

Imersão Dev - Aula 3

Rafaella: Olá Devs! Gostaríamos de dar as **boas vindas** ao nosso terceiro dia de Imersão Dev! Estamos muito felizes de ver os resultados de vocês nas redes sociais, e orgulhosa do quanto vocês têm se superado a cada dia!

Nesta aula, nos aprofundaremos um pouco mais na **lógica de programação** com um projeto totalmente novo para você colocar no seu portfólio! Inclusive, não esqueçam de postar no **LinkedIn** e no **Discord** para podermos acompanhar o progresso.

O projeto desta aula está muito legal, então já abram o **CodePen** e vamos lá!

Lembrando tudo o que já vimos nas aulas anteriores, aprendemos sobre variáveis, sobre o console que é a tela que aparece onde conseguimos imprimir coisas com o `console.log()`, abordamos as funções e como criamos aqueles blocos de código para chamarmos quando quisermos, vimos o funcionamento do `onClick`, em que um botão é clicado e o que criamos é executado neste evento, aprendemos sobre o `.getElementById()` e tudo isso será trazido para esta e para as próximas aulas da Imersão.

Paulo: Antes de mais nada, gostaríamos de agradecer todo mundo que está postando seus projetos, inclusive no Instagram, e está marcando a gente em *hashtags* e nos *stories*, e não só no Discord. Muito obrigado, assim vocês incentivam as pessoas ao redor e compartilham conhecimento!

É importante avisar que ainda dá tempo de se inscrever e acompanhar o conteúdo! Então convide as amigas e amigos que curtiram e comentaram as postagens a fazer as aulas aqui da Imersão Dev e criar seus próprios projetos!

Guilherme: O projeto desta aula é um super desafio! Pediremos para o computador escolher um número de zero a dez, e tentaremos adivinhar qual o valor escolhido.

Passaremos por algumas decisões, então se o "chute" for o mesmo número que o programa escolheu, teremos que informar que deu certo! Mas caso contrário, teremos que criar outras condições dentro da programação.

Se algo específico acontecer, queremos que alguma coisa específica ocorra, e se não acontecer, outra coisa deve ser executada. Esse será o foco desta aula.

Já temos um *template* com os personagens da série *Strangers Things*, o título "Mentalista - ImersãoDev_" e o campo de *input* em um *layout* muito bonito que vamos usar.

Para não perdermos o costume, faremos o *fork* do projeto para a sua conta própria e começar o trabalho. A ideia é digitarmos um número de zero a dez, mas não falaremos para o computador escolher um valor desde o início.

Afinal, queremos realizar alguns testes para vermos de fato se temos os comportamentos esperados quando a resposta está certa e quando está errada. Então começaremos criando uma variável `var` na aba JS no CodePen, e a chamaremos de `numeroSecreto` com um valor conhecido, como `3` por exemplo.

```
var numeroSecreto = 3
```

Rafaella: Para conseguirmos dar um primeiro palpite e pegar o valor que a pessoa irá digitar no campo, lembraremos da aula passada em que criamos uma nova função com um bloco de código para ser chamado no momento do clique no botão, e isso é o que acontece com o `onClick`.

Então é interessante buscarmos o `<button>` na aba HTML que é o próprio botão de "chutar" e já possui algumas informações como seu tipo `"submit"`, que é quando queremos enviar as informações. Também há o `onClick` que irá passar a função que chamaremos logo após o clique.

Já aparece o nome da função `Chutar()` que iremos utilizar, e bastará a criarmos em nosso código JavaScript. Podemos copiar e colar para termos tudo certo, e no JS criaremos esta `function`.

Abriremos as chaves para colarmos o nosso bloco de código com tudo o que queremos que seja executado, e dentro deveremos pensar o que queremos fazer.

Primeiro, queremos pegar o valor do "chute" para depois podermos comparar e ver se está certo ou não. Para isso, também olharemos o `<input>` no HTML que é do tipo `"number"`, o mesmo que vimos anteriormente, pois só podemos inserir números neste tipo de input.

