

INSTITUTO INFNET ESCOLA SUPERIOR DE TECNOLOGIA



**Teste de Performance - TP7
- PB**

Engenharia de Software

ALUNO: GUILHERME DE MORAIS CHINA COSTA

E-MAIL: guilherme.mcosta@al.infnet.edu.br

Unidade: Introdução a Javascript

Programação

CURSO: PROGRAMAÇÃO > UNIDADE 1

Lição 12: Repetições

Introdução aos laços de repetição ...

Usando laços while 100%

Mais Laços While: Um Monte de B...

Laços For! Um novo tipo de Estrut...

Desafio: Folha com Linhas

Laços For Aninhados

Revisão: Repetições

Projeto: Construindo uma Casa

Computação > Programação > Introdução a

Mais Laços While: Um Monte de Balões

Google Sala de Aula

```
1 background(122, 224, 255);
2 stroke(64, 117, 207);
3 fill(196, 33, 255);
4
5
6 var x = 50;
7 while (x < 350) {
8   line(x, 120, 210, 300);
9   ellipse(x, 90, 40, 60);
10  x += 50;
11}
12 /* The Three Loop Questions:
13 1. What do I want to repeat?
14 2. What do I want to change each time?
15 3. How long should we repeat?
16 */
17 var hopper = getImage("creatures/Hopper-Jumping");
18 image(hopper, 208, 240);
```

0:00/5:30 Assistir no YouTube Derivação

[Sobre](#) [Documentação](#) [Derivados](#) [Transcrição](#)

Sophia mostra como usar um laço while para desenhar várias formas em sequência.

Programação

CURSO: PROGRAMAÇÃO > UNIDADE 1

Lição 13: Escrevendo um código limpo

Código Legível

Explicando com Comentários

Computação > Programação > Introdução a JavaScript: Desenho & Animação > Escrevendo um código limpo

© 2024 Khan Academy

[Termos de uso](#) [Política de privacidade](#) [Ajuda de cookies](#)

Explicando com Comentários

Google Sala de Aula

```
19 // Draw the ground
20 fill(255, 255, 255);
21 rect(0, 250, 400, 150);
22
23 // Calculate y according to the path of a parabola with respect to "t":
24 // y = (velocity)*t + (acceleration)*t^2/2
25 // We can tweak the velocity and acceleration constants for our fake world
26 y = -0.02*t*t + 5.0*t;
27
28 // We make the shadow lighter as we increase y
29 var grey = 0.1 * y + 200;
30 // We grow the shadow bigger as we increase y
31 var shadowSize = 0.2 * y + 50;
32 // Draw the shadow
33 fill(grey, grey, grey);
34 ellipse(200, 300, shadowSize, 10);
35
36 // Since y is a positive height, we need to flip it
37 // to look right on the inverted coordinate plane
38 var correctedY = 250 - y;
39
40 // Draw the ball
41 fill(71, 117, 255);
42 ellipse(200, correctedY, 50, 50);
43
44 // If y becomes negative, the ball has hit the ground
45 if (y < 0) {}
```

5:44/5:44 Assistir no YouTube Derivação

[Sobre](#) [Documentação](#) [Derivados](#) [Transcrição](#)

Pamela te encoraja a usar comentários para deixar seu código mais claro, legível, e fácil de entender.

Programação

CURSO: PROGRAMAÇÃO > UNIDADE 1

Lição 16: Design Orientado a Objeto

Tipos de Objetos

Métodos de Objetos

Herança de Objetos

Revisão: Modelo Orientado a Objeto...

Computação > Programação > Introdução a JavaScript: Desenho & Animação > Design Orientado a Objeto

© 2024 Khan Academy

[Termos de uso](#) [Política de privacidade](#) [Ajuda de cookies](#)

Revisão: Modelo Orientado a Objetos

Google Sala de Aula

Esta é uma revisão do que abordamos nesse tutorial sobre o modelo orientado a objetos.

Quando nós criamos programas, frequentemente descobrimos que queremos criar muitos objetos os quais todos compartilham propriedades parecidas - como muitos gatos, que têm a cor do pelo ou o tamanho levemente diferentes, ou muitos botões, com diferentes rótulos e posições. Nós queremos poder dizer "isto é o que geralmente um gato é" e então dizer "vamos criar esse gato específico e esse outro gato, e eles serão parecidos em algumas formas e diferentes em outras formas também." Nesse caso, nós queremos usar um *modelo orientado a objetos* para definir os tipos de objetos e criar novas instâncias de tais objetos.

