

Desenvolvimento para a Web



Roteiro

- Módulo 1 – Introdução à Lógica de Programação e Algoritmos
- Módulo 2 – Desenvolvimento para a Web

Recursos

- Github
 - <https://github.com/>
 - https://rogerdudler.github.io/git-guide/index.pt_BR.html
- GitHub – Repositório do Curso
 - <https://github.com/luisalvarenga-pde/>

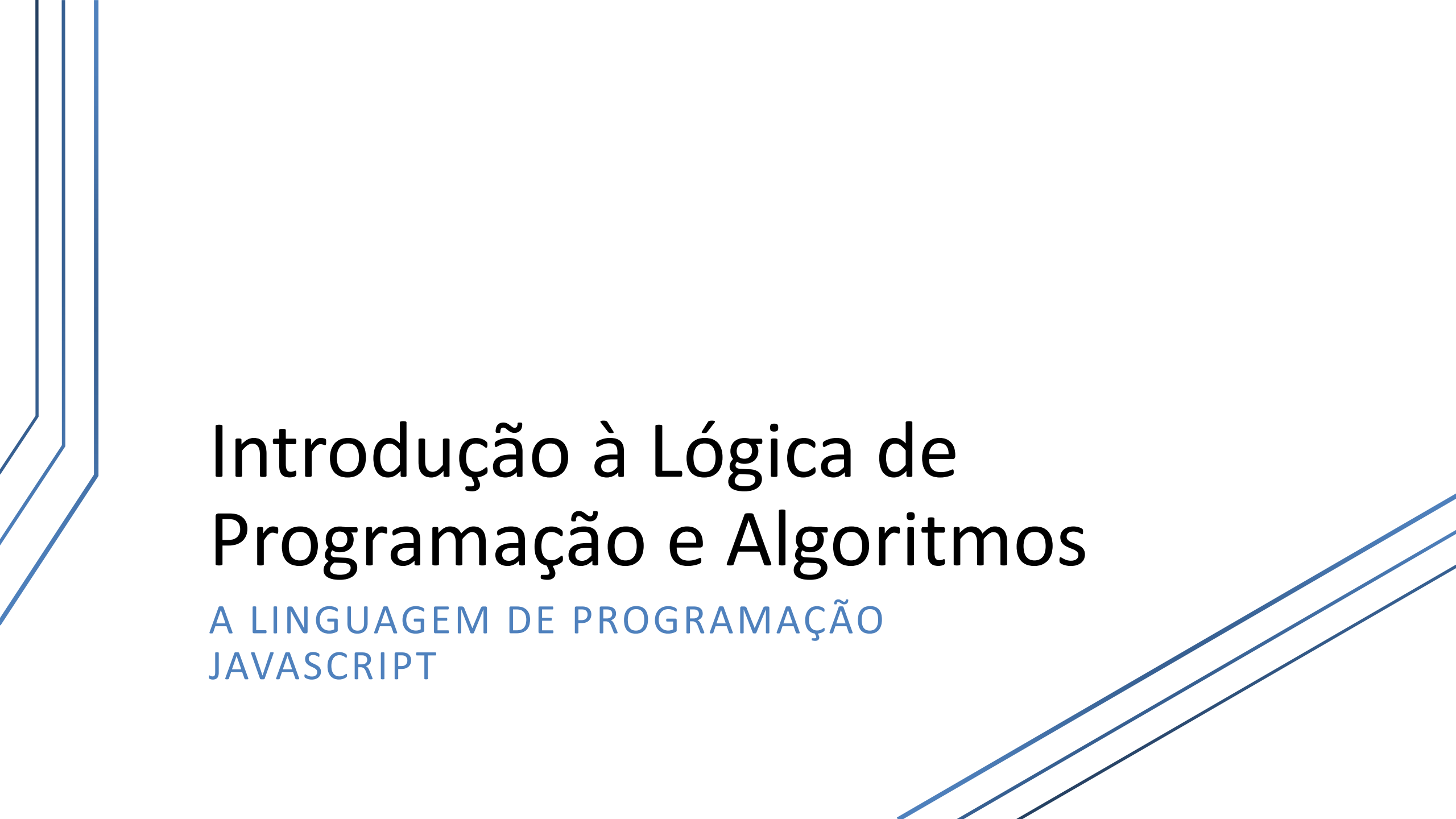
Módulo 1 - Introdução à Lógica de Programação e Algoritmos