Também tem o `id="valor"`, e sempre que quisermos referenciar algum elemento no HTML, usamos o já conhecido `.getElementById()` para "puxar" as informações inteiras.

Portanto, atribuiremos a uma variável chamada `Chute` criada dentro da função `Chutar()`. Em seguida, após o sinal de igualdade, aplicaremos o `document.getElementById()` recebendo o `id` do elemento que queremos pegar, que é o `"valor"` já visto.

Quando fazíamos este `.getElementById()` nas aulas anteriores, ele vinha com a *tag* inteira do input com seus atributos, e não é o que queremos. E quando precisávamos apenas do valor que foi digitado, bastava aplicar o `.value` na nova variável criada para chamar o elemento.

Mas não precisamos fazer isso em duas variáveis diferentes, e desta vez poderemos já colocar essa transformação na variável `Chute()`, colocando `.value` no final da sentença.

```
var numeroSecreto = 3

function Chutar() {
  var chute = document.getElementById("valor").value;
}
```

Rafaella: Outro ponto importante que vimos na aula anterior e que precisamos fazer agora é que, sempre que pegamos algum número dentro de um input, deveremos transformá-lo do tipo texto *string* para o tipo **"number"**.

Por exemplo, no projeto de hoje queremos comparar com números, então precisamos usar o tipo correto para tudo ficar mais claro para o programa fazer as operações.

Para isso, usamos o comando **parseInt()** e não o **parseFloat()** desta vez, pois não lidaremos com valores com casas decimais do tipo **float**, afinal são só valores inteiros **int** de zero a dez.

Envolveremos nos parênteses, e cada comando que estamos vendo como o **parseInt()** mesmo por exemplo, aparece antes e passamos para dentro dele todos o valor que queremos transformar para inteiro.

Se quisermos descobrir apenas o valor de algo, basta colocar um ponto ao final do que queremos obter com **.value**. Então existem diferentes formas de escrevermos alguns comandos próprios do JavaScript, que conseguimos entender melhor na prática.

Na aula anterior já fizemos desta forma, em que utilizamos três variáveis com algo para cada uma delas, mas poderemos fazer essas três coisas juntas em apenas uma única variável.

É sempre interessante testar e tentar agregar a alguma variável coisas que já fizeram em outros projetos com várias, então fica a dica de exercício.

Continuando, vamos usar o **console.log()** para imprimirmos a variável **chute** no Console e vermos qual o valor que pegaremos do input de fato, que é mais fácil para identificarmos qualquer erro.

Guilherme: Antes de mais nada, não esqueçamos de ir até a opção "Settings" na barra superior, acessar o item "Behavior" na lista lateral para aplicarmos a nossa formatação com a opção "On" ativada em "Format on Save", e depois acessar a opção "Pen Details" para escrevermos as tags "imersaodev, alura" no campo "Tags".

Por fim, podemos salvar o arquivo e testar inserindo o número "7" no campo de texto.

```
var numeroSecreto = 3

function Chutar() {
  var chute = parseInt(document.getElementById("valor").value);
  console.log(chute)
}
```

Guilherme: Escrevendo o valor e clicando em "Chutar", veremos o "7" aparecendo no Console.

Rafaella: Perfeito, então está certo, está fora das *strings* e está aparecendo como número inteiro.

Guilherme: Já temos o número secreto e o "chute". Agora temos que verificar se a resposta está certa ou errada, bem como informar a pessoa.

O computador sozinho ainda não sabe que o valor secreto `3` e o `chute` terão que ser comparados entre si, então teremos que mostrar esses passos para ele.

Começaremos verificando se o `chute` é igual ao `numeroSecreto` ou não. Mas não podemos simplesmente rodar uma pergunta escrita em português que com certeza receberemos um erro de sintaxe no Console.

O JavaScript tem como língua principal o idioma inglês, então faremos uma tradução. A palavra "se" em português que estabelece uma condição se traduz para `if`. Ao escrevermos este termo e é reconhecido, até a cor fica diferente no JS.

