

## Aula 9

# Senhas complexas

► **Unidade**

**Segurança digital: utilizando  
matemática para programar  
senhas seguras**

# O que vamos aprender?



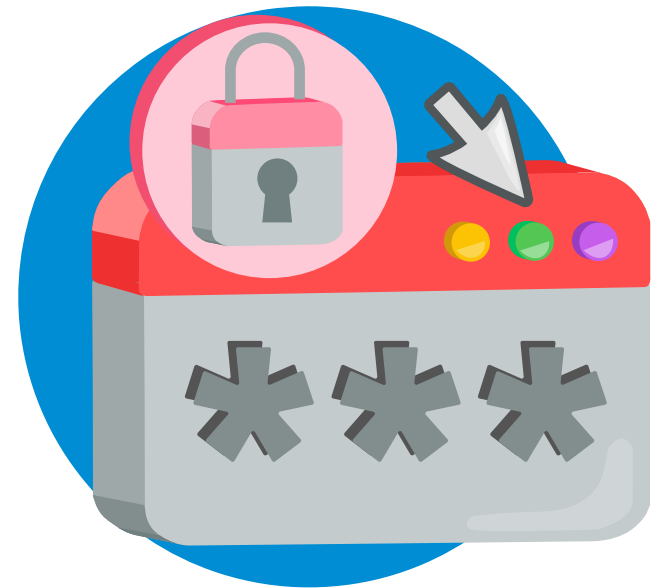
Gerar senhas com letras, números e símbolos aleatórios.



Testar retornos de uma função.



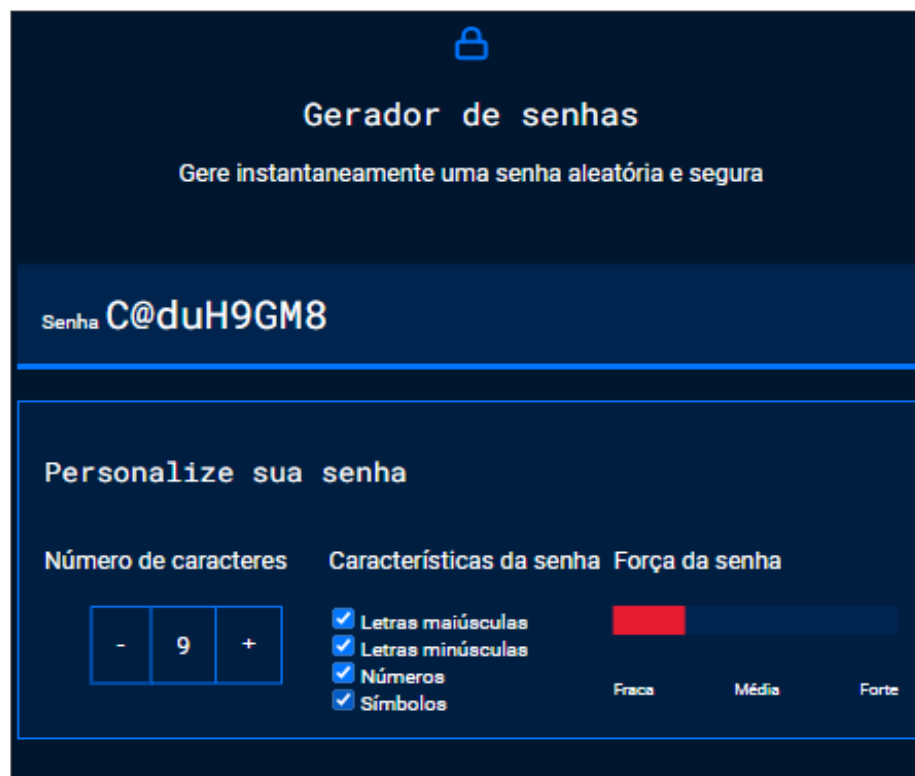
Usar condicionais para concatenar opções marcadas nos checkboxes.



