02

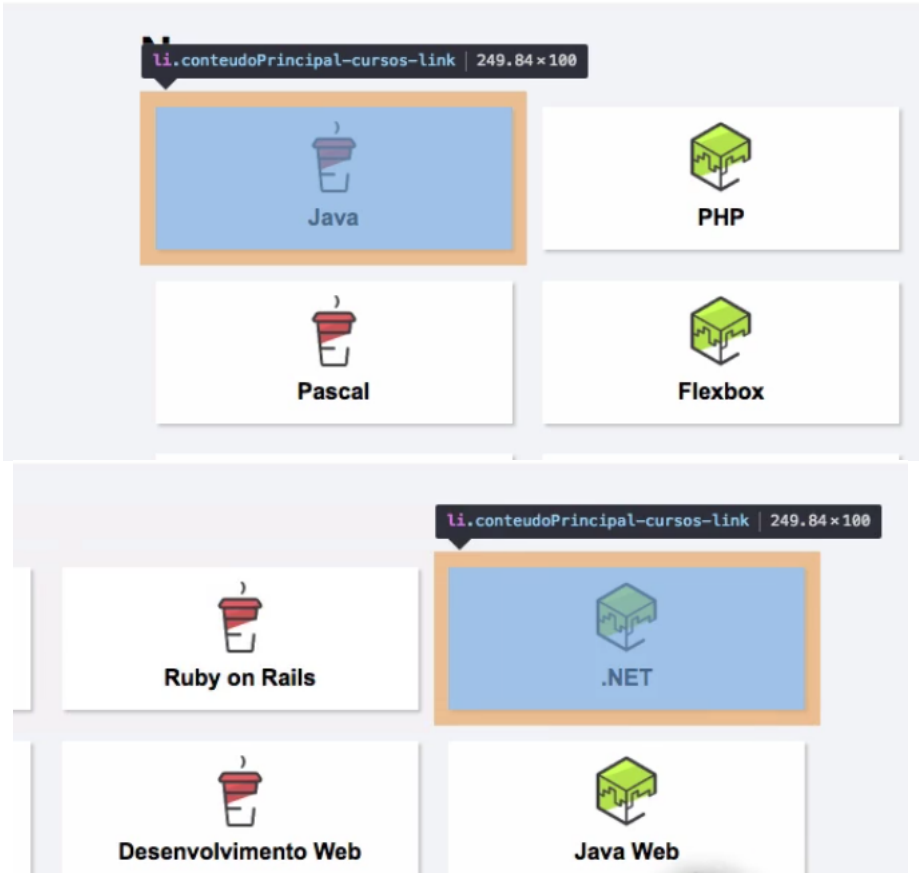
Terminando course grid

Transcrição

Na última aula não foi possível utilizar o recurso `flexbox` para reposicionar os elementos. Para solucionar este problema é preciso fazer contas! Nós tínhamos utilizado o `margin-bottom`, então, ao em vez de utilizar a margem apenas embaixo dos elementos vamos inserir também no entorno dos itens. Por isso, escrevemos apenas `margin: 1%` e teremos o seguinte código:

```
.conteudoPrincipal-cursos-link {
    width: 23%;
}
```

Repare que os elementos da primeira coluna possuem uma margem a esquerda e os objetos da última uma margem a direita:

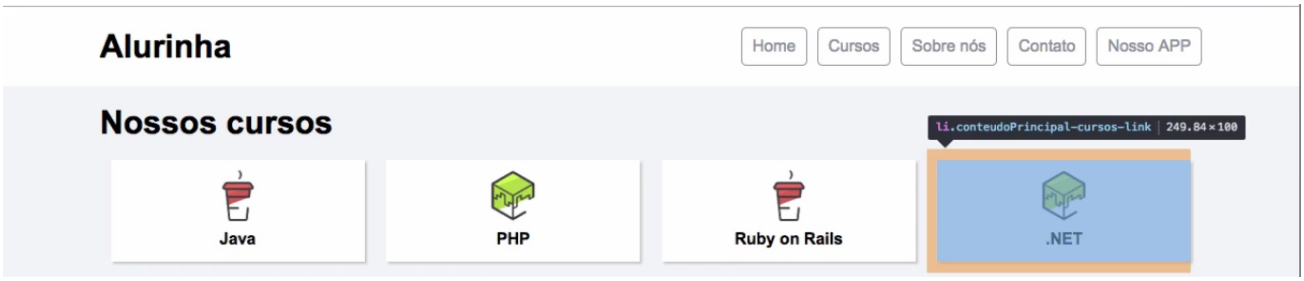


Porém, para manter as proporções tais margens não fazem sentido, elas atrapalham mais do que ajudam! Então, no caso dos elementos da última coluna, estamos lidando com objetos nas posições 4, 8, 12 e 16, ou seja, itens que são múltiplos de 4. Portanto, no arquivo `flexbox.css` vamos adicionar o `.conteudoPrincipal-cursos-link` e acrescentar o `nth-child()`. Como queremos pegar o "quarto filho" adicionamos entre os parênteses o `4n`. O código fica da seguinte maneira:

```
.conteudoPrincipal-cursos-link:nth-child(4n) {
    margin-left: 0;
```

```
}
```

Observando a página vemos que de fato os itens tiveram a margem zerada:

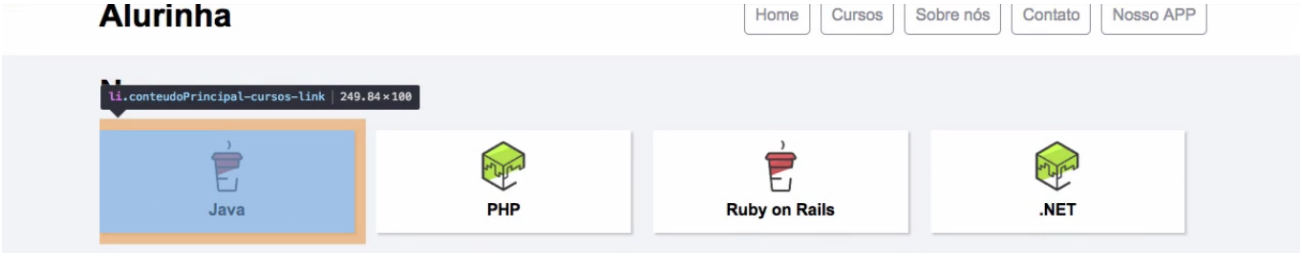


Falta retirar ainda a margem esquerda dos elementos da primeira coluna. Estes objetos estão na sequência dos objetos múltiplos de 4, isto é, depois do quarto vem o quinto, depois do oitavo vem o nono, depois do décimo segundo o décimo terceiro e assim por diante. Portanto, tais objetos equivalem a 4+1, então, vamos escrever `nth-child(4n+1)` e colocar também a `margin-left: 0` :

```
.conteudoPrincipal-cursos-link:nth-child(4n+1) {
    margin-left: 0;

}
```

Dando um `reload` vemos que todos os objetos foram arrumados inclusive o primeiro:



Isso acontece pois o `n` do primeiro elemento equivale a `0` , então, `4x0 + 1 = 1` e por isso ele é ajustado! No segundo item o `n =1` , assim, `4x1 + 1 = 5` , no terceiro ícone `n = 2` , `4x2 + 1 = 9` e assim por diante.

Utilizando o `nth` , que é relativamente fácil de usar, solucionamos a questão da distribuição! O que fizemos foi um `gride` de caixas organizadamente posicionadas. O `flex` é capaz de fazer `gride` , ainda assim, não existe uma maneira fácil de espaçar os elementos. Foi preciso utilizar não apenas o `nth` mas também fazer contas com a largura e a margem para resolver o problema!