Sempre que temos uma condição no JavaScript, precisaremos englobar toda a condição nos parênteses, semelhante ao que fizemos no `parseInt()` que transformou todo o conteúdo que passamos nos `()`.

Faremos a mesma coisa para a nossa comparação, verificando se o `chute` é igual ao nosso `numeroSecreto`, que é nossa condição. Caso seja verdade, queremos que realize algo específico, como mostrar no Console que `acertou`.

```
var numeroSecreto = 3

function Chutar() {
  var chute = parseInt(document.getElementById("valor").value);
  console.log(chute);
  if (chute for igual ao numeroSecreto)
    acertou
}
```

Guilherme: Se rodarmos o código como está, o JavaScript ainda não entenderá e "explodirá" um erro de sintaxe `SyntaxError` no Console.

Podemos substituir essa parte `for igual ao` da condição por apenas dois dígitos com sinais de igualdade: `==`. Isso é muito importante para prestarmos atenção.

Sempre que utilizamos apenas um sinal `=`, estamos pegando um determinado conteúdo ou valor e jogando dentro da memória de uma variável, ou seja, estamos fazendo uma **atribuição**.

Mas se queremos fazer uma **comparação**, como no caso de sabermos se o chute é igual ao número secreto, usamos dois sinais de igualdade `==`. Se a condição for verdadeira, iremos realizar algum **comportamento**.

Para identificarmos o que é o comportamento do `if` e o que faz parte do comportamento da nossa função, também abriremos chaves ou os "bigodes".

Tudo o que escrevermos dentro das `{ }` será executado se o `chute` for de fato igual ao `numeroSecreto`. Então partiremos do mesmo princípio do `acertou`, o qual será colocado entre aspas dentro de um `console.log()` para exibirmos essa mensagem no Console.

Rafaella: Deixaremos um artigo sobre esses **operadores relacionais**, pois não apenas conseguimos comparar se são iguais, mas também podemos saber se são diferentes, maiores ou menores. Inclusive temos até o triplo sinal de igualdade `===` que possui uma ação específica e diferente dos outros.

Guilherme: Vamos executar este programa para vermos o comportamento esperado. Já sabemos que o número secreto é `3`, então salvaremos e executaremos o projeto para vermos se, ao escrevermos o valor "3" que sabemos ser igual ao `numeroSecreto`, o programa funcionará corretamente.

```
var numeroSecreto = 3

function Chutar() {
  var chute = parseInt(document.getElementById("valor").value);
  console.log(chute);
  if (chute == numeroSecreto) {
    console.log("acertou");
  }
}
```

Guilherme: Apertando o botão "Chutar", aparecerá o `3` e a mensagem `"acertou"` no console.

Agora faremos o caminho oposto. Limparemos o Console e escreveremos o número "1" no campo de texto do projeto, e sabemos que não é a resposta correta. Quando clicarmos em "Chutar" novamente, o Console exibirá apenas o valor `1` e nenhuma mensagem.

Porém, ainda não faz muito sentido, e seria interessante mostrarmos que a pessoa errou a resposta. Mas como faremos isso?

Rafaella: Além do `if()` sendo o "se" que executa alguma coisa sob esta condição, também podemos fazer um "se não" para o caso contrário.

Caso o número secreto seja igual ao valor que "chutamos", receberemos a mensagem de "acertou", e **se não** for, receberemos a mensagem de "errou". Nesta última situação, usaremos o termo `else` do JavaScript.

Não precisaremos fazer uma outra condição somente para o `else`, pois já está implícito que se trata de uma situação diferente da colocada no `if()` anterior. Então poderemos abrir chaves novamente para colocarmos o que queremos executar neste caso contrário, e depois imprimiremos um `console.log()` com a mensagem de `"errou"`.

Guilherme: Vamos testar! Salvando e executando novamente, limparemos o Console e escreveremos o número "1" no campo de texto para depois clicarmos em "Chutar".

```
var numeroSecreto = 3

function Chutar() {
  var chute = parseInt(document.getElementById("valor").value);
  console.log(chute);
  if (chute == numeroSecreto) {
    console.log("acertou");
  } else {
    console.log("errou");
  }
}
```

Guilherme: Feito isso, exibiremos o `1` e a mensagem `"errou"`, como esperávamos.

