




Aula 8

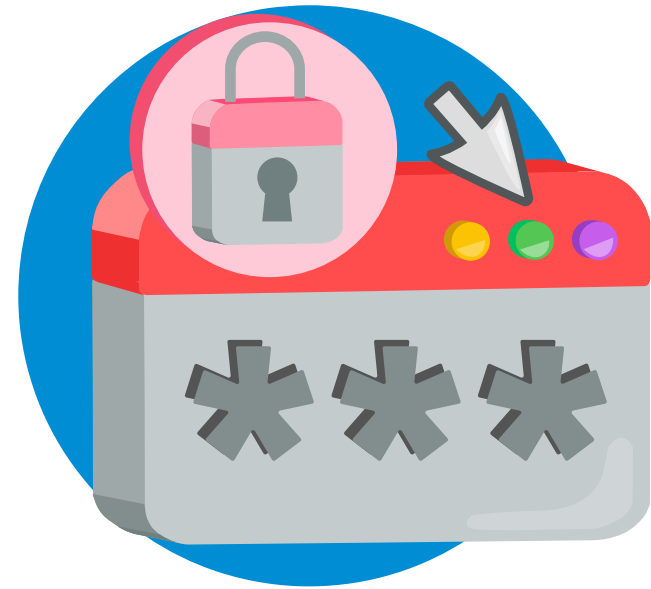
Gerando senhas

► **Unidade**

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

O que vamos aprender?

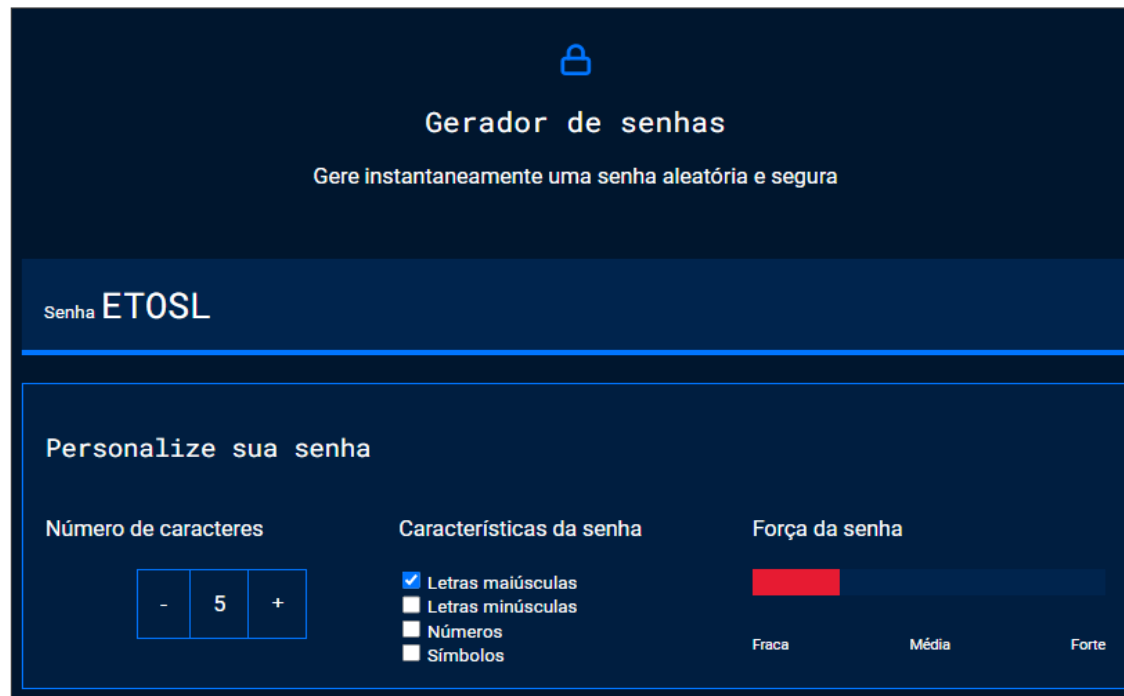
-  Gerar letras e números aleatórios.
-  Ajustar o campo senha.
-  Utilizar um laço de repetição para automatizar o código.




