

JavaScript:

Consumindo e tratando dados de uma API



alura

Como funciona o JavaScript?

Aula 1



alura

**Nesse curso, seremos
desenvolvedores Front-end
aqui da Alura, especificamente
trabalharemos com Javascript.**



Importando arquivos .js

Aula 1.2



alura

O primeiro passo é criar e importar o arquivo javascript.

Vamos abrir e testar pelo navegador através da ferramenta do desenvolvedor (f12).

Usaremos a página de cadastro do site Alura Books.

The screenshot shows the 'Cadastro' (Registration) page of the AluraBooks website. The page has a dark blue header with the AluraBooks logo and navigation links: CATEGORIAS, FAVORITOS, MINHA ESTANTE, and icons for a shopping cart and user profile. Below the header, the title 'Cadastro' is centered in a dark blue box. The main content area is titled 'Dados do cadastro' (Registration Data) in orange. It is divided into two sections: 'Dados pessoais' (Personal Data) and 'Endereço' (Address). The 'Dados pessoais' section includes fields for 'Nome completo' (Full Name) with the value 'Maria Pereira da Silva', 'Nascimento' (Birth) with a date picker set to '16/06/1991', 'Gênero' (Gender) with a dropdown menu set to 'Feminino', 'E-mail' with the value 'emaildemaria@gmail.com', 'CPF' with the value '123.456.789-10', and 'Contato' (Contact) with a phone number '(11) 98765-4321'. The 'Endereço' section includes fields for 'CEP' (Postal Code) with the value '12345-678', 'Endereço' (Address) with the value 'Avenida da Alameda Silva Souza', 'Número' (Number) with the value '958', 'Complemento' (Complement) with the value 'Apto 45', 'Bairro' (Neighborhood) with the value 'Vila do Jardim', 'Cidade' (City) with the value 'São Paulo', 'UF' (State) with a dropdown menu set to 'SP', and 'País' (Country) with a dropdown menu set to 'Brasil'. At the bottom right of the form is a large orange button labeled 'Finalizar cadastro' (Finish registration).

Síncrono e assíncrono?

Aula 1.3



alura

Síncrono ou assíncrono

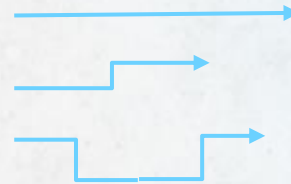
- Tendência de trabalho home office
- Comunicação pode ser por videochamada (síncrona) ou via chat (assíncrona)



Síncrono ou assíncrono



Videochamadas



Mensagens de texto

**Quais são os caminhos que você pode seguir enquanto
aguarda a resposta de um colega?**



Faltar seus
compromissos
pessoais



ou



Se ocupar em
outras tarefas até
obter resposta



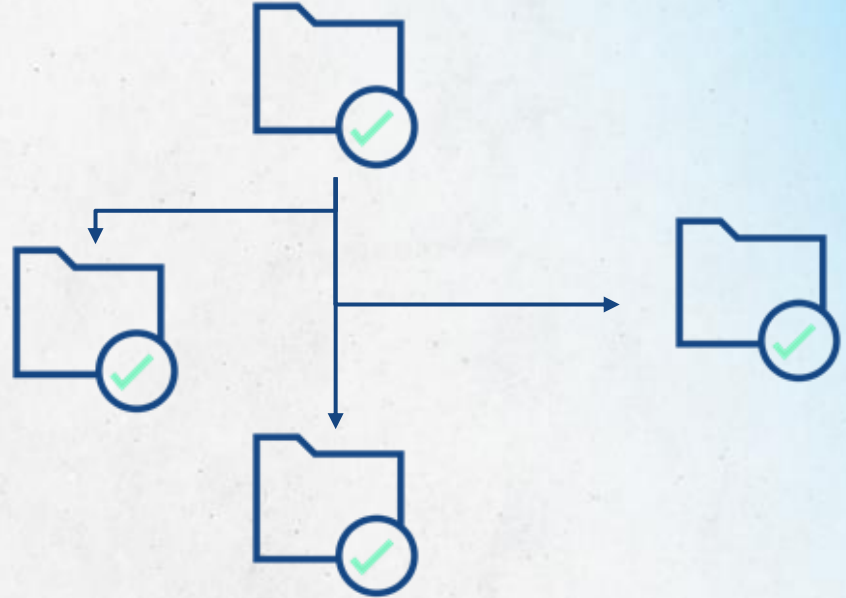
Isso é equivalente as decisões que podemos tomar em um sistema:

Síncrono



Uma tarefa é concluída após a outra (padrão do JavaScript)

Assíncrono



Há tarefas executadas em "segundo plano"

Event Loop e Call Stack

Aula 1.4



alura

Task Queue

Call Stack

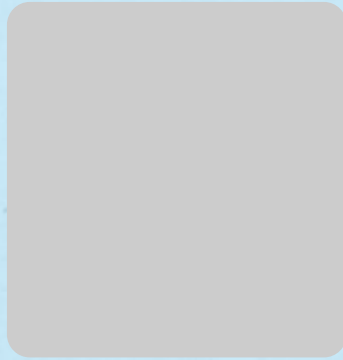


Event Loop

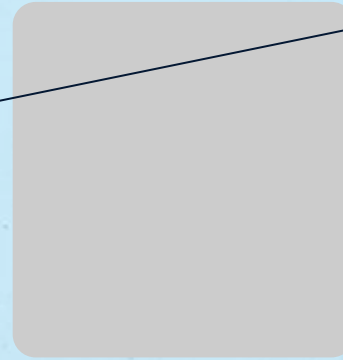
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Podemos
destrinchar o fluxo
de execução de
tarefas em
JavaScript em três
partes: **Event Loop**,
Call Stack e **Task
Queue**.

Task Queue



Call Stack

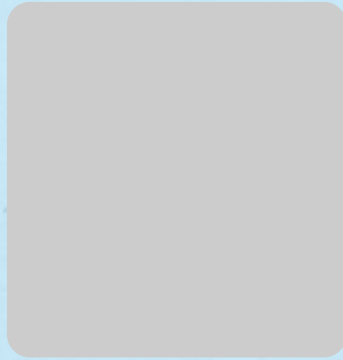


Event Loop

O Event Loop é um ciclo que monitora e executa as ações que mandamos para o JavaScript. O processo de leitura do código só é finalizado quando não existem mais ações a serem executadas.

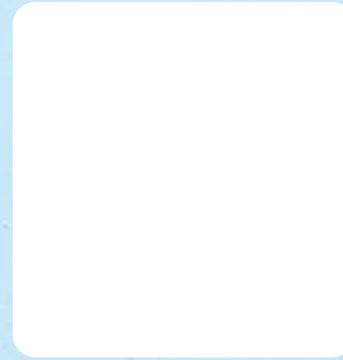
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

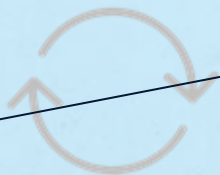
Call Stack



A pilha de chamadas (call stack) é um mecanismo que organiza como irá funcionar o script quando existem muitas funções: qual função está sendo executada, quais estão sendo chamadas dentro de alguma função, etc.

```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```


Task Queue



Event Loop

Call Stack

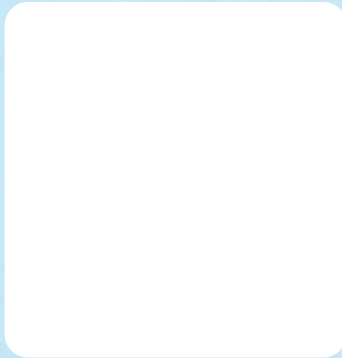


A task queue é a fila de tarefas assíncronas. Se algo precisa ocorrer em segundo plano ou mais tarde, é nessa fila que ele será adicionado e executado mais tarde.