Rodaremos novamente para garantirmos que nada mudou no comportamento do programa, "chutaremos" o número "3" que sabemos ser o acerto. E com isso, receberemos a mensagem de `"acertou"`.

Rafaella: Queremos testar uma outra coisa, como colocar o número "12" no input por exemplo.

Guilherme: Quando clicarmos em "Chutar" com este valor, mostraremos a mensagem de que erramos.

Rafaella: Mas colocamos o desafio de "Digite um número de 0 a 10", mas inserimos um valor maior no input. Seria interessante darmos um destaque para a pessoa, que pode não ter lido esta frase por exemplo e está tentando dar "chutes" fora dos limites determinados.

Então poderíamos colocar uma outra condição; se o número for igual ao valor "chutado", receberemos o texto de acerto no Console, mas se for maior do que o número dez - afinal não precisa ser maior -, poderemos mandar uma mensagem personalizada dizendo que o "chute" deve estar no intervalo entre os valores zero e dez, e por fim o `else` estará no final indicando que a pessoa errou.

Guilherme: Ainda temos uma questão, em que o `if()` valida uma condição para realizar um determinado comportamento e o `else` é executado no caso desta condição principal não acontecer. Neste caso específico, poderemos juntar os dois.

Escreveremos um `else` depois do primeiro `if()`, seguido de outro `if()` na mesma linha, o qual receberá entre os parênteses a condição que queremos, e depois abrir e fechar as chaves para a execução que queremos que faça.

Já o segundo `else` que executa a mensagem de `"errou"` e não possui nenhuma condição, será deixado ao final. Agora sim podemos fazer a condição.

```
var numeroSecreto = 3

function Chutar() {
  var chute = parseInt(document.getElementById("valor").value);
  console.log(chute);
  if (chute == numeroSecreto) {
    console.log("acertou");
  } else if () {

  } else {
    console.log("errou");
  }
}
```

Rafaella: Faremos primeiro para o caso do "chute" ser maior do que o valor máximo do número dez.

Guilherme: Certo! Então, se queremos colocar que o `chute` é maior do que `10`, usaremos o sinal de "maior que": `>`. Quando isso acontece, queremos passar a mensagem `"Você deve digitar um número de 0 a 10"` no `console.log()`.

```
var numeroSecreto = 3

function Chutar() {
  var chute = parseInt(document.getElementById("valor").value);
  console.log(chute);
  if (chute == numeroSecreto) {
    console.log("acertou");
  } else if (chute > 10) {
    console.log("Você deve digitar um número de 0 a 10")
  } else {
    console.log("errou");
  }
}
```

Paulo: Essa estrutura de **cascata de condições** com `if()`, vários `else if` e `else`, e até outras que veremos futuramente, é bem comum quando programamos algo que possui comportamentos específicos para ações específicas, como uma **árvore de decisões** com várias possibilidades do programa.

Rafaella: Estamos utilizando `else if` para validar outra coisa, mas se no lugar disso colocássemos apenas `if()`, o JavaScript não entenderia que é uma condição paralela a essa que vimos primeiro, pois entenderia que é outro bloco de condição. Então seria um novo `if()` onde poderíamos colocar um `else if` dentro também.

Por isso é interessante usarmos `else if` mesmo, pois estamos comparando o mesmo valor e vendo possíveis inputs e respostas que a pessoa colocará na página. Vamos testar para ver se está funcionando.

Guilherme: Então vamos lá. Colocaremos um número maior no campo de texto para vermos se o "12" que estamos escrevendo recebe a mensagem que queremos.

Clicando em "Chutar", receberemos a mensagem de `"Você deve digitar um número de 0 a 10"`. E caso coloquemos um número de zero a dez como "4" por exemplo e errarmos a resposta, receberemos a mensagem de `"errou"` apenas, como esperávamos. Então temos esses dois comportamentos.

