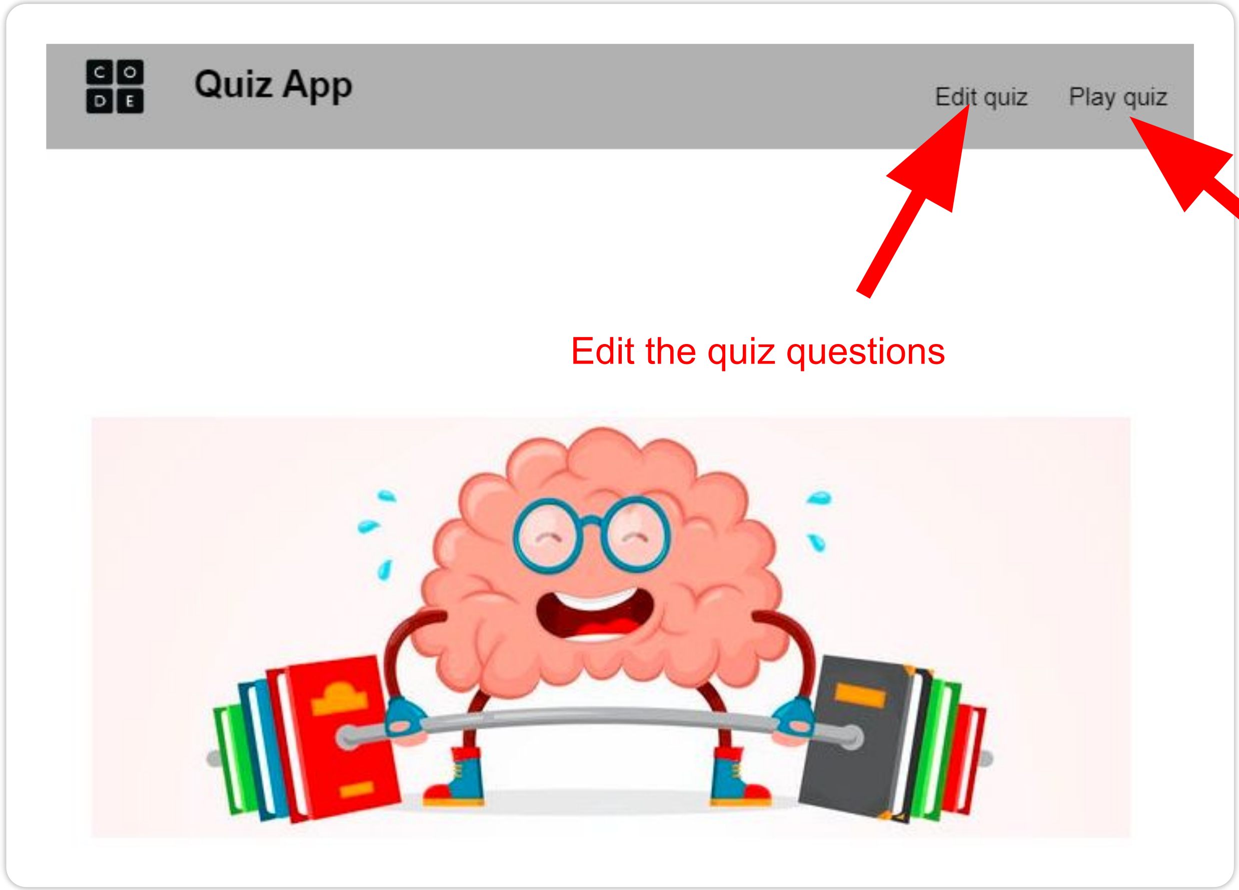
CHAPTER 2

CREATE A QUIZ

10 MIN

DEMO TEACHER



Edit the quiz questions

Play the quiz

Let’s work on last year’s project !