```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

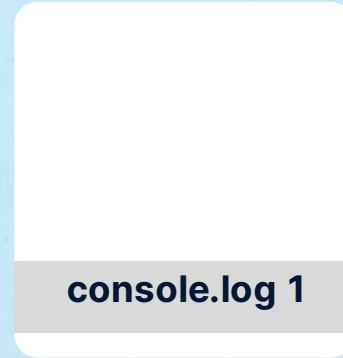

Exemplo síncrono

Task Queue



Event Loop

Call Stack

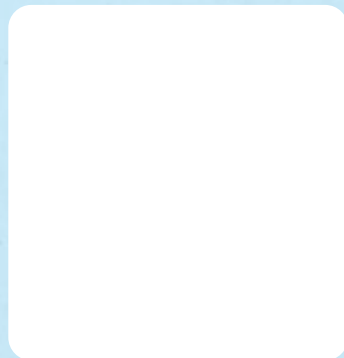


```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



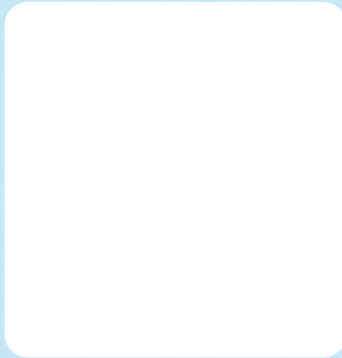
Call Stack



Event Loop

```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



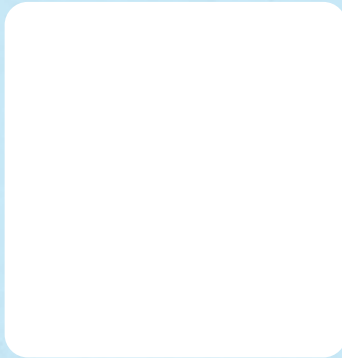
Event Loop

Call Stack



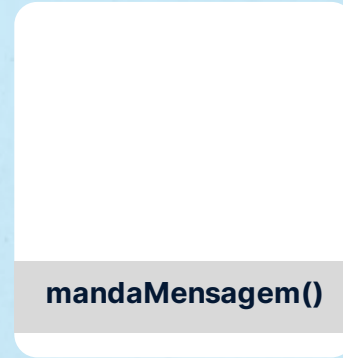
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



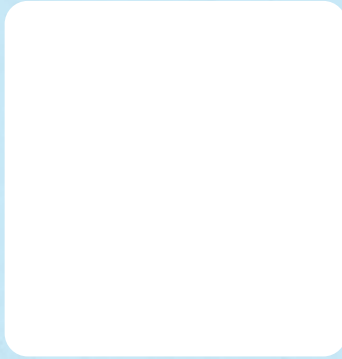
Event Loop

Call Stack



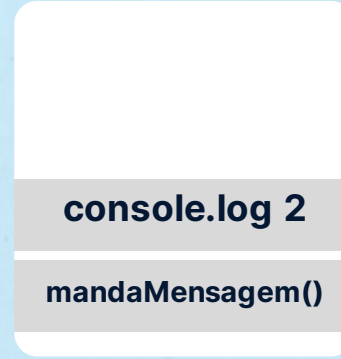
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



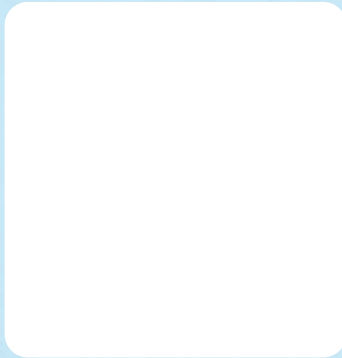
Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```


Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```


Task Queue



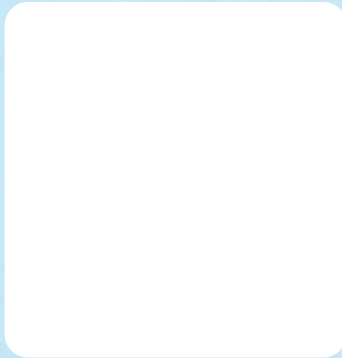
Event Loop

Call Stack



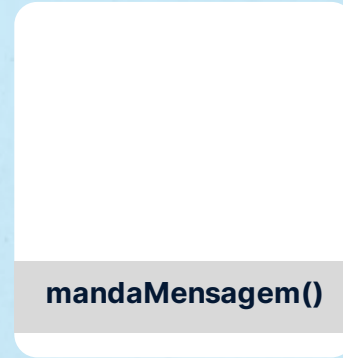
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



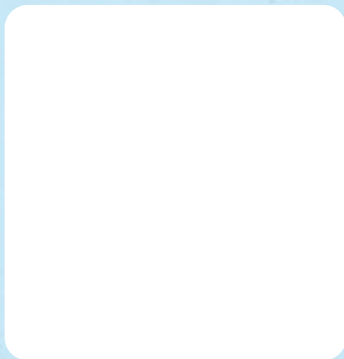
Event Loop

Call Stack



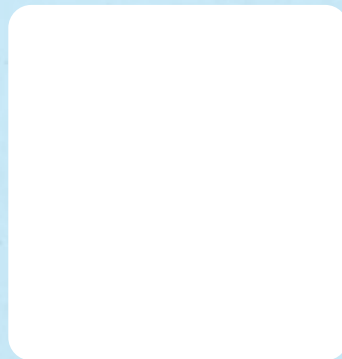
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



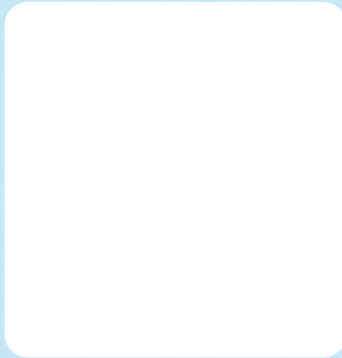
Event Loop

Call Stack



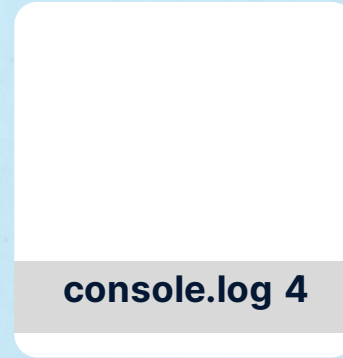
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



