




## Aula 7

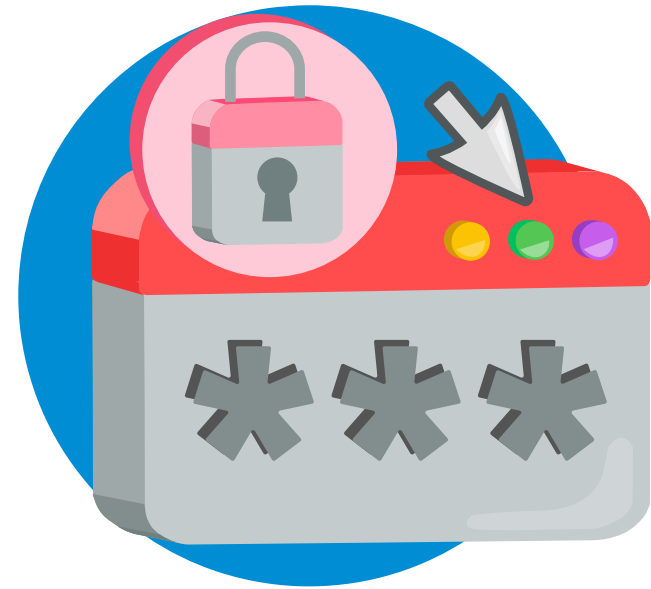
# Número de caracteres

► **Unidade**

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

# O que vamos aprender?

-  Trazer dinamismo para a página desenvolvida.
-  Ajustar o comprimento da senha.
-  Criar uma função para o botão de diminuir (-) e para o botão de aumentar (+).



▶ CLIQUE E ACOMPANHE A AULA NA ALURA

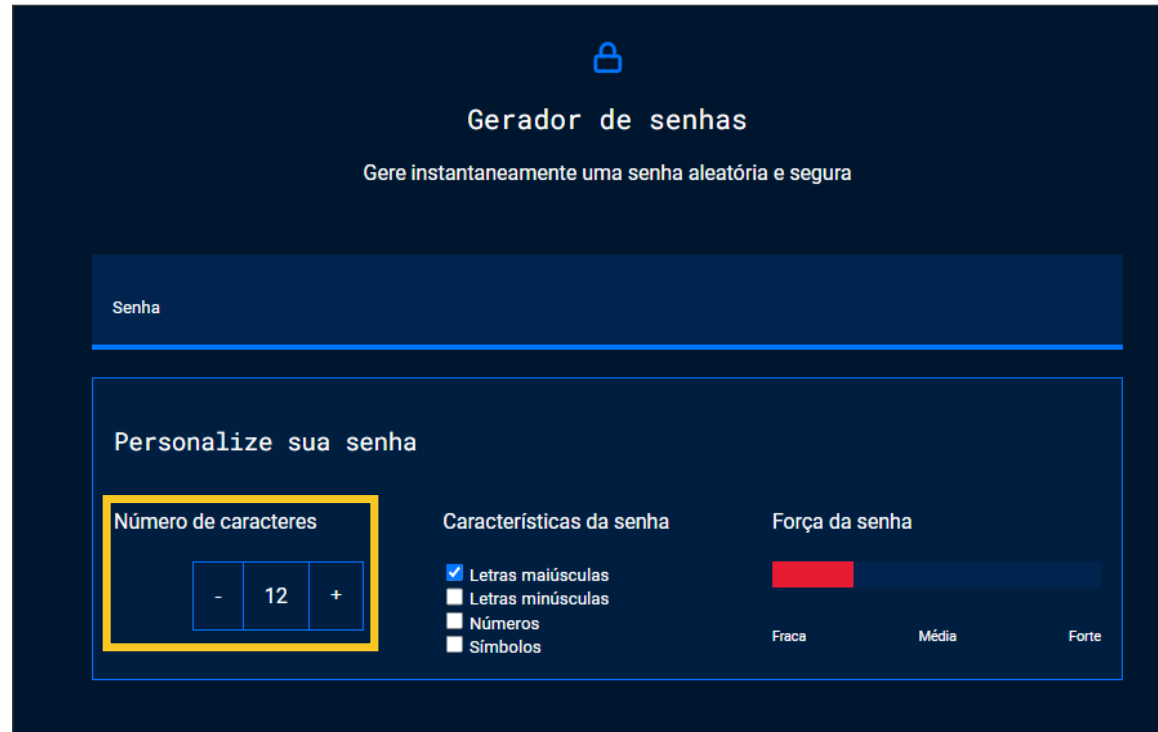
# Diminuindo o tamanho da senha

Na aula anterior, efetuamos a inserção da barra de força da senha e a estilizamos. Agora, o objetivo é tornar nossa página mais dinâmica. Sabemos que há algumas implementações que desejamos fazer para gerar uma senha segura. Faremos isso utilizando JavaScript e os arquivos já criados em HTML e CSS.