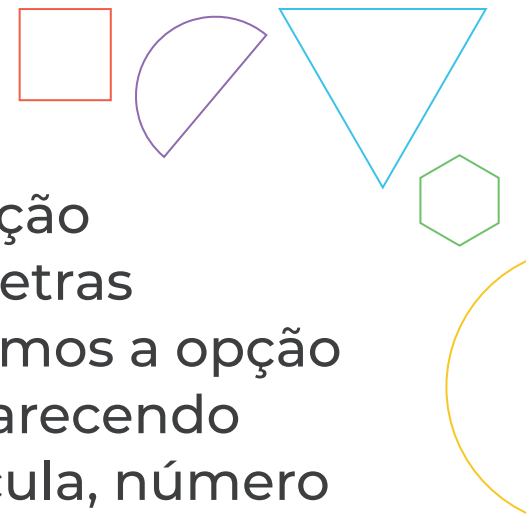
CLIQUE E ACOMPANHE A AULA NA ALURA

# Senhas aleatórias

Na aula anterior, adicionamos a opção de geração de senhas com números aleatórios ao nosso projeto, levando em consideração o número de caracteres da senha a ser gerada. Agora, o objetivo é considerar a aplicação de características dos checkboxes selecionados. Faremos as implementações necessárias utilizando JavaScript e os arquivos HTML e CSS já criados.



The screenshot shows a web application titled "Gerador de senhas" (Password Generator) with a subtitle "Gere instantaneamente uma senha aleatória e segura" (Generate instantly a random and secure password). The interface is dark-themed. A generated password "Senha C@duH9GM8" is displayed. Below, the "Personalize sua senha" (Customize your password) section allows users to adjust settings. It includes a "Número de caracteres" (Number of characters) field with a range from 1 to 9, "Características da senha" (Password characteristics) with four checked checkboxes: "Letras maiúsculas" (Uppercase letters), "Letras minúsculas" (Lowercase letters), "Números" (Numbers), and "Símbolos" (Symbols). A "Força da senha" (Password strength) indicator shows a red bar, indicating a weak password, with labels for "Fraca" (Weak), "Média" (Medium), and "Forte" (Strong).



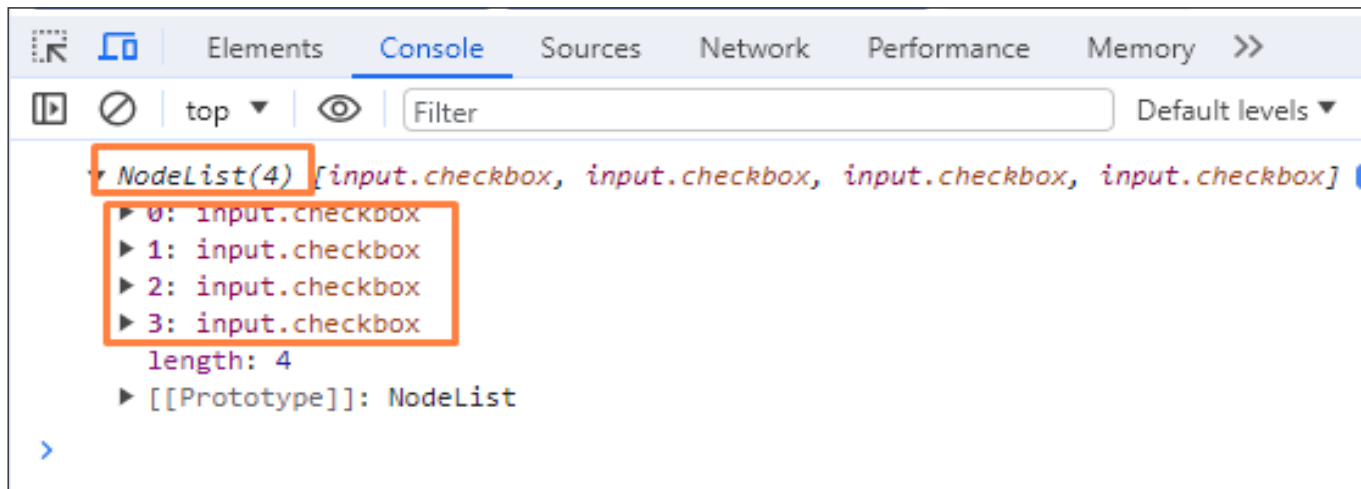
Nesta aula, vamos desenvolver mais um pouco nossa aplicação adicionando as opções dos checkboxes às senhas geradas: letras maiúsculas, minúsculas, números e símbolos. Se desmarcarmos a opção de letra maiúscula, perceberemos que a senha continua aparecendo em caixa-alta e, se selecionarmos as opções de letra minúscula, número ou símbolo, nada acontece. Ou seja, nosso programa está considerando apenas o alfabeto em letras maiúsculas até o momento.