Roteiro

- Fundamentos
- Variáveis e Atribuição
- Comandos de Entrada e Saída
- Operações Aritméticas
- Operações Relacionais, expressões lógicas e condicionais
- Estruturas de repetição
- Procedimentos e Funções
- Linguagem de Programação JavaScript

Objetivos

- Desenvolver o pensamento lógico;
- Entender como são feitos os sistemas de computador;
- Construir algoritmos com os elementos básicos de programação;
- Construir programas em Linguagem JavaScript.



Introdução à Lógica de Programação e Algoritmos

A LINGUAGEM DE PROGRAMAÇÃO
JAVASCRIPT



Roteiro

- A Linguagem JavaScript
- Imersão na Linguagem JavaScript

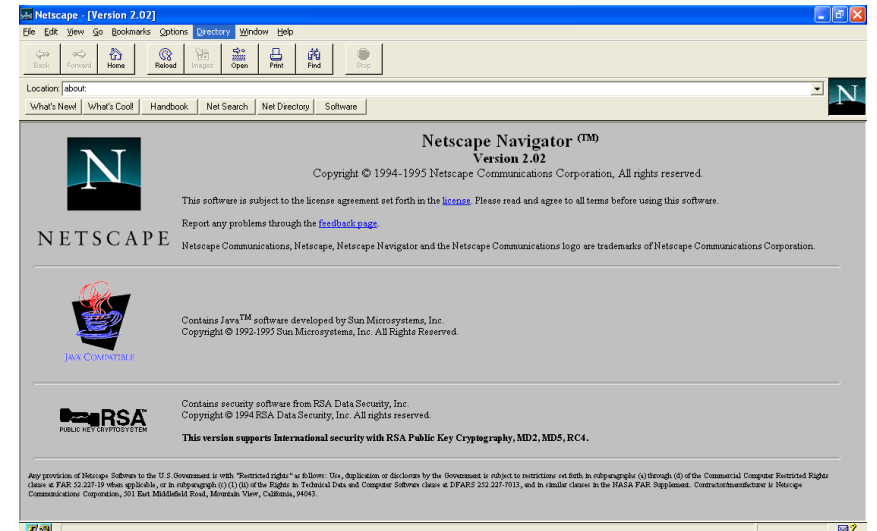


Objetivos

- Conhecer um pouco da Linguagem JavaScript
- Ter uma primeira experiência em escrever código em JavaScript
- Adquirir uma compreensão básica do que envolve escrever código em JavaScript.

A Linguagem JavaScript

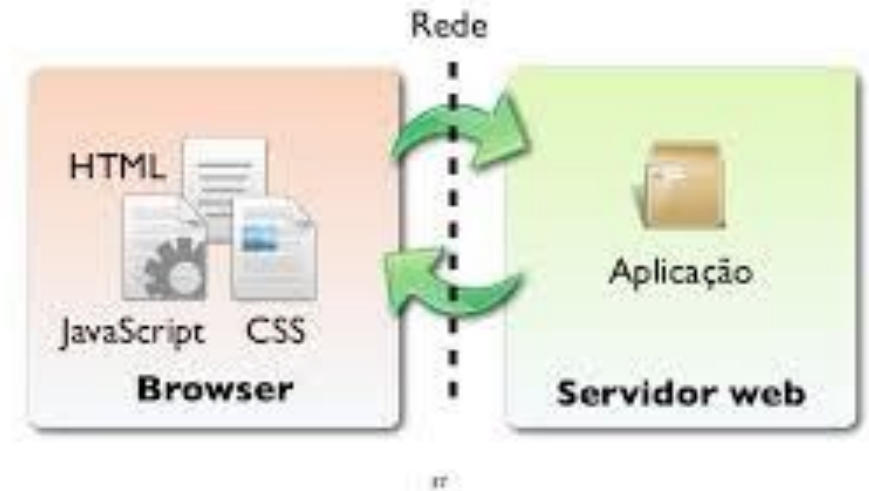
- Criada em 1995 por Brendan Eich da Netscape, nomeada inicialmente como Mocha, depois LiveScript e por fim JavaScript;
- Também chamada de JS, é a linguagem de criação de scripts para a Web;
- É utilizado por bilhões de páginas para:
 - Adicionar funcionalidades;
 - Verificar formulários;
 - Comunicar com servidores;
 - Adicionar efeitos gráficos;
 - E muitos mais.