 CLIQUE E ACOMPANHE A AULA NA ALURA

JavaScript e números aleatórios

Na aula anterior, implementamos a funcionalidade para o botão de diminuir (-) e de aumentar (+) o número de caracteres da senha gerada mediante uma função e um evento. Nosso objetivo nesta aula é continuar tornando nossa página mais dinâmica, adicionando uma opção de geração de senhas com números aleatórios e gerando uma senha segura.



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 password "ET0SL" is displayed in a light blue box. Below it, a section titled "Personalize sua senha" (Customize your password) contains three columns: "Número de caracteres" (Number of characters) with buttons for "-", "5", and "+"; "Características da senha" (Password characteristics) with checkboxes for "Letras maiúsculas" (checked), "Letras minúsculas", "Números", and "Símbolos"; and "Força da senha" (Password strength) with a red progress bar and labels "Frac", "Média", and "Forte".



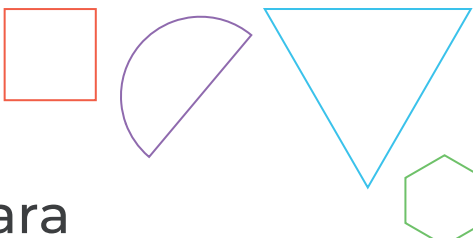
Nesta aula, vamos desenvolver mais um pouco da nossa aplicação adicionando um conjunto de caracteres, todos em letras maiúsculas, ao campo senha.

A ideia é que o campo senha faça a geração desses caracteres de forma aleatória e automática, ou seja, o usuário não escreverá nada. Sendo assim, precisamos, primeiramente, bloquear a funcionalidade da escrita nesse campo.

Para isso, acessaremos o arquivo *index.html*. No trecho de **<input>** **campo-senha**, adicionaremos um comando chamado **readonly**, para que o usuário não consiga modificar o campo senha manualmente. Observe:

```
<div class="conteudo-senha">  
  <label for="senha">Senha</label>  
  <input name="senha" type="text" id="campo-senha"  
  readonly>  
</div>
```

Agora, sabemos que o **campo-senha** será o responsável por gerar o resultado da senha e está bloqueado para alterações manuais. Portanto, já podemos começar a programar no arquivo *main.js*.




Uma vez que, anteriormente, empregamos **console.log** para solucionar os problemas com os botões, agora podemos eliminá-lo e criar uma constante chamada **campoSenha**, utilizando **document.querySelector(' ')**. Seu código ficará da seguinte forma:

```
// código omitido  
  
const campoSenha = document.querySelector( '#campo-senha' );
```

Dessa forma, utilizaremos o mesmo comando que foi usado no seletor para facilitar o uso no CSS. Observe que, para o id, adicionaremos um jogo da velha (#). Se for uma classe, adicionaremos um ponto-final (.). Como já temos o **campoSenha**, podemos escrever algo nele, como um alfabeto maiúsculo. Consequentemente, precisamos adicionar **const letrasMaiusculas**, com todas as letras maiúsculas do alfabeto, entre aspas simples. Você pode fazer isso com a tecla *Caps Lock* ativada. Observe:

```
// código omitido  
  
const campoSenha = document.querySelector( '#campo-senha' );  
  
const letrasMaiusculas = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
```



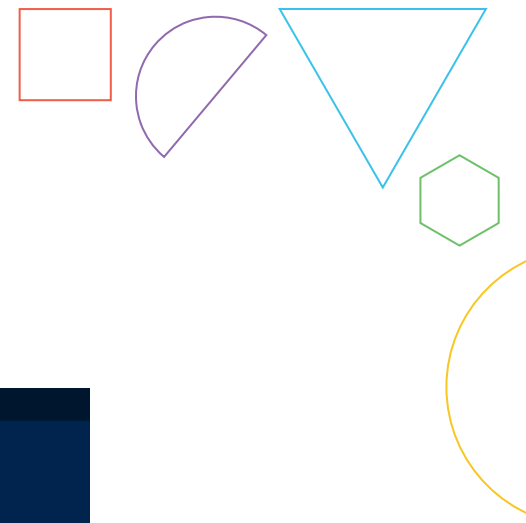
Se desejamos escrever todo esse alfabeto maiúsculo no **campoSenha**, podemos fazer **campoSenha.textContent** igual a **letrasMaiusculas**, produzindo uma correspondência entre o campo senha e as letras maiúsculas inseridas, como mostrado a seguir:

```
// código omitido

const campoSenha = document.querySelector('#campo-senha');
const letrasMaiusculas = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
campoSenha.textContent = letrasMaiusculas;
```