Rafaella: Mas e se colocarmos o número -4 por exemplo, Gui?

Guilherme: Vamos ver, vou colocar o número -4 e chutar. Aparece o "errou".

Rafaella: Então seria legal deixarmos claro que o número não pode ser maior que 10 nem menor que 0. Podemos fazer essa condição dentro do `else if()`. Então podemos colocar que se o chute for maior que 10 **ou** menor que 0, deve aparecer a mensagem.

```
else if (chute > 10 ou chute < 0) {  
    console.log("Você deve digitar um número de 0 a 10");  
}
```

Rafaella: Isto também não é algo que utilizamos. Primeiro porque está em português e segundo porque não é assim que o JavaScript funciona. Em outras linguagens pode ser que funcione, mas no JavaScript nós usamos duas barras verticais `||`. Isso também está no artigo que deixaremos aqui embaixo. Na verdade, o artigo tem várias tabelas que vão mostrando essas comparações.

```
else if (chute > 10 || chute < 0) {  
    console.log("Você deve digitar um número de 0 a 10");  
}
```

Rafaella: Então `||` significa "ou". Agora vamos testar para ver se colocando um número negativo ou maior que 10 ele funciona.

Guilherme: Vamos lá. Vou testar com o mesmo -4 e chutar. A mensagem "Você deve digitar um número de 0 a 10" aparece certinho. Ficou bem legal isso. Inicialmente, o que vem na cabeça é uma outra condição `else if()`, né?

Paulo: Poderia até ser, mas nesse caso não há necessidade, funcionou bem a decisão da Rafa.

Rafaella: É legal vermos que existem esses operadores que juntam coisas, que você pode fazer esse tipo de operação em uma única linha. Até porque a mensagem era a mesma, não precisaríamos colar e colocar o mesmo comando se fosse uma condição diferente.

Guilherme: Olha, ficou legal, estou gostando. Lembram que falei no começo sobre o próprio programa escolher um número de 0 a 10? Não é o que está acontecendo. Se eu vou e escolho o número 7, por exemplo, e vou mostrar o projeto para alguém talvez isso não fique muito bacana. Isso porque se a pessoa acertar o número, ela vai sempre conseguir acertar! É mais interessante que o programa faça o sorteio do número sozinho.

Este é um momento muito interessante, reparar nessa questão de número aleatório, como pegamos um elemento de página com `getElement` e outras coisas que estamos realizando que são funcionalidades que a própria linguagem nos fornece como um número aleatório,

ou pseudo número aleatório para os mais puristas. Existe uma biblioteca do JavaScript que pode nos ajudar nessa função. E para rodarmos isso, vamos abrir o console.

Paulo: Para testarmos e brincarmos um pouco com a maneira que ele sorteia um número para nós, é isso?

Guilherme: Isso. Existe uma função no JavaScript chamada `Math.random()`. E o que ela faz? Ela vai sortear um número aleatório. Se fizermos essa função e dermos enter ele sorteia o número `0.4460231090508771`, nesse caso. O que é esse número bizarro? Vamos pesquisar essa função na biblioteca do JavaScript e ver o que ela fala.

Paulo: Isso é algo que alguns alunos já fizeram lá no Discord, procurar no Google para cair na biblioteca padrão. Embora nem sempre seja a padrão de fato, a maioria vai cair no site do Mozilla Developer Network, que é o criador de muitos padrões dos navegadores, é uma comunidade muito forte. Essa e a Stack Overflow vocês ainda ouvirão muito sobre elas.

