How to use the “.split” function

How to use the querySelector

# 03 JavaScript: Password Generator

Create an application that generates a random password based on user-selected criteria. This app will run in the browser and feature dynamically updated HTML and CSS powered by your JavaScript code. It will also feature a clean and polished user interface and be responsive, ensuring that it adapts to multiple screen sizes.

If you are unfamiliar with special characters, take a look at [some examples](https://www.owasp.org/index.php/Password\_special\_characters).

## User Story

```

AS AN employee with access to sensitive data

I WANT to randomly generate a password that meets certain criteria

SO THAT I can create a strong password that provides greater security

```

## Acceptance Criteria

GIVEN I need a new, secure password

1. ~~WHEN I~~ **~~click~~** ~~the button to generate a password THEN I am presented with a series of prompts for password criteria~~
2. ~~WHEN prompted for password criteria~~ **~~THEN I select which criteria to include in the password~~**
3. ~~WHEN prompted for the~~ **~~length~~** ~~of the password THEN I choose a length of at least~~ **~~8 characters and no more than 128 characters~~**
4. ~~WHEN prompted for character types to include in the password THEN I choose~~ **~~lowercase, uppercase, numeric, and/or special characters~~**
5. ~~WHEN I answer each prompt THEN~~ **~~my input should be validated~~** ~~and~~ **~~at least one character type should be selected~~**
6. ~~WHEN all prompts are answered~~ **~~THEN a password is generated~~** ~~that matches the selected criteria~~
7. ~~WHEN the password is generated~~ **~~THEN the password is either displayed in an alert~~****~~or written~~** ~~to the page~~

```

The following image demonstrates the application functionality:

![password generator demo](./Assets/03-javascript-homework-demo.png)

## Review

You are required to submit the following for review:

\* The URL of the deployed application.

\* The URL of the GitHub repository. Give the repository a unique name and include a README describing the project.

\*

\* Generates a random numer between 0 and i.

\*/

// function getRandom(i) {

// return (Math.floor(Math.random() \* i));

// }

// /\*

// \* Picks a random character out of the given string.

// \*/

// function getRandomChar(characters) {

// return characters.charAt(getRandom(characters.length));

// }

// function getPassword(length, uppercase, lowercase, numbers, extras) {

// var hallo = document.getElementsByName("password")[0];

// var uppercaseLetters = "QWERTZUIOPASDFGHJKLYXCVBNM";

// var lowercaseLetters = "qwertzuiopasdfghjklyxcvbnm";

// var numberChars = "0123456789";

// var extraChars = "<>\,.;:-\_+\*%&/()=?!{}[]";

// // get the characters

// var pw = "";

// var i=0;

// while (i<length) {

// if (uppercase==true) {

// pw += getRandomChar(uppercaseLetters);

// i++;

// }

// if (lowercase==true && i<length) {

// pw += getRandomChar(lowercaseLetters);

// i++;

// }

// if (numbers==true && i<length) {

// pw += getRandomChar(numberChars);

// i++;

// }

// if (extras==true && i<length) {

// pw += getRandomChar(extraChars);

// i++;

// }

// }

// // scramble the pw

// for (var j=0; j<20; j++) {

// var pos1 = getRandom(pw.length);

// var pos2 = getRandom(pw.length);

// var newPw="";

// for (var k=0; k<pw.length; k++) {

// if (k==pos1)

// newPw += pw.charAt(pos2);

// else if (k==pos2)

// newPw += pw.charAt(pos1);

// else

// newPw += pw.charAt(k);

// }

// pw = newPw;

// }

// return pw;

// }