Observe que, se atualizarmos a aba do navegador, não conseguiremos mais editar o campo senha devido ao **readonly**, mas a senha ainda não foi gerada. Isso ocorre porque o comando utilizado dentro do VSCode não deve ser o **textContent**. Nesse caso, devemos substituir o nome da palavra por **value**. Seu código ficará da seguinte forma:


```
campoSenha.value = letrasMaiusculas;
```



Assim, ao atualizar o navegador, teremos:



Desse modo, podemos identificar que, no campo senha, aparecem todas as letras do alfabeto. No entanto, a senha deve ter até 12 caracteres, ou seja, não pode conter todo o alfabeto. Além disso, ela precisa ser aleatória.



Vamos começar a organizar o processo de geração aleatória. Imagine um número aleatório que começa na posição 0 da senha, ou seja, o primeiro caractere. Ele pode ser tanto a letra A, que também está na posição 0 (inicial) do nosso alfabeto, até o último valor, ou seja, a letra Z. Portanto, temos 26 possibilidades diferentes para o primeiro caractere da senha a ser gerada. O seja, tamanho da lista é de 26 itens.

Desse modo, necessitamos criar números aleatórios que sejam capazes de gerar valores entre 0 e 26. Assim, vamos criar uma função, **geraSenha()**, para iniciar a geração de números aleatórios da nossa senha. Seu código ficará da seguinte forma:

```
// código omitido

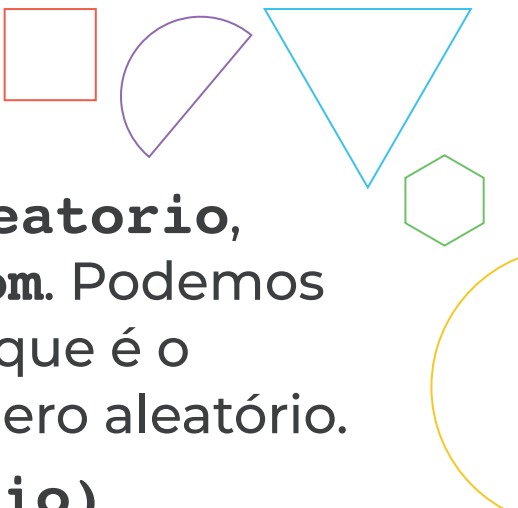
const campoSenha = document.querySelector('#campo-senha');

const letrasMaiusculas = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';

function geraSenha(){

}

campoSenha.value = letrasMaiusculas;
```

Nesse contexto, criaremos uma variável chamada **numeroAleatorio**, que deve variar de 0 a 26, utilizando o comando **Math.random**. Podemos multiplicar essa variável por ***letrasMaiusculas.length**, que é o comprimento da frase. Assim, conseguiremos gerar um número aleatório. Por fim, adicionaremos um **console.log(numeroAleatorio)** para visualizar precisamente esse número e chamaremos a função **geraSenha()**. Seu código ficará da seguinte forma:

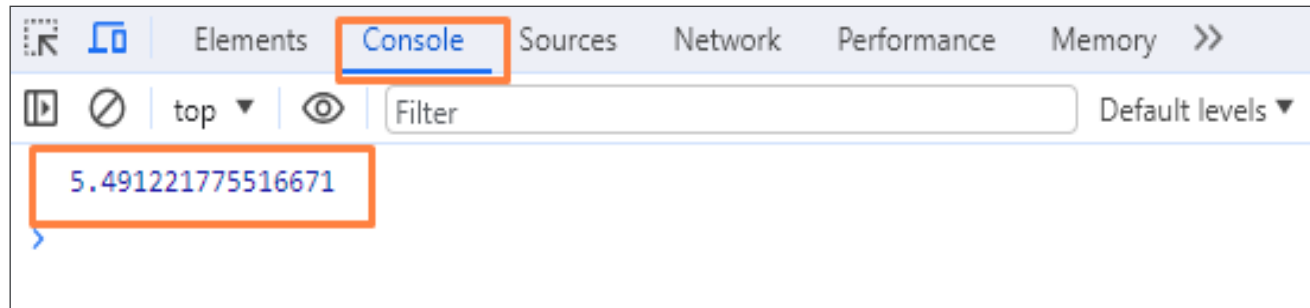
```
const campoSenha = document.querySelector('#campo-senha');

const letrasMaiusculas = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
geraSenha();

function geraSenha(){
  let numeroAleatorio = Math.random()*letrasMaiusculas.length;
  console.log(numeroAleatorio);
}

campoSenha.value = letrasMaiusculas;
```

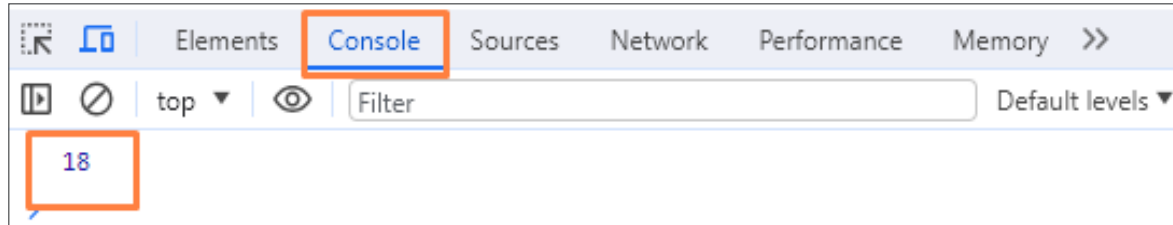
Assim, quando chamarmos a função **geraSenha()**, um número aleatório será gerado. Se acessarmos o console no navegador (pressionando *F12*), poderemos observar que geramos um número:



No entanto, esse número é muito extenso. Podemos arredondá-lo depois do ponto decimal utilizando o comando **Math.floor**. Portanto, após o **let numeroAleatorio**, o resultado será o **Math.floor()** do número aleatório. Veja:

```
function geraSenha(){  
  let numeroAleatorio = Math.random()*letrasMaiusculas.length;  
  numeroAleatorio = Math.floor(numeroAleatorio);  
}
```

Após atualizar a página, teremos:



Observe que, agora, não há mais casas decimais do lado direito no console, apenas números aleatórios.

Portanto, agora somos capazes de gerar um número aleatório. No entanto, a intenção não é apenas essa, mas sim selecionar um número aleatório para exibir a letra correspondente nesse mesmo índice, pois se, por exemplo, a posição da letra for 1, sabemos que ela deve corresponder à letra B. Nesse caso, a ideia não é exibir o número aleatório, mas sim a letra de **letrasMaiusculas**, na posição do número aleatório.



Por isso, no `console.log()` aplicaremos `letrasMaiusculas[numeroAleatorio]`, como mostrado a seguir:

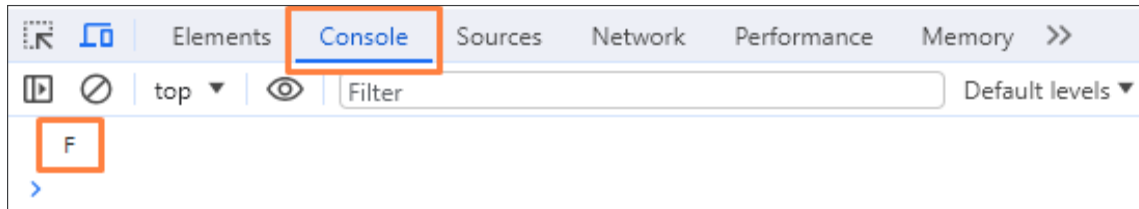
```
// código omitido
const campoSenha = document.querySelector('#campo-senha');

const letrasMaiusculas = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
geraSenha();

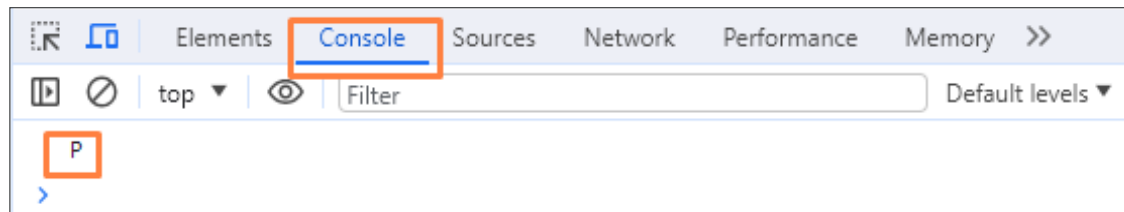
function geraSenha(){
  let numeroAleatorio = Math.random()*letrasMaiusculas.length;
  numeroAleatorio = Math.floor(numeroAleatorio);
  console.log(letrasMaiusculas[numeroAleatorio]);
}

campoSenha.value = letrasMaiusculas;
```

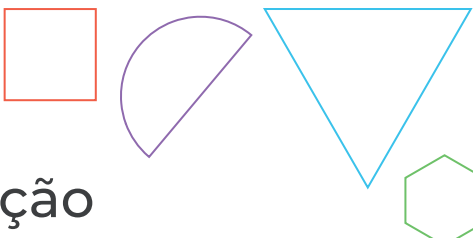
Se voltarmos à aba do navegador, notamos que foi gerada uma letra do lado direito.



Ao atualizarmos a página, será gerada uma nova letra. Observe que, toda vez que atualizarmos a página, uma nova letra será gerada, pois um novo número diferente também será gerado.



Entretanto, nosso objetivo é gerar 12 letras aleatórias e tê-las todas juntas para formar a senha, em vez de exibir o alfabeto inteiro. Dessa forma, precisamos automatizar esse processo de repetição 12 vezes.



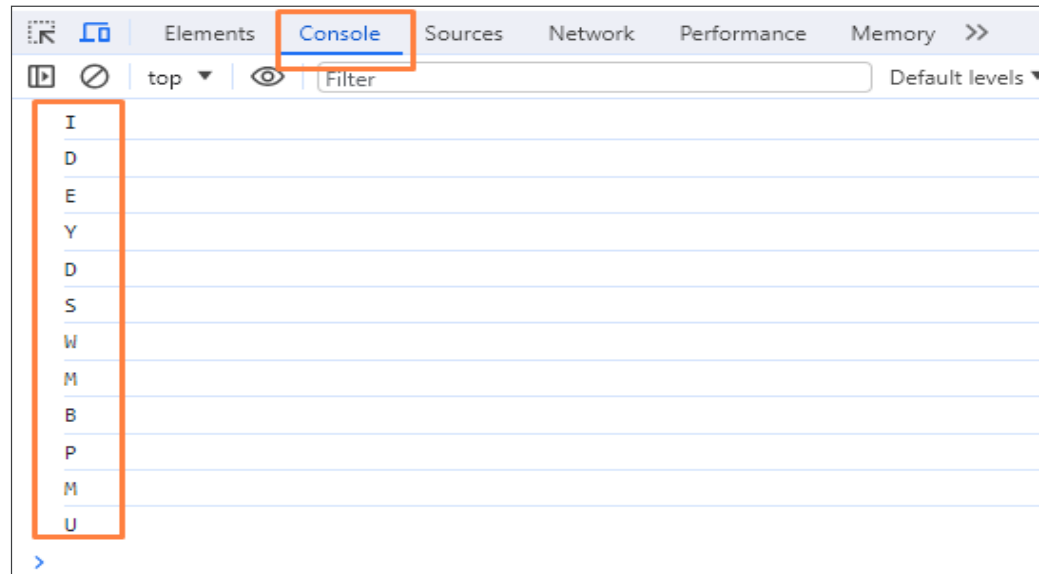
Para isso, é possível usar um laço de repetição dentro da função **geraSenha()**. Podemos adicionar **for()**, abrindo e fechando parênteses, e definir **let i = 0**. Enquanto **i** for menor do que o número de caracteres definidos, podemos continuar. Portanto, temos **tamanhoSenha; i++**. Agora sim, é possível gerar diversos valores. Logo, **let numeroAleatorio** pode selecionar, arredondar e mostrar uma letra no alfabeto. Então, passamos essas informações para dentro de **for()** **{}**. Seu código ficará da seguinte forma:

```
// código omitido
const campoSenha = document.querySelector('#campo-senha');


const letrasMaiusculas = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
geraSenha();

function geraSenha(){
  for (let i = 0; i < tamanhoSenha; i++) {
    let numeroAleatorio = Math.random()*letrasMaiusculas.length;
    numeroAleatorio = Math.floor(numeroAleatorio);
    console.log(letrasMaiusculas[numeroAleatorio]);
  }
}
campoSenha.value = letrasMaiusculas;
```