Assim, verificaremos o arquivo *index.html* para localizarmos os itens de **<input>** do tipo checkbox. Neles, existe uma classe chamada **.checkbox** em que podemos usar o DOM para fazer uma consulta e selecionar todos os itens com esse seletor.

Em seguida, no arquivo *main.js*, abaixo de **campoSenha**, usaremos:

```
const checkbox = document.querySelectorAll('.checkbox');  
  
console.log(checkbox);
```

Utilizamos o **console.log(checkbox)** para verificar os itens e sabermos exatamente qual é o elemento do HTML que selecionamos. No navegador, ao clicar no *F12*, selecionaremos a opção *Console*. Observe os inputs dos checkboxes:

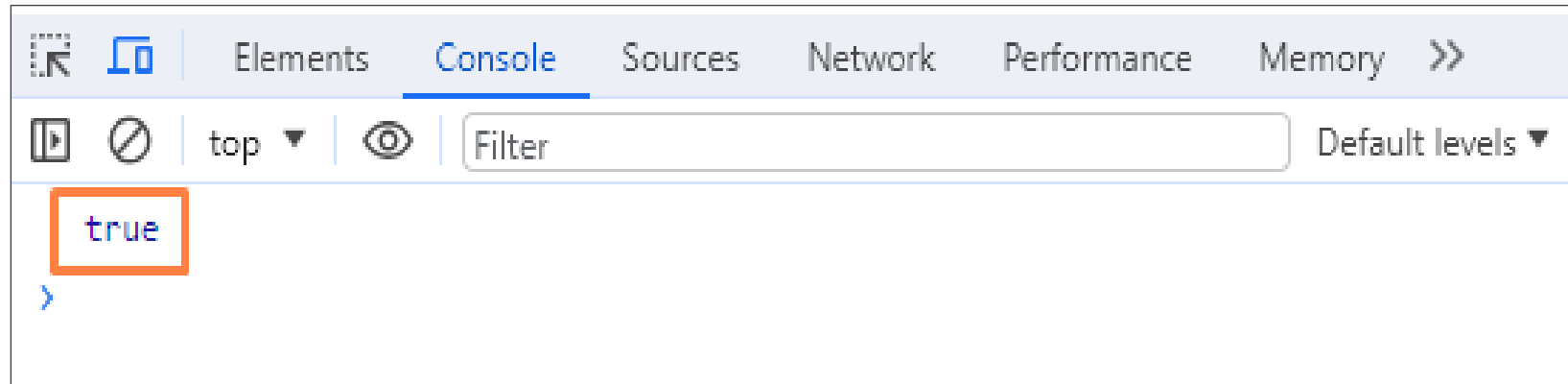


Existem quatro itens, sendo que o primeiro está na posição 0, o segundo, na posição 1 e assim por diante. Eles estão agrupados em uma lista. Agora, precisamos saber quais estão selecionados ou não.