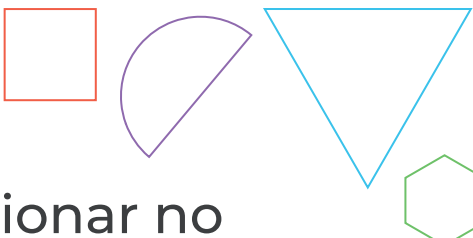


The screenshot shows a web application titled "Gerador de senhas" (Password Generator) with a subtitle "Gere instantaneamente uma senha aleatória e segura" (Generate instantly a random and secure password). The interface includes a large input field for the generated password. Below it, a section titled "Personalize sua senha" (Customize your password) allows users to adjust settings. On the left, "Número de caracteres" (Number of characters) is set to 12, with minus and plus buttons for adjustment. In the center, "Características da senha" (Password characteristics) includes four checkboxes: "Letras maiúsculas" (checked), "Letras minúsculas", "Números", and "Símbolos". On the right, "Força da senha" (Password strength) is shown with a red progress bar and three labels: "Fraca" (Weak), "Média" (Medium), and "Forte" (Strong).

Nesta aula, vamos desenvolver mais um pouco da nossa aplicação adicionando dinâmizações. Além disso, vamos dar os primeiros passos no JavaScript. Ao visualizar o projeto, podemos verificar que, na parte inferior do lado esquerdo, temos a área de *Número de caracteres* da senha. Gostaríamos que, ao clicar no botão menos (-), o número 12 diminuísse de um em um, ou seja, para 11, 10, 9, 8 e, assim, sucessivamente. O mesmo deveria ocorrer com o botão mais (+), porém, ao clicar, o botão deverá aumentar esse valor de um em um, até um limite estabelecido.





The screenshot shows a dark-themed web application titled "Gerador de senhas" (Password Generator). Below the title is a subtitle: "Gere instantaneamente uma senha aleatória e segura" (Generate instantly a random and secure password). There is a text input field labeled "Senha". Below this is a section titled "Personalize sua senha" (Customize your password). This section contains three main controls: 1. "Número de caracteres" (Number of characters): A box containing a minus sign (-), the number 12, and a plus sign (+). This box is highlighted with a yellow border. 2. "Características da senha" (Password characteristics): A list of checkboxes for "Letras maiúsculas" (checked), "Letras minúsculas", "Números", and "Símbolos". 3. "Força da senha" (Password strength): A red progress bar and three labels: "Fraca", "Média", and "Forte".



Dessa forma, temos alguns elementos que precisamos selecionar no HTML e manipularmos no JavaScript. Iniciaremos com o arquivo *index.html* para entendermos onde estão esses elementos e quais deles precisamos acessar usando o DOM. Perceberemos que existem três elementos. Um elemento do botão, com a classe **parametro-senha\_\_botao**, e o sinal de menos. Há outra classe, **parâmetro-senha\_\_botao**, que adiciona, ou seja, está escrita com o sinal de mais. Por fim, também temos um texto expressando o valor 12 que mantivemos como padrão. Observe:

```
<div class="parametro">
  <h3 class="parametro-titulo">Personalize sua senha</h3>
  <div class="parametro-coluna__senha">
    <div class="parametro-senha">
      <h4 class="parametro-senha__titulo">Número de
caracteres</h4>
      <div class="parametro-senha-botoes">
        <button class="parametro-senha__botao">-</button>
        <p class="parametro-senha__texto">12</p>
        <button class="parametro-senha__
botao">+</button>
      </div>
    </div>
  </div>
```

