

Programação assíncrona - *Callbacks*

O que é programação assíncrona?

Os códigos e funções assíncronas no JavaScript, quando são executados, não bloqueiam a execução das linhas de código seguintes. Uma função assíncrona é executada em segundo plano enquanto o programa continua interpretando as linhas de código seguintes.

Um exemplo de uma função assíncrona no JavaScript é a função `setTimeout()`. Com essa função nós podemos programar para que um determinado código seja executado depois que o tempo em milisegundos, passados como parâmetro, se esgotar. Por exemplo:

```
setTimeout(() => {  
  console.log('Será executado após 2 segundos');  
}, 2000);
```

Note que o primeiro parâmetro da função `setTimeout` é uma outra função que possui um `console.log` internamente. O segundo parâmetro é o tempo em milissegundos que deverá ser aguardado até que a função do primeiro parâmetro seja executada.

Funções assíncronas são amplamente utilizadas em sistemas web. Isso porque muitos dos dados que são necessários nesses sistemas geralmente estão em um servidor, e o acesso a esse servidor se dá pela internet. Como o tempo referente ao acesso ao servidor, ao processamento e ao retorno de dados pode demorar muito, usa-se a programação assíncrona para que o sistema não fique bloqueado enquanto esse processo está sendo feito.

Obviamente nós não vamos implementar um servidor para fazermos todos os processos. Para a nossa aula, nós vamos utilizar a função `setTimeout` para simular um tempo de resposta de um processo envolvendo um servidor na internet.

Callbacks

Uma função *callback* no JavaScript é uma função que é passada como parâmetro para outra função, que pode chamar a função *callback* para completar algum tipo de rotina ou ação.

Exemplo:

```
function obterNome(callback) {  
    var nome = prompt('Entre com o seu nome: ');  
    callback(nome);  
}  
  
obterNome((nome) => {  
    console.log(nome);  
}));
```

No exemplo acima, a função `obterNome` recebe uma outra função *callback* como parâmetro. Essa função *callback* é chamada após o usuário entrar com o nome dele, passando a variável `nome` como parâmetro.