



Assignment 2 (HF, major subject)

Due: Wed 09.11.2016; 20:00h (1 Week)

Goals

After doing these exercises,

- You will know how to use cookies and inform the users about them
- You can utilize sessions to persist data and make web apps stateful

Task 1: Cookies & Session Quiz

Difficulty: Easy

Please answer the following questions in **one** sentence each. Some of them require a small internet search.

- a) Why do we need cookies?
- b) Why is it necessary to call `session_start()` before any output is generated by PHP?
- c) Cookies are “serialized”. Explain this term.
- d) What’s the name of a cookie that is deleted after the browser is closed?

Put your answers into a file ‘task1.pdf’.

Task 2: Codebreaker

Difficulty: Difficult

This is a quite work-intensive task, so you might want to start early during the week. We can also help you solve the task during the code labs.

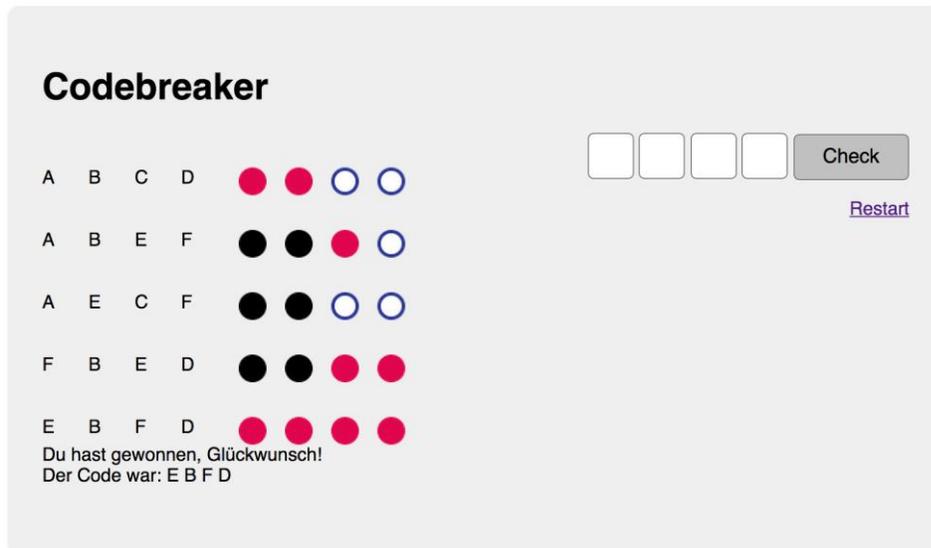
Create a game where the player has to guess a **four**-letter code. The code is formed of a random combination of the letters *A, B, C, D, E, F,* and *G*. Every letter can only appear once inside the four-letter code. The player enters his or her guess into a form and receives a hint until they found out the code. The hint has the following semantics:

- A **red dot** indicates that one of the guessed letters is part of the code and also at the correct position.
- A **black dot** indicates that one of the guessed letters is part of the code but not at the correct position
- A **white dot** indicates that a letter is not part of the code.

Task description continued on next page



Example: The code is **EBFD** here. If the player enters these guesses (on the left), the web page should display the corresponding hints (colored dots):



Consider these requirements:

- Use PHP and sessions to keep track of all the attempts.
- Create a random four-letter code from the alphabet { A, B, C, D, E, F, G } when the user starts the game (i.e. visits the site for the first time).
- The user can guess at most **ten** times. If the code is incorrect then the game is lost and the code is revealed to the player.
- The hints are presented as four colored dots. You can use image files for this or CSS / SVG.
- All previous guesses are displayed.
- The player can restart the game.

Optional:

- Sanity-check the player's input. For example: do not count a guess if it contains a letter twice or a letter from outside the limited alphabet.
- Show a leaderboard after the game is finished. The player can enter their name and save it. Use cookies for this.

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!

Let’s collaborate on GitHub!

As we do not provide sample solutions, we encourage you to collaborate with all your peers on a sample solution for this assignment on GitHub.

We created a public repository for this purpose:
<https://github.com/MIMUC-MMN/assignments-16-17>

It includes a short “how to contribute” guide, to get you started.

The staff will always check what’s in there and add comments or push updates.

The screenshot shows the GitHub repository page for `MIMUC-MMN / assignments-16-17`. The repository is public and has 27 commits, 1 branch, 0 releases, and 3 contributors. The file list includes:

File/Folder	Description	Last Commit
01 - php basics	added utf-8 support for scrambler and prettified it a lil	4 days ago
02 - php sessions	folder structure and read me	11 days ago
03 - php database access	folder structure and read me	11 days ago
04 - jquery basics	folder structure and read me	11 days ago
05 - jquery ajax	folder structure and read me	11 days ago
06 - vanilla js	folder structure and read me	11 days ago
07 - nodejs basics	folder structure and read me	11 days ago
08 - nodejs authentication	folder structure and read me	11 days ago
09 - nodejs misc	folder structure and read me	11 days ago
10 - web dev tools	folder structure and read me	11 days ago
11 - metadata epub	folder structure and read me	11 days ago
12 - polymer basics	folder structure and read me	11 days ago
13 - polymer components	folder structure and read me	11 days ago
14 - repetition	folder structure and read me	11 days ago
.gitignore	folder structure and read me	11 days ago
README.md	inserted a missing step into the setup tutorial	7 days ago