Agora, ao repetirmos esse processo 12 vezes, podemos verificar que, na aba do navegador, são geradas não apenas uma, mas sim 12 letras aleatórias. Observe:

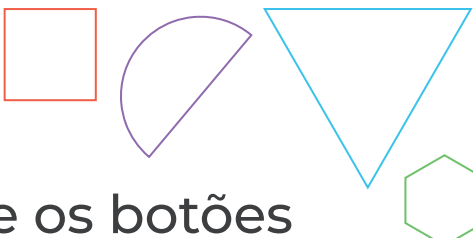


Ainda temos algumas questões de implementação que precisamos resolver. Para testarmos novamente nosso projeto, no VS Code, vamos alterar o tamanho da senha para 5 caracteres, para que tenhamos uma senha pequena, por enquanto. Se acessarmos o navegador, conseguiremos perceber que foram geradas apenas cinco letras no console.



Porém, ao clicarmos no botão para aumentar ou diminuir o número de caracteres da senha, nosso projeto não gera uma senha nova. Isso ocorre porque a função que criamos é executada apenas uma vez, e não a cada evento de clique. Sendo assim, também precisamos incluir a função **gerarSenha()** nos eventos, chamando-a para diminuir e aumentar a senha. O código ficará assim:

```
function diminuiTamanho(){
    if (tamanhoSenha > 1){
        // tamanhoSenha = tamanhoSenha-1;
        tamanhoSenha--;
    }
    numeroSenha.textContent = tamanhoSenha;
    gerarSenha();
}
function aumentaTamanho(){
    if (tamanhoSenha < 20){
        // tamanhoSenha = tamanhoSenha+1;
        tamanhoSenha++;
    }
    numeroSenha.textContent = tamanhoSenha;
    gerarSenha();
}
```





Podemos testar o projeto no navegador. Identificaremos que os botões estão funcionando corretamente, tanto para acrescentar quanto para diminuir caracteres no console.

Agora, podemos aprimorar um pouco o código encontrando letras aleatórias e unindo-as. Neste caso, necessitamos concatenar as letras sorteadas. Para isso, na função **gerarSenha()**, precisamos armazenar as informações em uma variável, **let senha**, que inicialmente, estará vazia. Observe:

```
// código omitido
function gerarSenha(){
  let senha = '';
  for (let i = 0; i < tamanhoSenha;i++) {
    let numeroAleatorio = Math.random()*letrasMaiusculas
length;
    numeroAleatorio = Math.floor(numeroAleatorio);
    console.log(letrasMaiusculas[numeroAleatorio]);
  }
}

campoSenha.value = letrasMaiusculas;
```



A cada interação, depois de sortear um número e adicioná-lo, podemos informar a senha que está vazia na primeira interação (**senha = senha**). Isso deve ser feito após o arredondamento do número aleatório pelo comando **Math.floor**.

