUTURA





# ELEMENTOS DA GRID



**Display: grid** - Devemos aplicar o display grid ao container que vai conter os elementos de nossa página.

```
.container{  
  display: grid;  
}
```

**Grid-template-columns** - É usado para especificar quantas colunas queremos em nossa grid e a largura de cada coluna

```
.container{  
  display: grid;  
  grid-template-columns: 300px 300px 300px;  
}
```

**Columns-gap** — Cria um espaçamento entre as colunas da nossa grid

```
.container{  
  display: grid;  
  grid-template-columns: 300px 300px 300px;  
  column-gap: 5px;  
}
```





# ELEMENTOS DA GRID

FIAP

**Grid-template-rows** - É usado para especificar quantas linhas queremos em nossa grid e a largura de cada linha

```
.container{  
  display: grid;  
  grid-template-rows: 150px 150px 150px;
```

**Grid-auto-rows** - É usado para especificar a altura de linhas que estiverem fora da grade especificada.

```
.container{  
  display: grid;  
  grid-template-rows: 150px 150px 150px;  
  grid-auto-rows: 200px;
```

**Row-gap** – Cria um espaçamento entre as linhas da nossa grid

```
.container{  
  display: grid;  
  grid-template-rows: 150px 150px 150px;  
  grid-auto-rows: 200px;  
  row-gap: 5px;
```



# ELEMENTOS DA GRID



**Repeat** - É uma função que nos permite definir um valor e o número de vezes que queremos representar.

```
.container{  
  grid-template-columns: repeat(3, 300px);
```

• **Minmax** - É uma função que nos permite definir um valor mínimo e um máximo para cada elemento da grid

```
.container{  
  grid-template-columns: minmax(300px, 60%) repeat(2, 300px);
```

**Obs** – no caso de altura de linha, se o valor max for definido como “auto” ele se adaptará ao conteúdo.

**Unidade “fr”** – a unidade fr significa fração, assim podemos dizer para a coluna usar x frações do espaço disponível.

```
.container{  
  grid-template-columns: 1fr 100px 100px;
```



# ALINHAMENTOS

FIAP

**Align-items** – Define o alinhamento do conteúdo nas linhas em relação a sua altura.

Podemos usar os valores:

- start – alinha no início do espaço da linha;
- center – alinha ao centro do espaço da linha;
- end – alinha ao fim do espaço da linha;
- stretch – ocupa todo o espaço da linha;

```
.container{  
  display: grid;  
  grid-auto-rows: 200px;  
  align-items: start;
```

**Justify-items** – Define o alinhamento o conteúdo nas colunas em relação a sua largura.

Podemos usar os valores:

- start – alinha no início do espaço da coluna;
- center – alinha ao centro do espaço da coluna;
- end – alinha ao fim do espaço da coluna;
- stretch – ocupa todo o espaço da coluna;

```
.container{  
  display: grid;  
  justify-items: center;
```





# ALINHAMENTO INDIVIDUAL

FIAP

**Align-self** – Define o alinhamento do conteúdo de um elemento individualmente na linha em relação a sua altura.

Podemos usar os valores:

- start – alinha no início do espaço da linha;
- center – alinha ao centro do espaço da linha;
- end – alinha ao fim do espaço da linha;
- stretch – ocupa todo o espaço da linha;

```
.container :nth-child(3){  
  align-self: center;  
}
```

**Justify-self** – Define o alinhamento o conteúdo de um elemento individualmente na coluna em relação a sua largura.

Podemos usar os valores:

- start – alinha no início do espaço da coluna;
- center – alinha ao centro do espaço da coluna;
- end – alinha ao fim do espaço da coluna;
- stretch – ocupa todo o espaço da coluna;

```
.container :nth-child(3){  
  justify-self: center;  
}
```



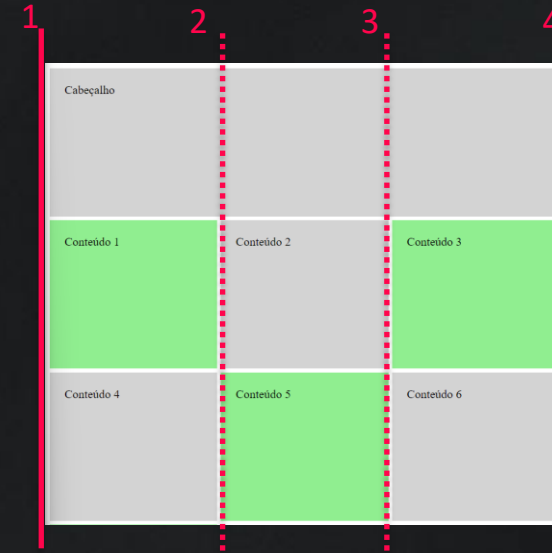
# POSICIONAMENTO INDIVIDUAL

FIAP

**Grid-column** – Define o início e o fim do posicionamento de um elemento em relação as colunas da grid.

```
.container :nth-child(1){  
  grid-column: 1 / 4;  
}
```

**Obs** – Se for definido -1 no último valor ele vai até a última coluna.



**Grid-row** – Define o início e o fim do posicionamento de um elemento em relação as linhas da grid.

```
.container :nth-child(1){  
  grid-row: 1 / 4;  
}
```





GRID AREAS

FIAP

# TRABALHANDO COM GRID AREAS



# GRID AREAS



**Grid-area** - Nomeia um elemento para que ele possa ser identificado na grid.

```
header{  
  grid-area: header;  
}
```

**Obs** - devemos fazer isso para todos os elementos.

**Grid-template-areas** – posicionado no container principal é responsável por definir o posicionamento de cada item da grid.

```
.container{  
  display: grid;  
  grid-auto-rows: 150px;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-template-areas:  
    'header header header'  
    'section1 section2 section3'  
    'section4 section5 section6'  
    'footer footer footer'  
  ;  
  column-gap: 5px;  
  row-gap: 5px;  
}
```





# GRID AREAS

FIAP

**Auto-fill** - repete um valor automaticamente até ocupar todo o espaço disponível.

```
.container{  
  display: grid;  
  grid-auto-rows: 150px;  
  grid-template-columns: repeat(auto-fill, 100px);  
}
```

**Auto-fit** – repete o valor de largura automaticamente somente o número de vezes necessárias para atender o grid-template-areas.

```
.container{  
  display: grid;  
  grid-auto-rows: 150px;  
  grid-template-columns: repeat(auto-fit, 100px);  
}
```



# OBRIGADO

## FIAP

Copyright © 2020 | Professor Titulares

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.