PRACTICE **LEARNING OBJECTIVES**

- ✔ Separate your website into 2 **sub views**
- ✔ Use the **local storage** to save/load data
- ✔ Use the **dataset** on HTML elements
- ✔ Create a **dialog** to edit or add a new question

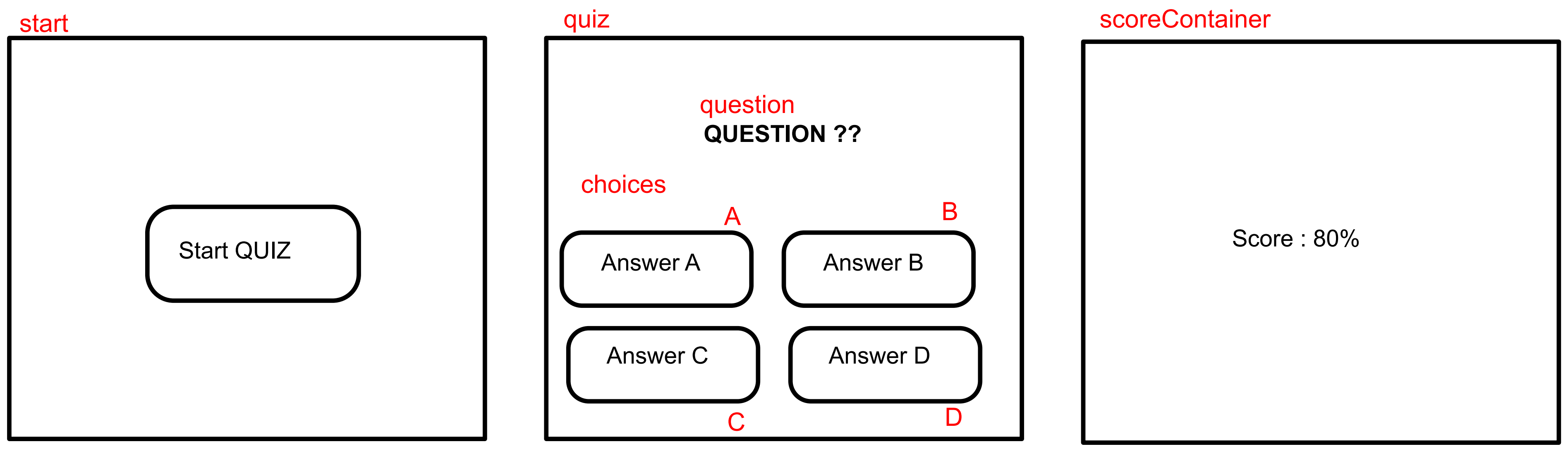
UNDERSTAND CODE FIRST!

In this practice you will be asked to **extend an existing project**. You **first need to understand** the existing variables and functions, before starting your code!