Para definir um tipo de objeto em JavaScript, nós primeiro temos que definir uma "função construtora". Essa é a função que nós iremos usar sempre que quisermos criar uma nova instância desse tipo de objeto. Aqui está uma função construtora para o tipo de objeto `Book`:

```
var Book = function(title, author, numPages) {
  this.title = title;
  this.author = author;
  this.numPages = numPages;
};
```

Unidade: HTML/JS: tornando páginas web interativas

Programação

CURSO: PROGRAMAÇÃO > UNIDADE 6

< Lição 1: Prepare-se para tornar suas páginas web interativas >

Bem-vindo à criação de páginas we...

Revisão de HTML/CSS

Recaptação de HTML/CSS 80%

Revisão de JavaScript

Recaptação de JavaScript 92%

Computação > Programação > HTML/JS: tornando páginas web interativas > Prepare-se para tornar suas páginas web interativas

© 2024 Khan Academy

[Termos de uso](#) [Política de privacidade](#) [Aviso de cookies](#)

Bem-vindo à criação de páginas web interativas

Google Sala de Aula

Oriando páginas web interativas

Apps Script Adobe Photoshop Johnny5 ProcessingJS Command-line

Transcrição

1:20 então talvez você tenha aprendido JavaScript com o comando de linhas simples.

Programação

CURSO: PROGRAMAÇÃO > UNIDADE 6

< Lição 2: JS e DOM >

Adicionando JS a uma página web

Desafio: o script de uma tag script

DOM (Modelo de Objeto de Docu...

Entendendo o DOM 100%

Depuração de páginas web com o c...

Computação > Programação > HTML/JS: tornando páginas web interativas > JS e DOM

© 2024 Khan Academy

[Termos de uso](#) [Política de privacidade](#) [Aviso de cookies](#)

Depuração de páginas web com o console do navegador

Google Sala de Aula

Depurando páginas web usando o console no navegador

Sobre Transcrição

A seguir: Aula 3

Programação

CURSO: PROGRAMAÇÃO > UNIDADE 6

< Lição 3: Métodos de acesso DOM >

Encontrando elementos por ID

Desafio: modificador de ID

Encontrando elementos DOM múlt...

Desafio: tags de nome personalizado

Encontrando elementos por seletor...

Resumo: métodos de acesso DOM

Computação > Programação > HTML/JS: tornando páginas web interativas > Métodos de acesso DOM

© 2024 Khan Academy

[Termos de uso](#) [Política de privacidade](#) [Aviso de cookies](#)

Ha dois métodos que retornam um único objeto do tipo `Element`, `getElementById` e `querySelector`:

```
var salsMotto = document.getElementById("salsMotto");
salsMotto.innerHTML = "Matemática é legal";
```

Os métodos `getElementsByClassName` e `getElementsByTagName` retornam um objeto `HTMLCollection` que funciona como um array. Essa coleção é "dinâmica", o que significa que a coleção é atualizada se elementos adicionais com nome de tag ou de classe forem adicionados ao documento.

```
var teamMembers = document.getElementsByClassName("team-member");
for (var i = 0; i < teamMembers.length; i++) {
  console.log(teamMembers[i].innerHTML);
}
```

O método `querySelectorAll()` retorna uma `NodeList`, que também funciona como um array. Essa lista é "estática", o que significa que a lista não será atualizada se novos elementos correspondentes foram adicionados à página. Provavelmente você não vai perceber a diferença entre dois tipos de retorno de dados quando estiver usando esses métodos, mas é bom ter isso em mente.

```
var teamMembers = document.querySelectorAll("team-member");
```

CursosPesquisar

Khan Academy

guilherme.mcosta

Programação

CURSO: PROGRAMAÇÃO > UNIDADE 6

Lição 4: Modificação DOM

Alterando atributos

Alterando estilos

Alterando classes CSS

Configurando innerHTML e textCo...

Desafio: A história interior

Criando elementos desde o início

Resumo: técnicas de modificação D...

Computação > Programação > HTML/JS: tornando páginas web interativas > Modificação DOM

Resumo: técnicas de modificação DOM

Google Sala de Aula

Modificando um elemento existente

Cobrimos várias formas de modificar aspectos de um elemento existente:

Modificando atributos

Você pode definir um atributo em um elemento definindo a propriedade de mesmo nome. Por exemplo, para alterar o `src` de uma ``:

```
imgEl.src = "http://www.dogs.com/dog.gif";
```

Além disso, você pode usar o método `setAttribute`, assim:

```
imgEl.setAttribute("src", "http://www.dogs.com/dog.gif");
```

Se você quiser remover um atributo, você deve fazer isso com `removeAttribute` - como para remover o atributo `disabled` de um botão, habilitando-o efetivamente:

© 2024 Khan Academy