Por padrão, deixaremos o primeiro item sempre marcado em nossa aplicação. Por isso inserimos, anteriormente, o `checked` entre a tag de `<input>`, deixando esse checkbox sempre selecionado ao carregar a página. Agora, adicionaremos essa característica ao **console.log()**:

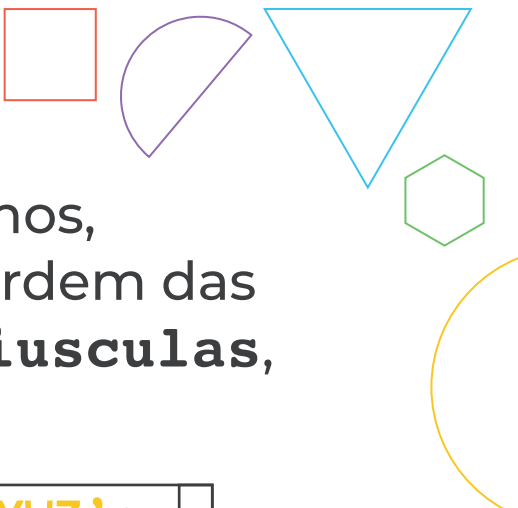
```
console.log(checkbox[0].checked);
```

Com isso, também podemos perguntar ao console se um determinado valor está realmente marcado:



No navegador, perceberemos que essa opção pode retornar *true* (verdadeiro) ou *false* (falso). Nesse caso, retornou *true* porque o primeiro checkbox está realmente marcado. Se fosse o contrário, a resposta obtida seria *false*. Por exemplo, se fosse **checkbox[1].checked**, o retorno no console seria *false*, pois o checkbox dessa posição não está marcado.


A ideia é gerar senhas levando em consideração esses valores. Por exemplo, se sabemos que o primeiro item está marcado, podemos considerar o alfabeto de letras maiúsculas. Agora, se a letra minúscula não está marcada, não podemos considerar as letras minúsculas do alfabeto.



No entanto, ainda não especificamos o alfabeto que desejamos, portanto, iremos criá-lo e indicar para o JavaScript qual é a ordem das letras minúsculas. Desse modo, abaixo de **const letrasMaiusculas**, criaremos as opções representadas nos demais checkboxes:

```
const letrasMaiusculas = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';  
const letrasMinusculas = 'abcdefghijklmnopqrstuvwxyz';  
const numeros = '0123456789';  
const simbolos = '!@%*?';
```

Agora, precisamos verificar quais desses itens estão marcados. Portanto, se o primeiro item estiver marcado, ele considerará apenas o alfabeto de A a Z em letras maiúsculas. Se o primeiro e segundo itens estiverem marcados, além das letras maiúsculas, também considerará as letras minúsculas de A a Z. Neste cenário, precisamos unir esses dois alfabetos. Em outras palavras, precisamos armazenar letras maiúsculas e minúsculas em um único local.




Assim, dentro da função **gerarSenha()**, indicaremos os itens marcados e como podemos organizá-los em um único local. Para isso, criaremos:

```
function gerarSenha(){  
  let alfabeto = '';  
  if (checkbox[0].checked){  
    alfabeto = alfabeto + letrasMaiusculas;  
  }  
  console.log(alfabeto);  
  
  // restante do código omitido  
}
```

Dessa forma, se o checkbox não estiver marcado, nenhuma letra do alfabeto será exibida, pois ficará vazia. Mas, se o checkbox da letra maiúscula estiver marcado, o alfabeto maiúsculo será considerado. Contudo, se testarmos no navegador, clicando no checkbox *Letras maiúsculas*, nenhuma atualização ocorrerá. Isso acontece porque estamos considerando o botão de número de caracteres para atualizar nosso programa. Nesse caso, podemos desmarcar o checkbox *Letras maiúsculas* e pressionar o botão para adicionar um caractere à senha.