5 MIN

STEP 1

✔ The application is composed of 3 views: start, quiz, score

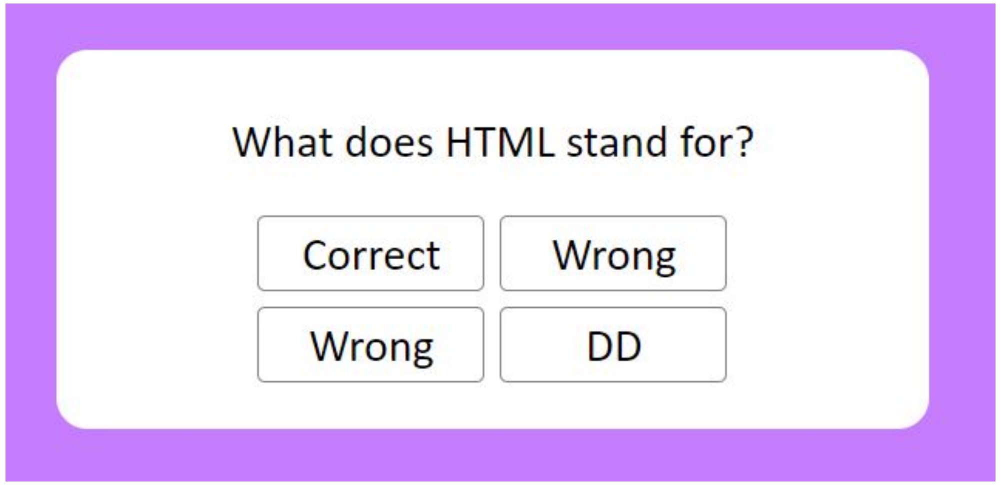


TODO

✔ Complete the **hide / show functions** to display only the start view at the beginning

30 MIN **STEP 2** Display quiz and compute score

```
{
    title: "What does HTML stand for?",
    choiceA: "Correct",
    choiceB: "Wrong",
    choiceC: "Wrong",
    correct: "A",
  },
```



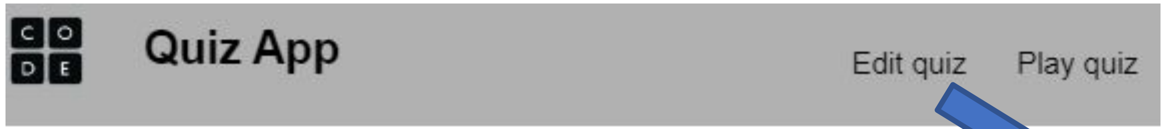
TODO:

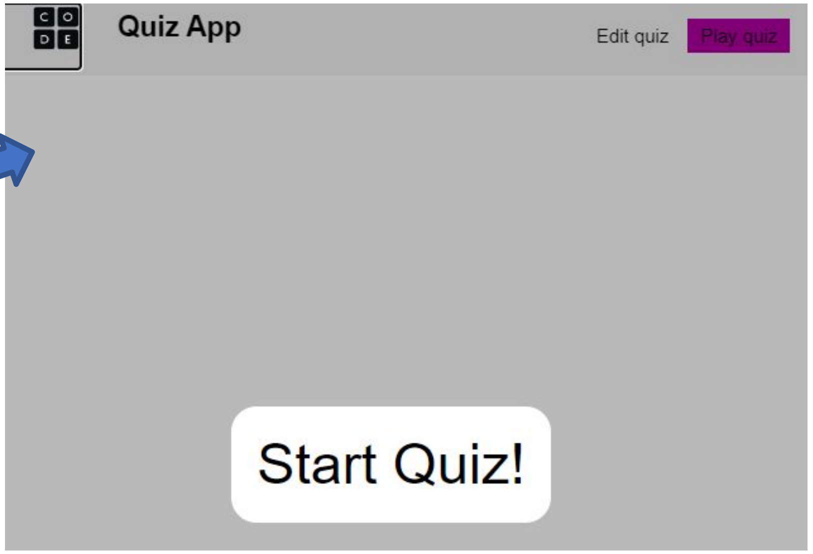
- ✔ Try to understand the meaning of the 3 global variables: *questions, runningQuestionIndex, score*
- ✔ Complete **renderQuestion()**
- ✔ When is the function **checkAnswer(answer)** called, and what does it do ?
- ✔ In **renderScore**(): compute the final score in **percentage**

Refactor the project into 2 sub views

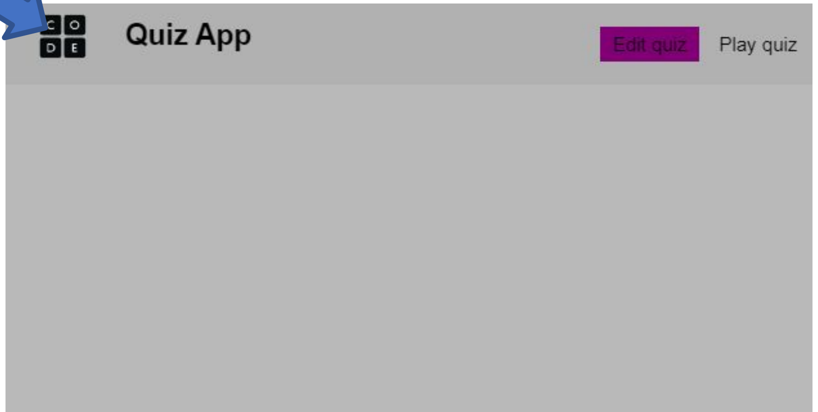
/views/play/play.html : the view to PLAY the QUIZ

