Assignment 4 (HF, major subject)

Due: Wed 23.11.2016; 20:00h (1 Week)

Goals

After doing these exercises,

- You can handle events with jQuery
- You know how to use JavaScript Objects
- You know how to traverse and manipulate the DOM with jQuery
- You know how to make sufficiently strong passwords

Task 1: Random Color Generator   Difficulty: Easy

Sometimes, you could use a little inspiration regarding the choice of colors. So you come up with the idea to create a small webpage that allows you to quickly generate a random color. The page is entirely colored in the random color, and if you like it, it shows you the hexadecimal color code. Here's how this could look:

Generating a random color is pretty easy and possible in many ways. Here's a solution from stackoverflow:

```
function getRandomColor() {
    return "#" + (Math.random() * 0xFFFFFF << 0).toString(16);
}
```

To help you get started, we provide a small skeleton on GitHub:
https://github.com/MIMUC-MMN/assignments-16-17/tree/master/04%20-%20jquery%20basics/skeleton/

Four simple steps: a) add an event listener to the button, b) generate a random color whenever the button is clicked, c) set the background color of the #bottom div with .css(), and d) add the new color name to the #colorDisplay div via .html().

Put your code into the file random-color-generator.html and include it in your submission.

Task 2: Password Meter   Difficulty: Medium-Hard

On many websites, users can create accounts with a username and password. For critical, high-value accounts (e.g. email, banking) it is advisable to choose a password that would withstand at least one million guesses in a so called *online attack*. In *offline attacks*, the attackers can try many more times, so passwords should withstand one trillion guesses (10^{12}) or more to provide reasonably strong protection.

Users often struggle to create such passwords. Therefore, websites often provide some kind of feedback when users enter a password during registration. You can often see this as "strength bar" or "password meter", for example when you create an Apple ID online.

In this task, you will create such a feedback system using the popular password estimation library zxcvbn (https://github.com/dropbox/zxcvbn). zxcvbn provides everything you need:

a) a score between 0 (weak) and 4 (strong)
b) estimations about how long it would take to crack the given password.
c) feedback about the password characteristics (e.g. "This is a top-10 common password")
d) suggestions about how to improve the password.

You can see zxcvbn's output on its demo page:
https://dl.dropboxusercontent.com/u/209/zxcvbn/test/index.html

Create a webpage that immediately analyzes a password as the user types it. There should be a 'strength bar' and some verbal feedback to inform the user about the strength. Watch a demo of our solution on YouTube: https://youtu.be/fBwasnSQ8UA

Remember, you write code humans – not for machines. Make sure to comment your code as much as reasonable.

Put your solution (all program code) into a folder 'task2'.

Submission

Please turn in your solution via UniWorX. You can form groups of up to three people.

We encourage you to sign up for Slack! All you need is a CIP account and an email address that ends in "@cip.ifi.lmu.de". Ask us if you don't know how to get them.

If you have questions or comments before the submission, please contact one of the tutors. They are on Slack @tobi.seitz, @peterjuras and @thomas-weber. Remember, that they also want to enjoy their weekends ☺

It also makes sense to ask the question in our #mmn-ws1617 channel. Maybe fellow students can help or benefit from the answers, too!

CodeLabs

As always, everyone is invited to participate in the CodeLabs on Wednesday between 6 and 8 pm. We encourage you to go there if you struggle with installation or nasty bugs that just don't seem to go away. The times are flexible, so you can go there sometime during the CodeLab.