Guilherme: É verdade. Então, acessando o site, ele diz que a função `Math.random()` retorna um número pseudo aleatório em um intervalo de 0 e 1. O 0, inclusive, pode aparecer, mas o 1 não é incluído. Ou seja, todas as vezes que executarmos essa função, ela sempre vai gerar um número maior do que 0 menor do que 1. Ele gera uma sequência de números muito doida que não faz muito sentido para o nosso projeto. Nós queremos um número inteiro de 0 a 10. Se ele gera esse número muito maluco, o que podemos fazer? Vamos jogar uma vírgula para o lado! Vou colocar `Math.random()` e multiplicar esse número por 10.

```
Math.random() * 10
```

Guilherme: Quando damos o Enter, aparece o `3.336653397286443`, o `7.552587793007845` e por aí vai. É mais ou menos o que queremos, mas temos um problema, se ficarmos executando isso várias e várias vezes, nunca vai aparecer o número 10, sempre vai aparecer do zero ao número 9. E o nosso label principal é "Digite um número de 0 a 10". Para colocarmos esse número de 0 a 10, no lugar de multiplicar por `10`, eu vou multiplicar por `11`.

```
Math.random() * 11
```

Guilherme: Aparece um 10! Demos muita sorte porque é totalmente aleatório isso. Enfim, deu um número `10.064706517673100`, mas esses números depois do ponto são desnecessários. A solução, porém, nós já conhecemos: lembram que anteriormente fizemos um `parseInt()` do valor que era digitado?

```
var chute = parseInt(document.getElementById("valor").value);  
console.log(chute);
```

Guilherme: O que a pessoa digita já é convertido englobando ele todo, então farei isso. Vou colocar todo o código `Math.random() * 11` com o `parseInt()` englobando ele todo.

```
parseInt(Math.random() * 11)
```

Guilherme: Ele vai gerar um número aleatório, vai multiplicar esse número por onze e depois vai transformar esse número em um número inteiro, removendo essas casas decimais. Será? Ao executar a primeira vez, conseguimos um 8, depois 4, depois 9, depois 10. Apareceu até um 0! Nossa, eu estou muito feliz! Esse código é exatamente o que precisamos para que o nosso computador gere um número aleatório. Então agora o número secreto sempre será gerado por esse valor.

```
var numeroSecreto = parseInt(Math.random() * 11);
```

Guilherme: Agora vamos testar isso! Mas temos um problema, quando nós testarmos e rodarmos, vamos ter só uma chance e caso a gente erre, nunca vamos saber qual o número aleatório.

Paulo: Na verdade, Guilherme, vai dar sim para saber. Porque tem um detalhe que você fez aí, não sei se propositalmente, esse sorteio está ocorrendo fora da função `Chutar()`. Então toda vez que chutarmos ele vai executar de novo, desde que a gente não clique no Run novamente. Lembra que essa linha fora da função só vai ser executada quando carregada? Então a variável vai sortear um número e vai ficar com esse número secreto na cabeça. Toda vez que clicarmos em Chutar, ele vai comparar com o que digitamos, então poderemos chutar várias vezes até acertar.

Guilherme: Boa. Vamos fazer isso, então. Salvei o projeto, limpei o terminal e vou rodar. Agora é a hora da verdade. Vamos descobrir qual dos três aqui é sortudo mesmo. Cada um chuta uma vez, pode ser? Vai lá, Rafa.

Rafaella: Vou chutar o 4.

Guilherme: Errou! Vai lá, Paulo.

Paulo: Eu quero 0

Guilherme: Errou também. Vou chutar 1. Errei.

Rafaella: 7!

Guilherme: Errou.

Paulo: 9!

Guilherme: Errou! Vou chutar o 3, errei!

Rafaella: Meu Deus, gente! 6!

Guilherme: Errou! 2 também não é.

Rafaella: 8!

Guilherme: Não.

Rafaella: É o 10 então.

Guilherme: Não é possível!

Paulo: Está faltando o 5!

Guilherme: 5! Agora sim acertamos.

Paulo: Chuta de novo o 5 só para vermos se funcionou aquilo que eu falei.

Guilherme: Acertou novamente.

Paulo: Está vendo? Ele guardou. Então o mecanismo está funcionando e acho que aqui ficou uma lição boa para os alunos, já que ele manteve o sorteio do número porque esse número secreto foi sorteado fora da função. Inclusive, ele podia estar abaixo da função. Se essa linha estivesse dentro da função `Chutar()` o que ia ser? Daria aquele problema em que toda vez que alguém clica em "Chutar" o computador sorteia um novo número e verifica. Aqui, não. Ele só sorteou na vez que clicamos no Run, só nesse momento. Então reparem que o tal do escopo, que é onde declaramos, onde colocamos cada coisa, não é só declaração de uma variável, eu pedi para ele fazer alguma coisa que pode impactar muito a execução do programa.