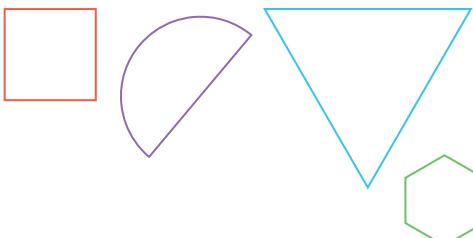
No entanto, agora ela receberá mais um valor, que é a letra maiúscula na posição do número aleatório: **+ letrasMaiusculas[numeroAleatorio]**.

```
function geraSenha(){
  let senha = '';
  for (let i = 0; i < tamanhoSenha;i++){
    let numeroAleatorio = Math.random()*letrasMaiusculas.length;
    numeroAleatorio = Math.floor(numeroAleatorio);
    senha = senha + letrasMaiusculas[numeroAleatorio];
    console.log(letrasMaiusculas[numeroAleatorio]);
  }
  campoSenha.value = letrasMaiusculas;
}
```



A cada interação, o código vai trazendo o anterior e concatenando os seguintes, ou seja, unindo as letras.

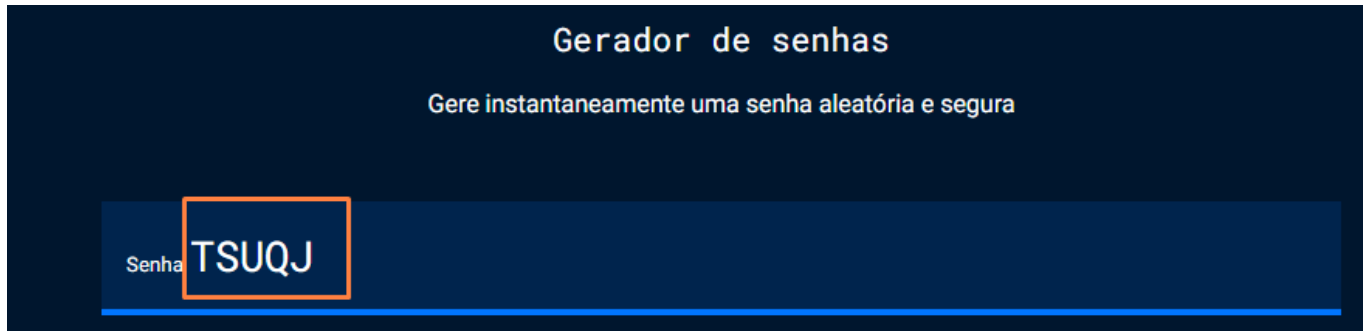
Agora, não vamos mais usar **console.log**, porque já sabemos que a aleatoriedade das letras está funcionando. Desse modo, podemos apagar esse comando. Desejamos mostrar essa senha no **campoSenha**, portanto, vamos remover **letrasMaiusculas** desse campo, já que não desejamos que todo o alfabeto seja mostrado nele.



Também precisamos prestar atenção na chave. A chave amarela está encerrando nossa função, então, o campo senha precisa estar dentro de nossa função **gerarSenha**.

```
function gerarSenha(){  
  let senha = '';  
  for (let i = 0; i < tamanhoSenha;i++){  
    let numeroAleatorio = Math.random()*letrasMaiusculas  
length;  
    numeroAleatorio = Math.floor(numeroAleatorio);  
    senha = senha + letrasMaiusculas[numeroAleatorio];  
  }  
  campoSenha.value = senha;  
}
```

Se visualizarmos no navegador, veremos que nossa senha está funcionando:



Na próxima aula, adicionaremos os ajustes necessários para que, ao clicarmos em *Letras maiúsculas* e no campo de *Características da senha*, o valor mude, levando em consideração o campo da caixa de seleção. Até breve!

► Desafio

Nesta aula, aprendemos como gerar letras conforme números aleatórios para o campo senha. Ajustamos algumas partes do código para gerar a quantidade desejada de letras maiúsculas. Seu desafio será refazer o mesmo código, porém, fazendo com que as opções de senha sejam letras minúsculas e maiúsculas, ou seja, estejam misturadas.



CLIQUE **AQUI** PARA AVALIAR ESTE MATERIAL