A Linguagem JavaScript

- Linguagem Cliente – Servidor em Navegadores Web;
- É uma Linguagem de Scripts, Orientada a Objetos;
- Influenciada por C, Python e Java;
- JavaScript não está relacionada a Linguagem Java.

Arquitetura Cliente/Servidor



A Linguagem JavaScript

Os principais padrões a destacar são:

- A Linguagem Núcleo:
 - [ECMA-262](#) (ECMAScript)
 - Padrão mantido por ECMA - European Computer Manufacturers Association.
- DOM:
 - Document Object Model;
 - Define a Interface da Linguagem com o Browser;
 - Padrão mantido pelo [W3C](#) - Consórcio World Wide Web.

Imersão na Linguagem JavaScript

- Este módulo apresenta as características básicas da Linguagem JavaScript através de um passo-a-passo completamente prático.
- Vamos construir um jogo simples de "Adivinhe o número".
- Uma das coisas mais difíceis de aprender na programação não é a sintaxe da linguagem, mas como aplicá-la para resolver problemas do mundo real.
- Quanto mais você programa, melhor programador se torna.

Jogo adivinhe um número

Imagine que recebeu a seguinte especificação para criar o jogo:

Crie um jogo simples do tipo adivinhe um número. Ele deve gerar um número aleatório de 1 a 100, depois desafiar o jogador a adivinhar o número em 10 rodadas. A cada rodada deve ser dito ao jogador se ele está certo ou errado, se estiver errado, deve ser dito se o palpite é muito baixo ou muito alto. Também deve ser mostrado ao jogador os números que ele tentou adivinhar anteriormente. O jogo termina se o jogador acertar o número ou acabarem o número de tentativas. Quando o jogo acabar, deve ser dado ao jogador a opção de jogar novamente.

Jogo adivinhe um número

1. Gerar um número aleatório entre 1 e 100.
2. Gravar o número do turno que o jogador está. Iniciar em 1.
3. Dar ao jogador uma forma de adivinhar o número.
4. Após a tentativa ter sido submetida, primeiro gravar em algum lugar para que o usuário possa ver as tentativas anteriores.
5. Depois, verificar se o palpite está correto.
6. Se estiver correto:
 1. Escrever mensagem de parabéns.
 2. Impedir que o jogador insira mais respostas (isso pode bugar o jogo).
 3. Mostrar controle que permita ao jogador reiniciar o jogo.
7. Se o palpite estiver errado e o jogador ainda tem turnos sobrando:
 1. Dizer ao jogador que ele está errado.
 2. Permitir que ele insira outra resposta.
 3. Incrementar o número do turno em 1.
8. Se o jogador está errado mas não tem turnos sobrando:
 1. Dizer ao jogador que o jogo acabou.
 2. Impedir que o jogador insira mais respostas (isso pode bugar o jogo).
 3. Mostrar controle que permita ao jogador reiniciar o jogo.
9. Quando reiniciar, tenha certeza de resetar todas as variáveis e a interface do jogo, então volte para o passo 1

Jogo adivinhe um número

Configuração Inicial

- Copiar o arquivo **jogo-adivinhe-o-numero-inicio.html** no repositório do GitHub.
- Abrir o arquivo em um editor de texto (Visual Studio Code) e também no seu browser.
- Escrever o código dentro da tag **<script>** na parte inferior do HTML.

```
<script>  
// Seu código JavaScript vai aqui  
</script>
```