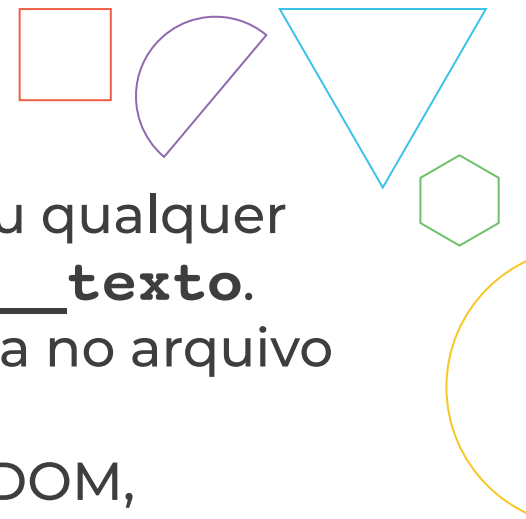


A ideia é acessar exatamente esses elementos. Para fazer isso, precisamos abrir um arquivo que contenha apenas códigos em JavaScript. Assim, na parte superior esquerda do VSCode, no topo da aba do explorador de arquivos, clicaremos no ícone *Novo Arquivo*  e criaremos um arquivo chamado *main.js*. Ao criarmos esse arquivo, ele aparecerá vazio, pronto para ser programado.

Contudo, precisamos referenciar esse arquivo *main.js* no *index.html*. Isso será feito acima do fechamento do **<body>** e abaixo do fechamento da **<section>**. Dessa forma, criaremos uma tag chamada **<script>** e, dentro dela, adicionaremos um atributo chamado **src**, que significa source (fonte). Nele, adicionaremos o arquivo criado, *main.js*, entre aspas duplas. Seu código ficará da seguinte forma:

```
</section>
  <script src="main.js"></script>
</body>
</html>
```

Com isso, os arquivos *index.html* e *main.js* estão se comunicando, permitindo-nos acessar elementos desse último diretamente no arquivo HTML.




Sabemos que, se quisermos alterar o texto 12 para 11, 15, 18 ou qualquer outro valor, precisamos acessar a classe **parâmetro-senha\_\_texto**. Assim, vamos copiar essa classe para que possamos utilizá-la no arquivo *main.js*, no DOM.

Na primeira linha, escreveremos uma palavra reservada do DOM, chamada **document**, e efetuaremos uma consulta nesse DOM, ou seja, adicionaremos um **.querySelector()**. Entre os parênteses desse seletor, abriremos aspas simples e usaremos o mesmo seletor que está no arquivo *style.css*. Seu código, até o momento, ficará assim:

```
document.querySelector('.parametro-senha__botao');
```

Agora já podemos armazenar essa informação em uma variável; observe:

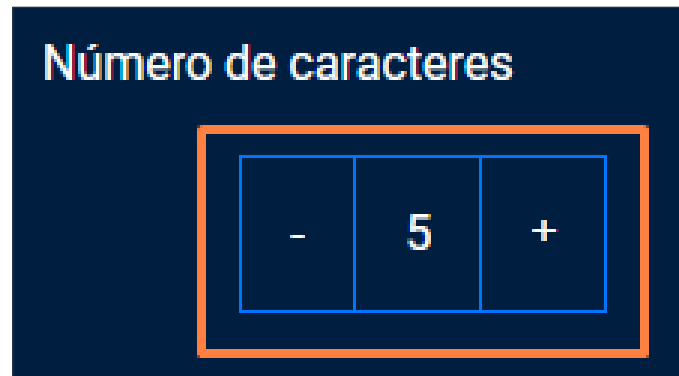
```
const numeroSenha = document.querySelector('.parametro-senha__texto');
```



Agora, se quisermos alterar o número 12, podemos acessá-lo por meio de **numeroSenha.textContent**, em que **textContent** é o conteúdo do texto. Assim, podemos reatribuir a ele um novo valor, adicionando o sinal de igual. Colocaremos 5, por exemplo. Seu código ficará da seguinte forma:


```
const numeroSenha = document.querySelector('.parametro-senha__texto');  
numeroSenha.textContent = 5;
```

Ao salvar o código e atualizar nossa página, teremos:



Perceba que, em vez do número 12, agora temos 5. Em outras palavras, conseguimos alterar esse valor usando o *main.js*.





