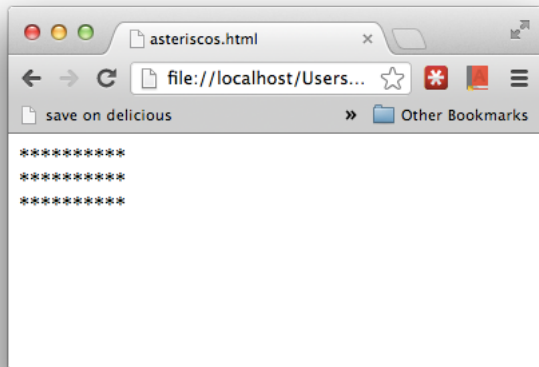


Repetições aninhadas

Transcrição

Repare que os `loops` são frequentes em nosso código e também que é comum que eles apareçam inseridos uns dentro dos outros. Imagine que queremos mostrar a sequência de 3 linhas com 10 asteriscos cada:



Você pode pensar em uma solução bem fácil, mas nada desafiadora. Algo como, por exemplo:

```
mostra("*****");  
mostra("*****");  
mostra("*****");
```

Mas, pode ser mais interessante. Ao em vez de fazer um "copia e cola", podemos usar um `for` :

```
for(var linha = 0; linha < 3; linha = linha + 1) {  
    mostra("*****");  
}
```

Melhorou! Mas, se precisássemos alterar o código e passar a imprimir em cada linha 250 asteriscos? Teríamos que reescrever o código adicionando diversos asteriscos dentro da chamada `mostra` .

A ideia é utilizar dois `loops` , o primeiro deve comunicar algo como: para cada linha, faça... . E o segundo deve falar para cada coluna, imprima um `*` . Ao final de cada linha vamos imprimir também um `
` . O código será curto, mas a primeira vista ele pode assustar. Crie o `asteriscos.html` e digite as seguintes linhas, dentro das tags de `script` :

```
for(var linha = 0; linha < 3; linha = linha + 1) {  
    for(var coluna = 0; coluna < 10; coluna = coluna + 1) {  
        document.write("*");  
    }  
}
```

```
document.write("<br/>");  
}
```

Repare que declaramos duas variáveis! A variável `linha` que corresponde ao primeiro `loop` e a variável `coluna` para o segundo `loop`. Cada vez que o `loop` das colunas termina, ele pula a linha e começa a execução da próxima iteração do `loop` das colunas! Exatamente como queríamos! Nesse caso, dizemos que temos um `loop` encadeando outro `loop`. Um termo mais técnico para dizer isso é que temos "loops **aninhados**"!