Jogo adivinhe um número

Variáveis

- Primeiramente, vamos declarar e inicializar as variáveis. Adicione as seguintes linhas na sua tag **<script>**.

```
var numeroAleatorio= Math.floor(Math.random() * 100) + 1;
```

```
var palpites = document.querySelector('.palpites');
```

```
var ultimoResultado = document.querySelector('.ultimoResultado');
```

```
var baixoOuAlto = document.querySelector('.baixoOuAlto');
```

```
var envioPalpite = document.querySelector('.envioPalpite');
```

```
var campoPalpite = document.querySelector('.campoPalpite');
```

```
var contagemPalpites = 1;
```

```
var botaoReinicio;
```

Jogo adivinhe um número

Variáveis

- A primeira variável — numeroAleatorio — é atribuído um número aleatório entre 1 e 100, calculado usando um algoritmo matemático.
- As próximas três variáveis são criadas para guardar uma referência para os parágrafos resultantes em nosso HTML, e são usadas para inserir valores nos parágrafos no código:

```
<p class="palpites"></p>
```

```
<p class="ultimoResultado"></p>
```

```
<p class="baixoOuAlto"></p>
```

Jogo adivinhe um número

Variáveis

- As próximas duas variáveis armazenam referências para o campo de texto e o botão de envio e são usados para controlar o envio do palpite.

```
<label for="campoPalpite">Digite seu palpite: </label>  
<input type="text" id="campoPalpite" class="campoPalpite">  
<input type="submit" value="Enviar palpite" class="envioPalpite">
```
- As últimas duas variáveis (contagemPalpites e botaoReinicio) são usadas para armazenar a contagem dos palpites do usuário, e o outro é uma referência para o botão de reset, que não existe ainda (mas irá existir).

Jogo adivinhe um número

Funções

- Em seguida, adicione as seguintes funções abaixo da declaração das variáveis:

```
function conferirPalpite() {  
}
```

```
function configFimDeJogo() {  
}
```

```
function reiniciarJogo() {  
}
```

Jogo adivinhe um número

Operadores Aritméticos e Relacionais

Operador	Nome	Exemplo
+	Adição	6 + 9
-	Subtração	20 – 15
*	Multiplicação	3 * 7
/	Divisão	10 / 5

Operador	Nome	Exemplo
==	Igualdade	5 == 2+4
!=	Diferença	'Chris' != 'Ch' + 'ris'
< ; <=	Menor que; Menor ou igual	10 < 6; 10 <= 9
> ; >=	Maior que; Maior ou igual	10 > 20; 10 >= 5 + 5

Jogo adivinhe um número

Condicionais

- Voltando à nossa função **conferirPalpite()**, nós queremos verificar se o palpite do jogador está correto ou não, e responder apropriadamente.
- Neste ponto, substitua sua função **conferirPalpite()** atual pela versão do arquivo **conferir_palpite.txt**, contida no repositório do GitHub.

Jogo adivinhe um número

- A primeira linha declara uma variável chamada **palpiteUsuario** e define seu valor igual ao valor inserido pelo jogador no campo de texto.
- Em seguida, encontramos nosso primeiro bloco de código condicional. Neste caso, o teste está verificando se a variável **contagemPalpites** é igual a 1 (isto é, se essa é ou não a primeira tentativa do jogador):
`contagemPalpites === 1`
- Se a condição for verdadeira, nós tornamos o conteúdo do parágrafo de palpites, `<p class="palpites"></p>` igual a "Palpites anteriores: ". Caso contrário, o texto não é exibido.
- Após o bloco de teste, acrescenta-se o valor atual de **palpiteUsuario** ao final do parágrafo palpites, mais um espaço em branco para que haja espaçamento entre cada palpite mostrado.

Jogo adivinhe um número

- O próximo bloco faz as seguintes conferências:
 - O primeiro `if(){}` confere se o palpite do jogador é igual ao número aleatório (**numeroAleatorio**) definido no topo do nosso JavaScript. Se for, o jogador adivinhou corretamente o número e venceu o jogo. Então mostramos ao jogador uma mensagem de congratulação com uma agradável cor verde, limpamos o conteúdo do parágrafo que informa sobre o palpite ser alto ou baixo `<p class="baixoOuAlto"></p>`, e executamos uma função chamada **configFimDeJogo()**, que iremos discutir mais tarde.
 - Agora encadeamos outro teste ao final deste usando uma estrutura `else if(){}` . Este teste confere se o palpite do jogador é sua última tentativa. Se for, o programa faz o mesmo que no bloco anterior, porém com uma mensagem de fim de jogo ao invés do texto de congratulação.
 - O bloco final encadeado ao final do código `else {}` contém código que só é executado se nenhum dos outros dois testes retornar verdadeiro. Neste caso informamos que o palpite está errado e executamos outro teste para checar se o palpite foi maior ou menor do que a resposta certa, exibindo uma mensagem apropriada se foi maior ou menor.