Porém, em vez de mudar esse número diretamente no arquivo *main.js*, queremos utilizar os botões de diminuir ou de aumentar presentes na página. Para isso, também é necessário selecionar elementos dos botões e fazer uma consulta utilizando o **querySelector** para esses dois elementos. Observe:

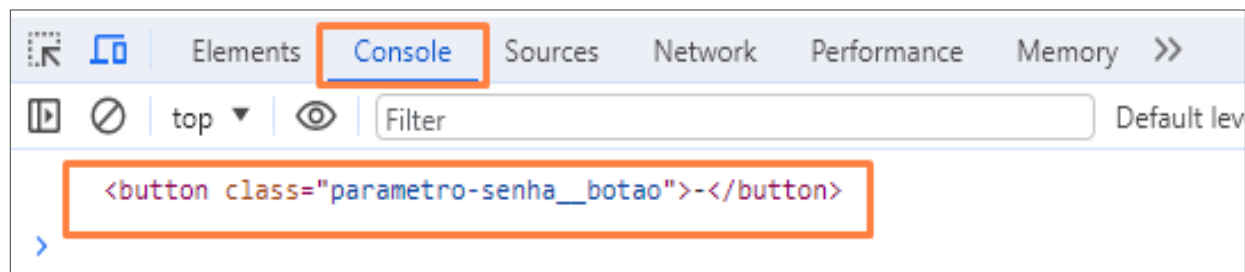
```
const botoes = document.querySelector('.parametro-senha__botao');
```

Acessando esse botão, perceberemos que o **querySelector** só consegue selecionar um elemento. Portanto, se existir uma classe que estiliza apenas um elemento, o comando funcionará. Ou seja, o **querySelector** selecionará apenas o primeiro botão.

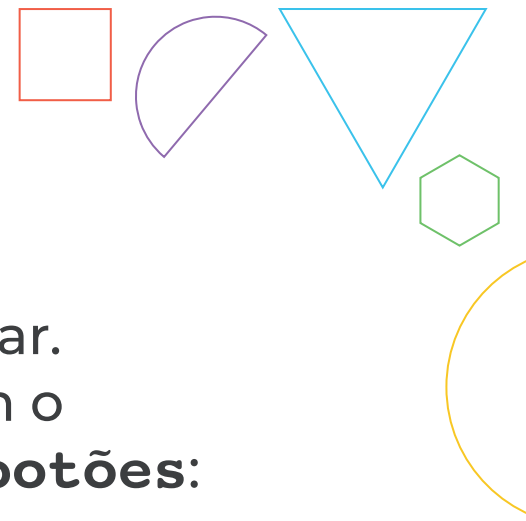
Faremos um **console.log(botoes)** para verificarmos o resultado dessa constante:

```
const botoes = document.querySelector('.parametro-senha__botao');  
  
console.log(botoes);
```

Assim, podemos ir ao navegador e checar qual botão acabamos de selecionar. Acessando a aplicação pelo navegador, pressionaremos *F12* no teclado para abrir a aba de inspeção na guia *Console*, localizada no canto superior direito da tela:



Perceberemos que o elemento selecionado é apenas o botão diminuir. Se passarmos o mouse por cima desse código no console, notaremos que somente o botão da esquerda fica em verde.

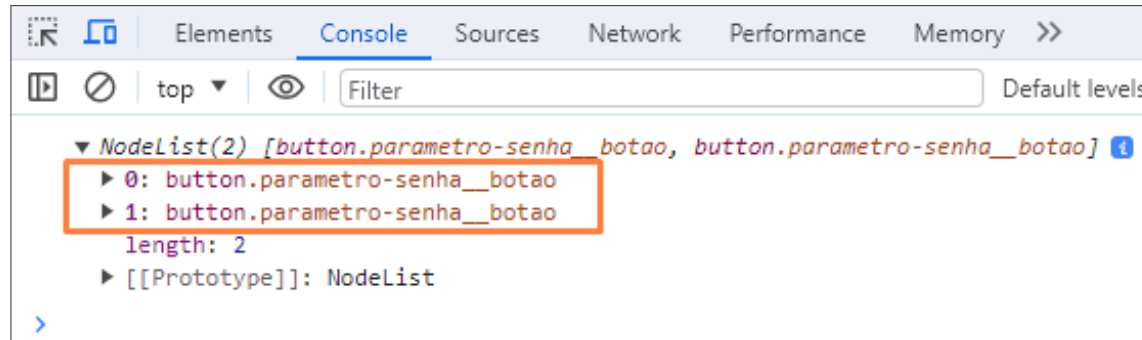


Contudo, também queremos selecionar o botão de aumentar. Portanto, não usaremos apenas o **querySelector**, mas sim o **querySelectorAll**. Dessa forma, modificaremos **const botões**:

```
const botoes = document.querySelectorAll('.parametro-senha__botao');  
console.log(botoes);
```

Agora, estamos selecionando múltiplos seletores, ou seja, obteremos os elementos no plural, incluindo os botões de diminuir e de aumentar o número de caracteres.