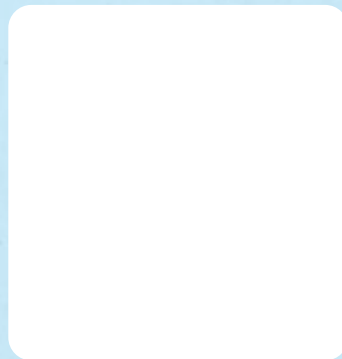
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
mandaMensagem();  
console.log("Tchau tchau!")
```

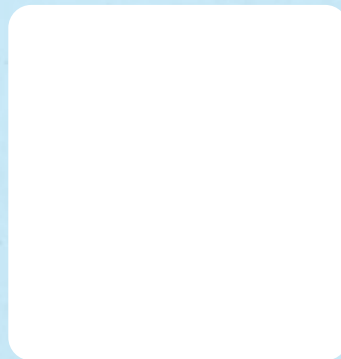
Exemplo assíncrono

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```


Task Queue



Event Loop

Call Stack



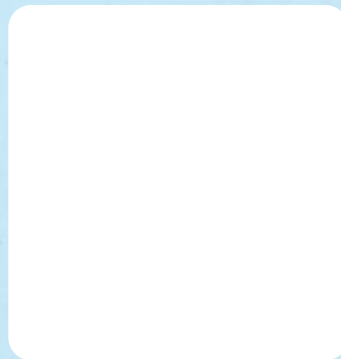
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```


Task Queue



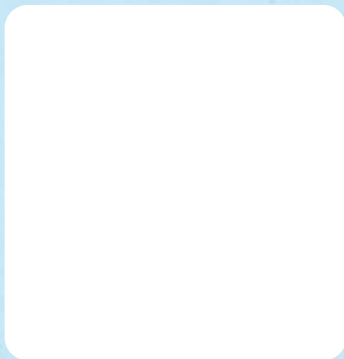
Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



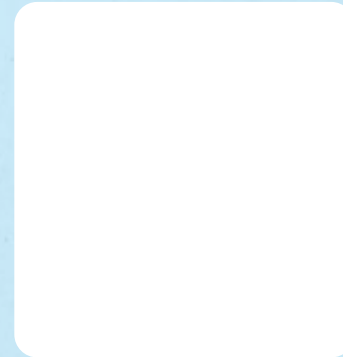
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



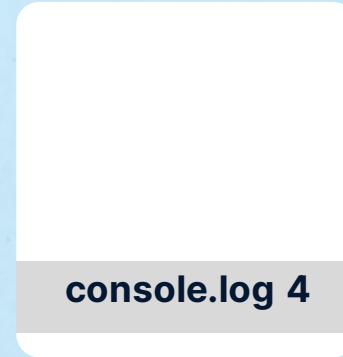
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



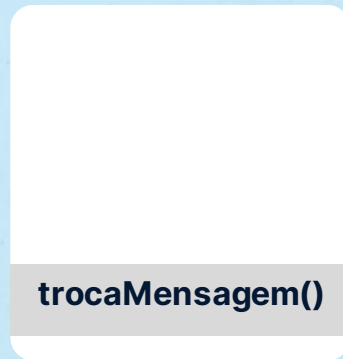
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



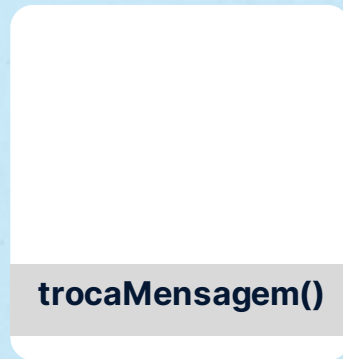
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```


Task Queue



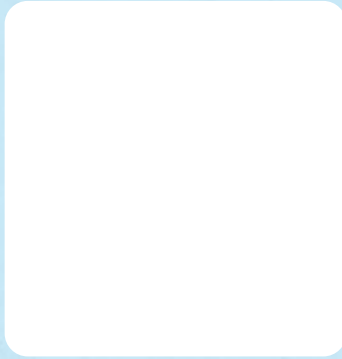
Event Loop

Call Stack



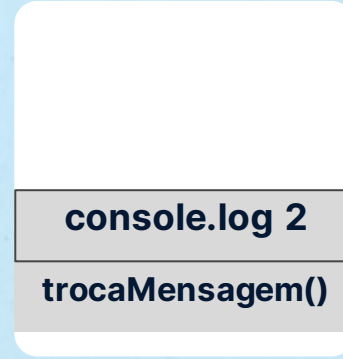
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```


Task Queue



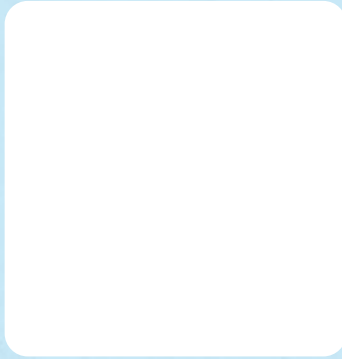
Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



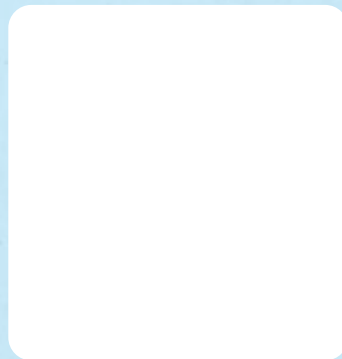
```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");  
function mandaMensagem() {  
  console.log("Tudo bem?");  
  console.log("Vou te mandar uma solicitação!");  
  console.log("Solicitação recebida!");  
}  
setTimeout(mandaMensagem, 5000);  
console.log("Tchau tchau!")
```