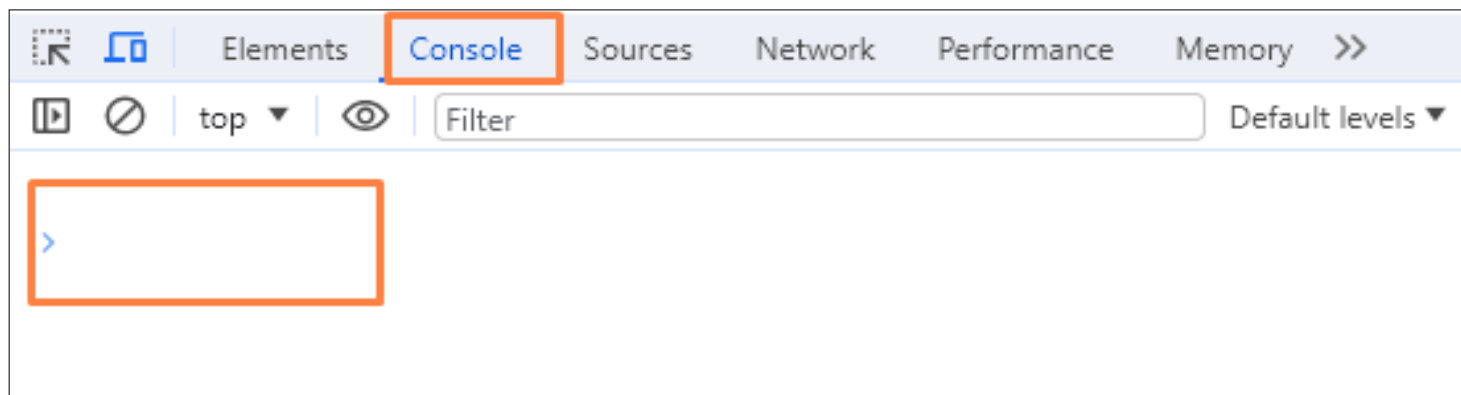


Abaixo de `querySelectorAll( '.checkbox' )`, podemos criar um laço de repetição `for()`. Caso a condição seja verdadeira, vamos adicionar um evento: `checkbox[i].onclick`. Assim, quando o checkbox estiver na posição `i` e também for clicado, queremos que ele execute a função `geraSenha`. Observe:

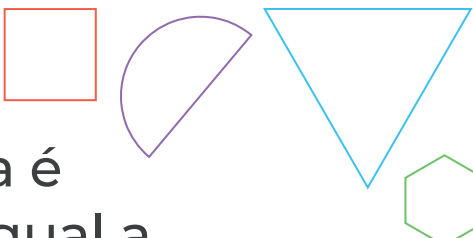
```
const checkbox = document.querySelectorAll( '.checkbox' );  
  
for (i=0; i < checkbox.length;i++){  
    checkbox[i].onclick = geraSenha;
```

Observe que não utilizamos parênteses em `geraSenha`, pois queremos passar um parâmetro para essa função. Caso fosse a nossa intenção executar a função para retorno de algum valor, poderíamos adicionar a pontuação.

No navegador, quando clicamos em qualquer item do checkbox, obtemos uma atualização no console. No entanto, a atualização sempre retorna um conjunto vazio se clicarmos nas opções de letras minúsculas, números e símbolos. Isso porque consideramos apenas o primeiro conjunto do alfabeto, ou seja, as letras maiúsculas. Observe:



Se selecionamos as opções de *Letras maiúsculas* e *Letras minúsculas*, além do alfabeto em letra maiúscula, também deveria aparecer um alfabeto em letra minúscula. Portanto, precisamos programar essa funcionalidade.