Fazendo o `console.log(botoes)` novamente, teremos a seguinte visualização:



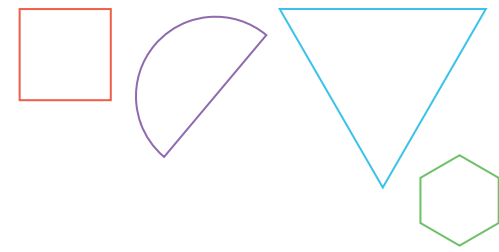
Agora, temos dois botões exibidos no console: um botão no índice 0, que é o botão diminuir, e um botão no índice 1, ou seja, o segundo botão, que aumenta o valor.

Com isso, além de utilizar esses valores, conseguiremos aplicar eventos a eles, porque a ideia não é só selecionar o valor, mas sim poder realizar o evento de clicar nos botões para que eles sejam alterados.




Então, em *main.js*, selecionaremos **botoes** na posição **[0]** — que sabemos que é o primeiro botão — e adicionaremos a ele um evento de escuta, o **onclick**, e o sinal de igual. Esse evento espera uma função de retorno, portanto, criaremos uma função para ele. Vamos chamar essa função de **diminuiTamanho**. Seu código deverá ficar da seguinte forma:

```
const botoes = document.querySelectorAll('.parametro-senha__botao');  
botoes[0].onclick = diminuirTamanho;  
  
console.log(botoes);
```



Agora que chamamos essa função, temos que garantir que, ao criá-la, o nome seja o mesmo. Portanto, criaremos a função **diminuiTamanho()**:. Observe:

```
const botoes = document.querySelectorAll('.parametro-senha__botao');  
botoes[0].onclick = diminuirTamanho;  
  
console.log(botoes);  
  
function diminuirTamanho(){  
  
}
```



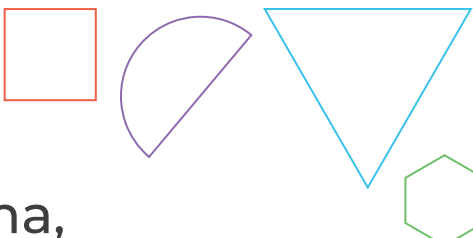
A ideia é diminuir o número de caracteres da senha quando clicarmos no botão menos (-). Ou seja, o número 5 deverá virar 4. Se clicarmos novamente, esse mesmo número deverá virar 3. Assim, necessariamente, ele deverá reduzir de um em um. Porém, para conseguir alterar esse valor 5, precisamos armazená-lo em algum lugar, ou seja, em alguma variável. E essa variável mudará o tempo todo.

Portanto, criaremos a variável **let tamanhoSenha**, que fará isso para nós. Ela receberá o valor 12, por exemplo, que é o nosso valor padrão.

```
const numeroSenha = document.querySelector('.parametro-senha__  
texto');  
let tamanhoSenha = 12;
```

Em vez de adicionar o **numeroSenha.textContent** igual a 5, vamos modificá-lo para adicionar **numeroSenha.textContent**, igual a **tamanhoSenha**. Observe:

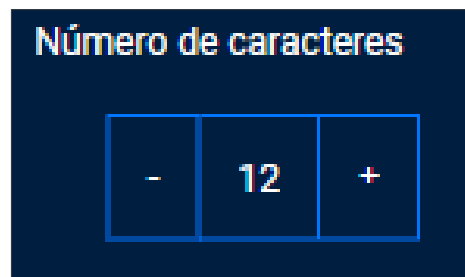
```
senha.textContent = tamanhoSenha;
```



Quando clicamos no botão para diminuir o tamanho da senha, **tamanhoSenha** precisa receber, ela mesma, menos 1. Ou seja, vamos adicionar **tamanhoSenha = tamanhoSenha - 1**; assim, quando clicarmos nesse botão, em vez de 12, teremos 12 - 1, que retornará o resultado 11 e, assim, sucessivamente. Observe:

```
function diminuiTamanho(){  
    tamanhoSenha = tamanhoSenha - 1;  
}
```

Porém, ao testarmos nosso código navegador, perceberemos que o valor ainda não está sendo alterado:



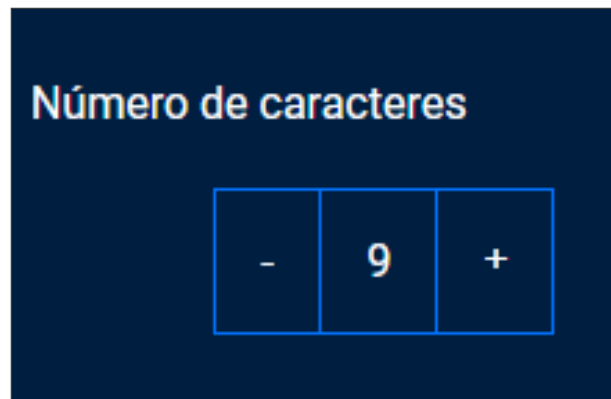
Isso ocorre porque ainda não solicitamos que o valor seja atualizado.



Dessa forma, precisamos copiar a linha `numeroSenha.textContent = tamanhoSenha` e adicioná-la dentro da função `diminuiTamanho()` para atualizar o valor.

```
function diminuiTamanho(){  
    tamanhoSenha = tamanhoSenha - 1;  
    numeroSenha.textContent = tamanhoSenha  
}
```

Após salvar o código e atualizar novamente o navegador, clicaremos no botão menos e verificaremos que, agora sim, o número está diminuindo:



Agora, faremos o mesmo processo para que o botão de mais possa aumentar o número de caracteres da senha.

# Implementando o botão de aumento

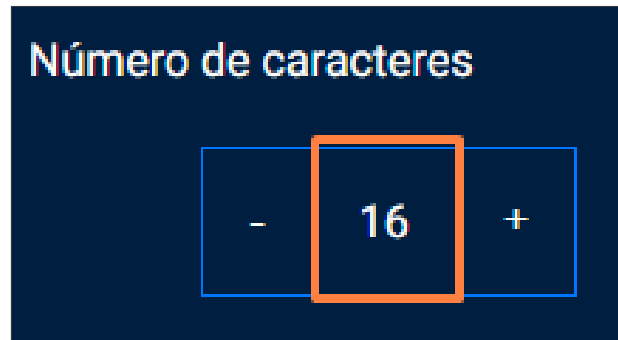
Acabamos de criar o botão no índice 0 com o evento **onClick()** para chamar a função de diminuir, a **diminuiTamanho()**. Agora, podemos incluir um botão no índice 1 utilizando o mesmo evento, porém, ele receberá a função que aumenta o tamanho da senha: a **aumentaTamanho()**. Assim, precisaremos adicionar:

```
botoes[0].onclick = diminuiTamanho;  
botoes[1].onclick = aumentaTamanho;
```

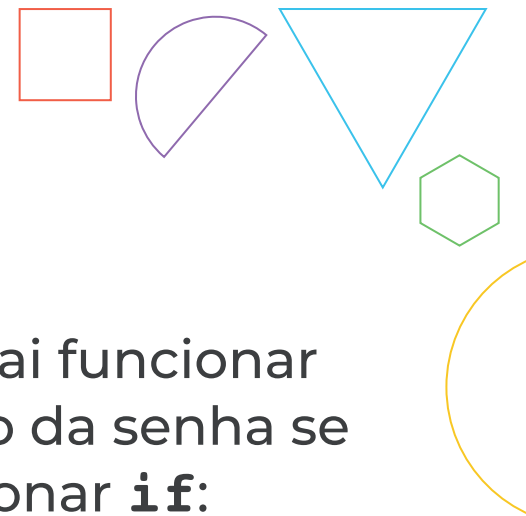
Desse modo, precisaremos também criar a função **aumentaTamanho()**, mencionada no botão de índice 1. Para isso, logo abaixo de **function diminuiTamanho**, escreveremos:

```
function aumentaTamanho() {  
    tamanhoSenha = tamanhoSenha + 1;  
    numeroSenha.textContent = tamanhoSenha;  
}
```

Vamos testar nosso código. Podemos acessar a página pela aba do navegador e clicar no botão de aumento do número de caracteres, com o símbolo de “+”. Observe que o número aumentará de um em um:



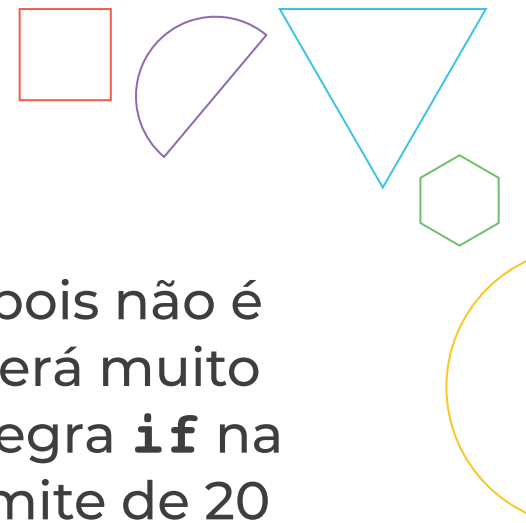
Se clicarmos no botão de diminuir (-), notaremos que ele também diminui de um em um. Contudo, se testarmos vários valores, perceberemos que é possível gerar números de senha negativos. Contudo, não faz sentido o gerador de senha fornecer uma senha negativa. Portanto, precisamos criar uma condição para que, caso o número seja menor que 1, o botão pare de diminuir pare de funcionar.



Para isso, precisamos trabalhar com uma condição que só vai funcionar se for verdadeira. Portanto, só podemos diminuir o tamanho da senha se soubermos que o valor é maior que 1. Portanto, vamos adicionar **if**:

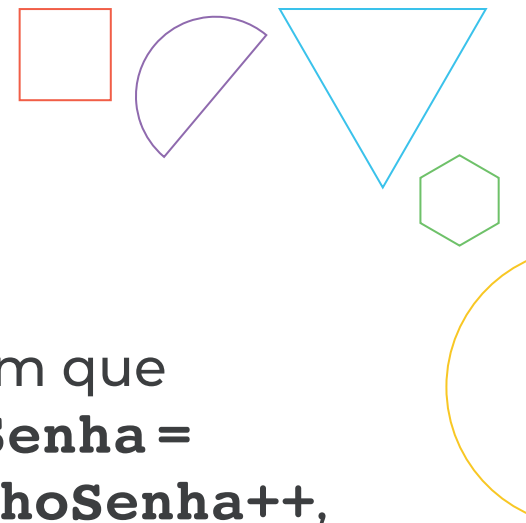
```
function diminuiTamanho() {  
  if (tamanhoSenha > 1)  
    tamanhoSenha = tamanhoSenha - 1;  
    numeroSenha.textContent = tamanhoSenha;  
}
```

Se testarmos o código, veremos que, ao clicarmos no botão de menos, o valor chegará a 0 e parará de funcionar.



Precisamos fazer a mesma coisa para o botão de aumento, pois não é conveniente gerar senhas com 100 ou 500 caracteres, pois será muito difícil lembrá-las. Assim, também podemos criar a mesma regra **if** na primeira linha de **aumentaTamanho()**. Vamos aplicar um limite de 20 caracteres, pois julgamos ser uma boa quantidade de caracteres para uma senha segura.

```
function aumentaTamanho() {  
  if (tamanhoSenha < 20) {  
    tamanhoSenha = tamanhoSenha + 1;  
  }  
  numeroSenha.textContent = tamanhoSenha;  
}
```



Para finalizarmos, podemos otimizar nosso código, o que é considerado uma boa prática de programação. Nas linhas em que utilizamos `tamanhoSenha = tamanhoSenha - 1` e `tamanhoSenha = tamanhoSenha - 1`, utilizaremos `tamanhoSenha--` e `tamanhoSenha++`, respectivamente. Neste caso, estamos simplificando a sintaxe do código sem modificar a mensagem emitida, que continuará funcionando normalmente.

Porém, nossa aplicação ainda não está gerando a senha. Então, na próxima aula, vamos gerar senhas usando o número de caracteres que acabamos de adicionar, apenas com letras maiúsculas, que é o único checkbox que foi marcado até o momento.

Até breve!

## ► Desafio

Nesta aula, aprendemos a adicionar um evento ao botão de menos, o qual, ao ser pressionado, diminui um do valor total inicial de caracteres da senha. Seu desafio será pesquisar sobre quais são os eventos que podemos utilizar num projeto usando o JavaScript. Procure no mínimo 5 para que você tenha noção do que é possível ser feito utilizando essa linguagem de programação e tente aplicar ao seu projeto aquele que julgar mais pertinente.



CLIQUE AQUI PARA AVALIAR ESTE MATERIAL