Rafaella: Vamos fazer isso ainda.

Guilherme: Rafa, eu confesso que foi muito divertido essa rodada.

Rafaella: Eu também gostei.

Guilherme: Porque nós falamos todos os números.

Paulo: Humilhante, foi humilhante.

Rafaella: Quais as chances disso acontecer enquanto nós gravamos?

Guilherme: Isso foi até tenso. Edição, arruma isso aí!

Paulo: Não arruma, não. Eu quero que mostre os 10 porque realmente foi muito bacana.

Guilherme: Beleza! Rafa, o que podemos fazer agora? Algo desse projeto te preocupa ou dá para melhorarmos?

Rafaella: Sim! Aparecer na página. Acho que ficar vendo o console é algo que incomoda bastante e nós já vimos na aula anterior como conseguir imprimir o resultado que estamos querendo. Por exemplo, podemos imprimir que você acertou, podemos imprimir que você errou, como já estamos imprimindo no console, podemos imprimir o número sorteado para a pessoa não ficar nesse negócio de ter que acertar todos para descobrir qual era. Então, vamos começar primeiro pegando onde queremos, no HTML, adicionar esse resultado, essa escrita.

No HTML conseguimos dar uma olhada e achar, já tem um `h2` preparado com o id `resultado`.

```
<h2 class="resultado" id="resultado"></h2>
```

Rafaella: Recordando o que já vimos, conseguimos referenciar algum elemento no HTML com aquele `getElementById()` e o id que queremos referenciar é esse `resultado`. Nós queremos escrever o código dentro dessas duas tags `h2`. Então vamos no JavaScript, podemos colocar fora do `if()`, não precisa colocar dentro dele, mas dentro da nossa função, nós vamos referenciar, vamos criar uma variável que vai ser a variável `elementoResultado`. Ele será o `document.getElementById()` e o id é `resultado`.

```
var elementoResultado = document.getElementById("resultado")
```

Rafaella: Para escrevermos alguma coisa, colocarmos dentro da tag com o `innerHTML` que vimos anteriormente. Para isso, vamos fazer ele antes dos `console.log()` e colocamos o elemento que queremos alterar, o `elementoResultado().innerHTML`. Vamos passar a mensagem que queremos imprimir na nossa tela, que vai ser o "Você acertou" em string mesmo.

Também vamos alterar o `else if()` para quando for menor que 0 ou maior que 10 mandar a mensagem "Você deve digitar um número entre 0 e 10". E em `else()` faremos a mesma coisa, pegamos o `elementoResultado.innerHTML` e colocar a mensagem de erro.

Guilherme: Aqui podemos falar qual era o número secreto ou ainda não?

Rafaella: Podemos. "Errou, o número secreto era " + .

Paulo: Só para ficar um pouquinho mais bonito, certo? Porque já funciona aqui no `console.log()`, mas fica aquela coisa meia estranha secreta.

Rafaella: Isso. E como já temos o número secreto guardado, ele foi randomizado, mas nós temos lá em cima a variável que guardamos o número secreto, podemos usar a mesma variável `numeroSecreto` para imprimir.

```
var numeroSecreto = parseInt(Math.random() * 11);
```

```
function Chutar() {
    var elementoResultado = document.getElementById("resultado")
    var chute = parseInt(document.getElementById("valor").value);
    console.log(chute);
    if (chute == numeroSecreto) {
        elementoResultado.innerHTML = "Você acertou"
    } else if (chute > 10 || chute < 0) {
        elementoResultado.innerHTML = "Você deve digitar um número de 0 a 10";
    } else {
        elementoResultado.innerHTML = "Errou, o número secreto era " +
numeroSecreto;
    }
}
```

Rafaella: Vamos testar agora! Salvar, rodar e ver se aparece na tela.