Jogo adivinhe um número

- As próximas três linhas nos deixa preparados para o próximo palpite a ser submetido. Nós somamos 1 à variável **contagemPalpites** para que o jogador use sua tentativa (++ é uma operação de incremento — incrementa em 1), e o campo de texto do formulário de inserção seja esvaziado e focado novamente, pronto para que o próximo palpite seja inserido.

Jogo adivinhe um número

Eventos

- Neste ponto temos uma função **conferirPalpite()** bem implementada, mas ela não irá fazer nada pois nós não a chamamos ainda.
- Idealmente nós queremos que ela seja acionada quando o botão "**Enviar palpite**" for pressionado, e para fazer isso precisamos usar um evento.
- Eventos são ações que acontecem no navegador, como um botão sendo clicado, ou uma página carregada; ações as quais podemos responder executando blocos de código.
- Os construtores que monitoram os acontecimentos de eventos são chamados de ***event listeners***, e os blocos de código executados em resposta ao acontecimento do evento são chamados de ***event handlers***.

Jogo adivinhe um número

Eventos

- Adicione a seguinte linha abaixo da chave de fechamento da sua função **conferirPalpite()**:

```
envioPalpite.addEventListener('click', conferirPalpite);
```

- Aqui nós estamos adicionando um ***event listener*** ao botão **envioPalpite**.
- Tente agora salvar e atualizar o código. O seu código deve funcionar até um certo ponto. O único problema é que se você acertar o palpite ou ficar sem mais tentativas o jogo irá falhar, porque ainda não definimos a função **configFimDeJogo()**.

Jogo adivinhe um número

Finalizando a funcionalidade do jogo

- Substitua a função **configFimDeJogo()** pelo seguinte código.

```
function configFimDeJogo() {  
  
    campoPalpite.disabled = true;  
    envioPalpite.disabled = true;  
    botaoReinicio = document.createElement('button');  
    botaoReinicio.textContent = 'Iniciar novo jogo';  
    document.body.appendChild(botaoReinicio);  
    botaoReinicio.addEventListener('click', reiniciarJogo);  
}
```

Jogo adivinhe um número

Finalizando a funcionalidade do jogo

- As primeiras duas linhas desabilitam a entrada de texto do formulário e o clique do botão, definindo a propriedade `disabled` (desabilitado) de cada um como `true` (verdadeiro). Isso é necessário, pois se não o fizermos, o usuário poderia submeter mais palpites depois do jogo ter terminado.
- As próximas três linhas geram um novo elemento `<button>`, define o texto de seu rótulo como "**Iniciar novo jogo**", e o adiciona ao final do nosso HTML.
- A linha final define um ***event listener*** em nosso botão, para que quando seja clicado, uma função chamada **reiniciarJogo()** seja executada.

Jogo adivinhe um número

Finalizando a funcionalidade do jogo

- Neste ponto, substitua a função **reiniciarJogo()** atual pela versão do arquivo **reiniciar_jogo.txt**, contida no repositório do GitHub.
- Esse longo bloco de código redefine completamente tudo ao modo como era no início do jogo, para que o jogador possa jogá-lo novamente.
- Neste ponto você deve ter um jogo simples, mas completamente funcional.

Jogo adivinhe um número

Finalizando a funcionalidade do jogo

- Coloca o valor da variável **contagemPalpites** novamente igual a 1.
- Limpa todos os parágrafos de informativos.
- Remove o botão resete do nosso código.
- Habilita os elementos do formulários, esvazia e direciona o foco ao campo de texto, pronto para que um novo palpite seja inserido.
- Remove a cor de fundo do parágrafo **ultimoResultado**.
- Gera um novo número aleatório para que o jogador não esteja tentando adivinhar o mesmo número novamente!