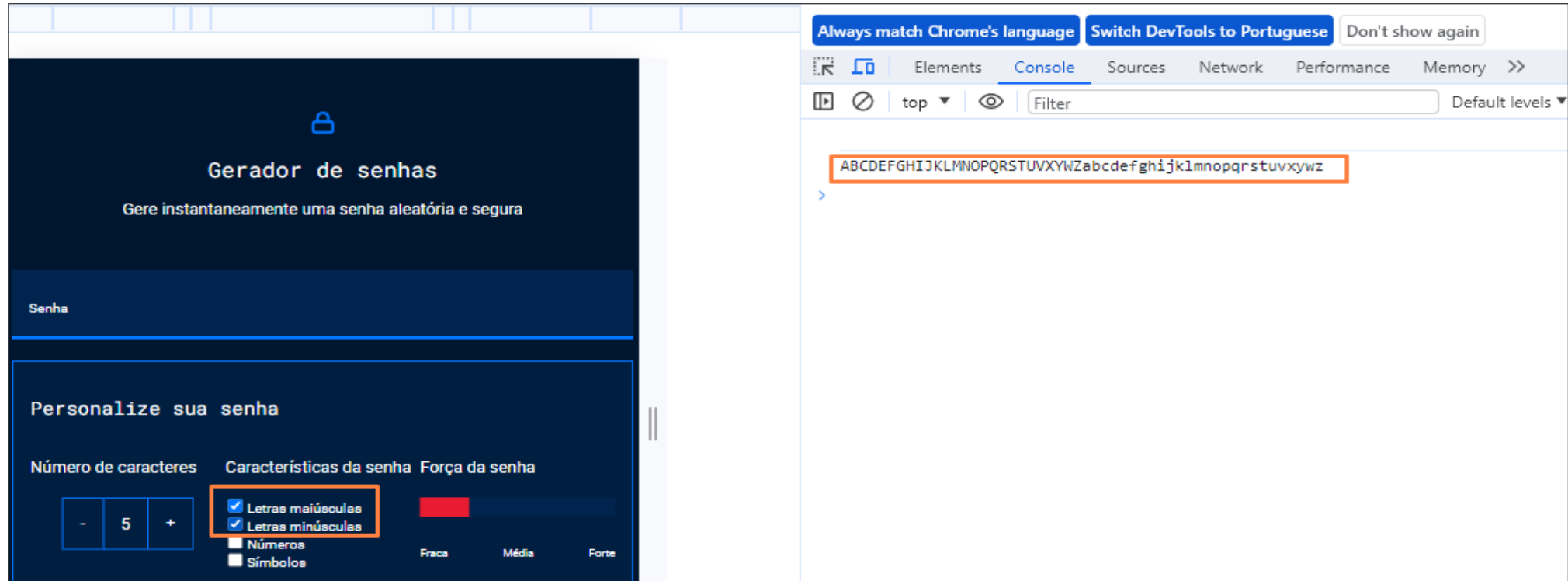


Dentro da função `geraSenha()`, após o primeiro `if`, a lógica é questionar se, em `checkbox[1].checked`, o alfabeto será igual a `alfabeto + letrasMinusculas`. Faremos o mesmo para outros casos e, posteriormente, faremos alguns testes.

```
function geraSenha(){
  let alfabeto = '';
  if (checkbox[0].checked){
    alfabeto = alfabeto + letrasMaiusculas;
  }
  if (checkbox[1].checked){
    alfabeto = alfabeto + letrasMinusculas;
  }
  if (checkbox[2].checked){
    alfabeto = alfabeto + numeros;
  }
  if (checkbox[3].checked){
    alfabeto = alfabeto + simbolos;
  }
  console.log(alfabeto);
  // código omitido...
}
```

Com isso, conseguimos organizar e concatenar todas as possibilidades de alfabeto.

Se voltarmos ao navegador, observaremos no console:



Ao clicar em *Letras minúsculas*, verificaremos que elas são adicionadas ao conjunto das letras maiúsculas. O mesmo ocorre para números e símbolos. Assim, para todas as posições, temos a possibilidade de aumentar ou diminuir o tamanho do nosso alfabeto. Caso desmarquemos todas as opções e deixemos selecionados apenas as letras maiúsculas e os símbolos, o alfabeto levará em consideração apenas esses itens.