Guilherme: Que número?

Rafaella: 4!

Guilherme: Chutei: "Errou, o número secreto era 2". Aparecendo essa mensagem com o número secreto fica meio sem graça.

Paulo: Põe aí o 2 só para ver se era mesmo.

Guilherme: Chutei: "Você acertou".

Rafaella: Teria que ficar rodando novamente, mas você pode colocar a mensagem com o número secreto ou não. Vai do que você acha mais legal.

Guilherme: Eu vou tirar e deixar só o "Errou" assim a pessoa vai tentando acertar mesmo.

```
elementoResultado.innerHTML = "Errou";
```

Rafaella: Então agora temos alguns desafios para vocês fazerem em casa! O primeiro desafio que eu gostaria de propor é você colocar um número de tentativas. Nós já vimos que a pessoa pode ficar tentando várias vezes, mas podemos pensar em algo mais difícil para quem for jogar. Como, por exemplo, ter apenas três tentativas e aí na terceira você imprime qual era o valor para a pessoa também saber qual era e qual ela errou. Para fazer isso, você vai ter que aprender um negócio a mais também que veremos nas próximas aulas, mas se você já souber do que eu estou falando, é um desafio interessante.

Paulo: O meu é que na mensagem de erro você mostre por exemplo "Errou, o número secreto é mais do que você está achando" ou "menor do que você está achando". Com isso você está dando uma dica e na próxima tentativa a pessoa já tem algum embasamento

para ir um pouco mais para lá ou um pouco mais para cá. Inclusive, com esse tipo de dica, quem for bom mesmo pode usar até uma estratégia, né? Você tenta chutar no meio, se for menor você pega o meio do meio. Você tenta 5, é menor? Tenta no 2 ou no 3, porque assim você vai pegando. Você já deve ter brincado com algum jogo de carta ou lógica e se futuramente você seguir essa carreira, você vai ver que isso faz parte de um dos principais algoritmos que a Rafa e o Guilherme citaram.

Aliás, isso reforça a minha fala de que a computação e programação são muito próximas àquelas problemas enigmáticos que gostávamos muito quando éramos criança e que às vezes perdemos um pouco desse gosto porque acha que a matemática é muito formal. Então esse jogo Mentalista tem tudo a ver, inclusive, com algumas bases da computação pura, é muito interessante.

Guilherme: Meu desafio para você é o seguinte: a Rafa comentou sobre dois iguais `==`, e três iguais `===` e eu queria que você pesquisasse sobre isso. Qual a diferença de dois iguais, três iguais. Se eu coloco aqui no chute `===` no `numeroSecreto`, o que isso significa? Eu quero que você pesquise sobre isso e descubra essas diferenças.

Paulo: Vou deixar também uma tarefa final para animar porque estou vendo que tem gente que está vindo de tantas carreiras diferentes. Aqui na Alura, montamos uma comunidade muito forte e temos conversado com nossos alunos e alunas sobre as histórias das carreiras dessas pessoas que passaram aqui pela Alura seja pela imersão, seja pelos nossos cursos, seja pelos Bootcamps, seja por diversos lugares em que Alura está envolvida.

Temos um podcast chamado **Scooba Ponto Dev** que vocês encontram no YouTube e também no Spotify com conversas de cinquenta minutos a uma hora com as pessoas que estudaram conosco contando como foi importante essa passagem para suas carreiras. Vocês também podem fazer cada vez mais parte desse universo da Alura, dessa comunidade forte que queremos criar. Como vocês estão vendo no Discord, nós temos essa comunidade em vários lugares. Temos o Discord interno da Alura, os desafios, temos o podcast e grupos de discussão em diversos lugares. Espero que vocês também venham fazer parte disso conosco.

Nos vemos amanhã na aula! Programe, tire suas dúvidas, nos marque, faça aquele vídeo que nos deixa orgulhosos, comece a divulgar seus projetos no LinkedIn para iniciar um portfólio! Nos vemos amanhã.

Rafaella: Até mais, gente!

Guilherme: Tchau tchau!