

Funções JS

Definições de função

Parâmetros de Função

Invocação de Função

Chamada de Função

Função Aplicar

Fechamentos de funções

Classes JS

Introdução à aula

Herança de classe

Class Static

JS Async

JS Callbacks

JS Assíncrono

JS Promises

JS Async / Await

Versões JS

Callbacks de JavaScript

< Anterior

Próximo >

"Eu vou ligar de volta mais tarde!"

Um retorno de chamada é uma função passada como um argumento para outra função

Esta técnica permite que uma função chame outra função

Uma função de retorno de chamada pode ser executada após a conclusão de outra função

Sequência de funções

As funções JavaScript são executadas na sequência em que são chamadas. Não na sequência em que são definidos.

Este exemplo acabará exibindo "Adeus":

Exemplo

```
function myFirst() {
  myDisplayer("Hello");
}

function mySecond() {
  myDisplayer("Goodbye");
}

myFirst();
mySecond();
```

Tente você mesmo "

Este exemplo exibirá "Olá":

Exemplo

```
function myFirst() {
  myDisplayer("Hello");
}

function mySecond() {
  myDisplayer("Goodbye");
}

mySecond();
myFirst();
```

Tente você mesmo "

Controle de Sequência

Às vezes, você gostaria de ter melhor controle sobre quando executar uma função.

Suponha que você queira fazer um cálculo e, em seguida, exibir o resultado.

Você pode chamar uma função de calculadora (`myCalculator`), salvar o resultado e, em seguida, chamar outra função (`myDisplayer`) para exibir o resultado:

Exemplo

```
function myDisplayer(some) {
  document.getElementById("demo").innerHTML = some;
}

function myCalculator(num1, num2) {
  let sum = num1 + num2;
  return sum;
}

let result = myCalculator(5, 5);
myDisplayer(result);
```

Tente você mesmo "

Ou você pode chamar uma função de calculadora (`myCalculator`) e deixar que a função de calculadora chame a função de exibição (`myDisplayer`):

Exemplo

```
function myDisplayer(some) {
  document.getElementById("demo").innerHTML = some;
}

function myCalculator(num1, num2) {
  let sum = num1 + num2;
  myDisplayer(sum);
}

myCalculator(5, 5);
```

Tente você mesmo "

O problema com o primeiro exemplo acima é que você precisa chamar duas funções para exibir o resultado.

O problema com o segundo exemplo é que você não pode impedir que a função de calculadora exiba o resultado.

Agora é hora de fazer um retorno de chamada.

Callbacks de JavaScript

Um retorno de chamada é uma função passada como um argumento para outra função.

Usando um retorno de chamada, você pode chamar a função de calculadora (`myCalculator`) com um retorno de chamada e deixar a função de calculadora executar o retorno de chamada após o cálculo ser concluído:

Exemplo

```
function myDisplayer(some) {
  document.getElementById("demo").innerHTML = some;
}

function myCalculator(num1, num2, myCallback) {
  let sum = num1 + num2;
  myCallback(sum);
}

myCalculator(5, 5, myDisplayer);
```

Tente você mesmo "

No exemplo acima, `myDisplayer` é o nome de uma função.

Isso é passado `myCalculator()` como um argumento.

Ao passar uma função como argumento, lembre-se de não usar parênteses.

À direita: `myCalculator (5, 5, myDisplayer);`

Errado: `myCalculator {5, 5, myDisplayer {}};`

Quando usar um retorno de chamada?

Os exemplos acima não são muito empolgantes.

Eles são simplificados para ensinar a sintaxe de retorno de chamada.

Onde os callbacks realmente brilham são nas funções assíncronas, onde uma função tem que esperar por outra função (como esperar o carregamento de um arquivo).

As funções assíncronas são abordadas no próximo capítulo.

< Anterior

Próximo >

PROPAGANDA

Formação Virtual Pós-graduação

Você pode estudar com estudantes e professores de todo o mundo. venha conhecer!

Funiber

Abrir

Reportar erro

Fórum

Cerca de

Comprar

Principais tutoriais	Referências principais	Top Examples	Web Courses
<div>Tutorial HTML Tutorial</div> <div>CSS Tutorial</div> <div>JavaScript</div> <div>Como fazer Tutorial</div> <div>SQL Tutorial</div> <div>Python Tutorial</div> <div>W3.CSS Tutorial</div> <div>Bootstrap Tutorial</div> <div>PHP Tutorial</div> <div>Java Tutorial</div> <div>C ++ Tutorial</div> <div>jQuery Tutorial</div>	<div>HTML Reference</div> <div>CSS Reference</div> <div>JavaScript Reference</div> <div>SQL Reference</div> <div>Python Reference</div> <div>W3.CSS Reference</div> <div>Bootstrap Reference</div> <div>PHP Reference</div> <div>HTML Colors</div> <div>Java Reference</div> <div>Angular Reference</div> <div>jQuery Reference</div>	<div>HTML Examples</div> <div>CSS Examples</div> <div>JavaScript Examples</div> <div>How To Examples</div> <div>SQL Examples</div> <div>Python Examples</div> <div>W3.CSS Examples</div> <div>Bootstrap Examples</div> <div>PHP Examples</div> <div>Java Examples</div> <div>XML Examples</div> <div>jQuery Examples</div>	<div>Curso de HTML</div> <div>Curso de CSS</div> <div>Curso de JavaScript</div> <div>Curso de front end</div> <div>Curso de SQL</div> <div>Curso de Python</div> <div>Curso de jQuery</div> <div>Curso de Java</div> <div>Curso de C ++</div> <div>Curso de C #</div> <div>Curso de XML</div>

Obter certificação



PROPAGANDA

NIVEA

O MÊS DA Beleza

ESTÁ DE VOLTA!

SABE MAIS

20% desconto

Sente-te linda

COLOR PICKER

COMO NÓS

Obtenha a certificação completando um curso hoje!

iniciar

JOGO DE CÓDIGOS

Jogar um jogo