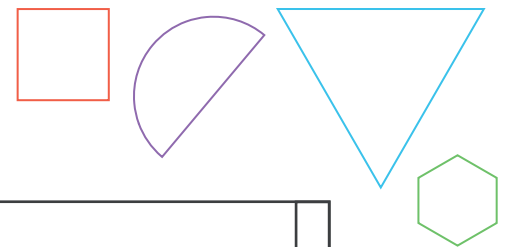


No entanto, a nossa senha ainda não está gerando senhas com símbolos, porque ainda não estamos considerando o alfabeto completo para sortear a letra.

Assim, para refinar nosso código, vamos a **for** dentro da função **geraSenha()**.

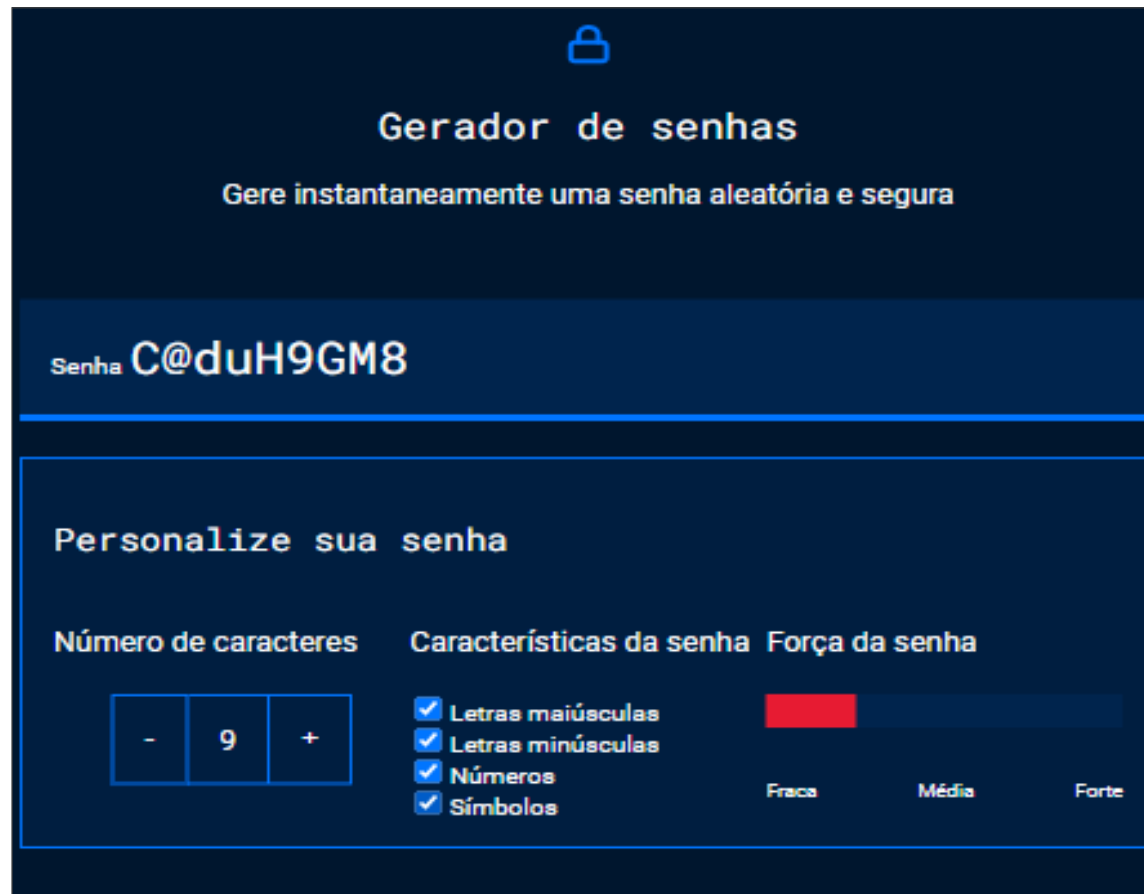
Em **let numeroAleatorio**, precisamos substituir **letrasMaiusculas** pela variável **alfabeto**. Com isso, podemos testar todas as possibilidades do alfabeto, ou seja, ele irá gerar um número aleatório que pode ir desde zero até um número, dependendo de quantos checkboxes marcamos.

Além disso, também precisamos selecionar o alfabeto na posição do **numeroAleatorio**, ou seja, é preciso transformar aquele número que sortearmos em uma letra. Por isso, vamos reescrever que **senha** deve ser igual a **senha + alfabeto[numeroAleatorio]**.



```
function geraSenha(){
    let alfabeto = '';
    if (checkbox[0].checked){
        alfabeto = alfabeto + letrasMaiusculas;
    }
    if (checkbox[1].checked){
        alfabeto = alfabeto + letrasMinusculas;
    }
    if (checkbox[2].checked){
        alfabeto = alfabeto + numeros;
    }
    if (checkbox[3].checked){
        alfabeto = alfabeto + simbolos;
    }
    console.log(alfabeto);
    let senha = '';
    for (let i = 0; i < tamanhoSenha;i++){
        let numeroAleatorio = Math.random()*alfabeto.length;
        numeroAleatorio = Math.floor(numeroAleatorio);
        senha = senha + alfabeto[numeroAleatorio];
    }
    campoSenha.value = senha;
}
```

Portanto, a expectativa é que, de fato, consigamos gerar uma senha. Para testar essa funcionalidade, vamos ao navegador:



The image shows a web application interface for a password generator. At the top, there is a blue padlock icon, the title "Gerador de senhas", and the subtitle "Gere instantaneamente uma senha aleatória e segura". Below this, a generated password "Senha C@duH9GM8" is displayed. The main section is titled "Personalize sua senha" and contains three columns: "Número de caracteres" with a numeric input set to 9, "Características da senha" with four checked checkboxes for uppercase letters, lowercase letters, numbers, and symbols, and "Força da senha" with a red progress bar and labels for "Fraca", "Média", and "Forte".

**Gerador de senhas**  
Gere instantaneamente uma senha aleatória e segura

Senha **C@duH9GM8**

**Personalize sua senha**

Número de caracteres	Características da senha	Força da senha
- 9 +	<input checked="" type="checkbox"/> Letras maiúsculas <input checked="" type="checkbox"/> Letras minúsculas <input checked="" type="checkbox"/> Números <input checked="" type="checkbox"/> Símbolos	<div><div></div></div> Fraca Média Forte



Por padrão, todas as letras da senha estão em maiúsculas. Ao aumentar o número de caracteres da senha para 9, percebemos que ela foi criada corretamente. Ao considerar também letras minúsculas, algumas letras aparecem dessa forma. Ainda, ao adicionarmos um número, conseguimos uma senha com letras maiúsculas, minúsculas e com números. Por fim, ao considerarmos símbolos, teremos o @, na segunda posição, por exemplo. Portanto, toda vez que clicarmos nos botões, uma senha será automaticamente gerada de acordo com os parâmetros estabelecidos.

Na próxima aula, refinaremos o nosso código e o incrementaremos com a classificação de senha. Até breve!



## ► Desafio

Nesta aula, aprendemos como gerar senhas aleatórias conforme as especificações que selecionamos nos checkboxes. Seu desafio será pesquisar sobre criptografia de senhas, buscando descobrir quais tipos existem. A partir da pesquisa feita, procure inserir mais um checkbox com uma nova característica de senha e faça-os funcionar de acordo com o que aprendemos nesta aula.



CLIQUE **AQUI** PARA AVALIAR ESTE MATERIAL