/index.html : The start page :





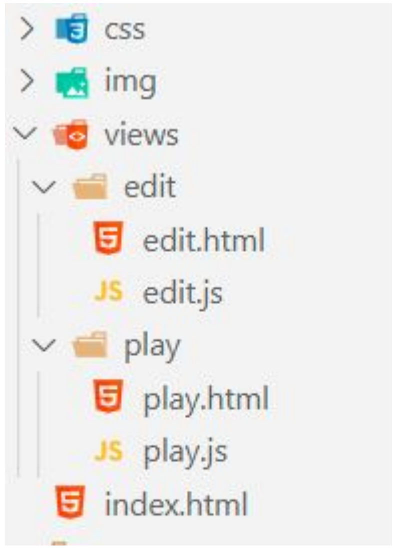
/views/edit/edit.html : the view to EDIT the QUIZ



20 MIN **STEP 3** Refactor the project into views

TODO

✔ Understand the project structure



HTML and JS to manage the QUIZ edition

HTML and JS to manage the QUIZ play

✔ Update the index.html to link with **the 2 sub views**

✔ *Don't forget to include the menu in the 2 sub views, and to display the selected menu item*

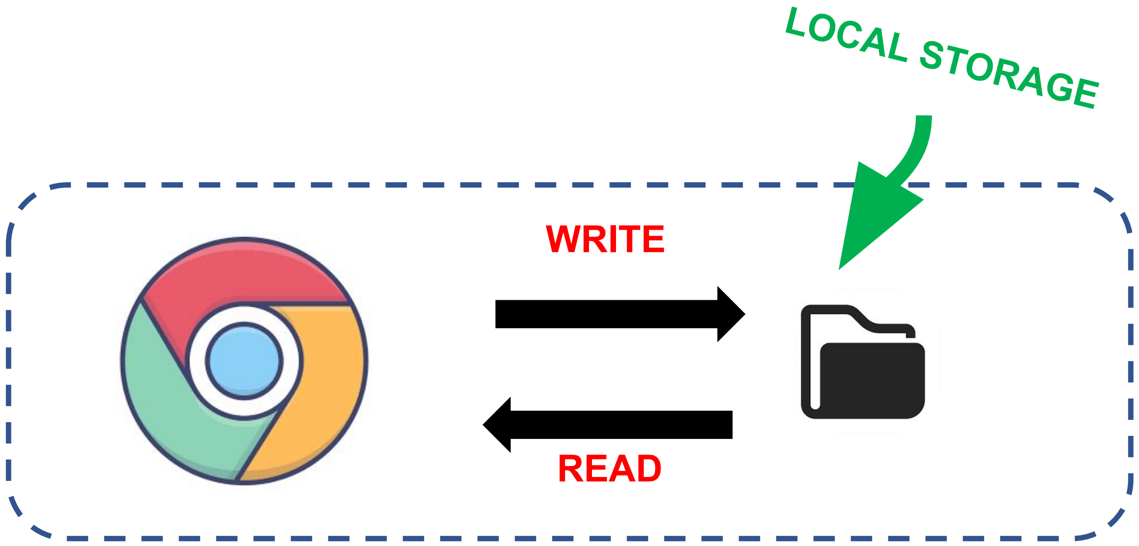
Use the CSS class `active` to style the active menu item

✔ Transfer your previous code from STEP 2 to the sub view **PLAY**

STEP 4

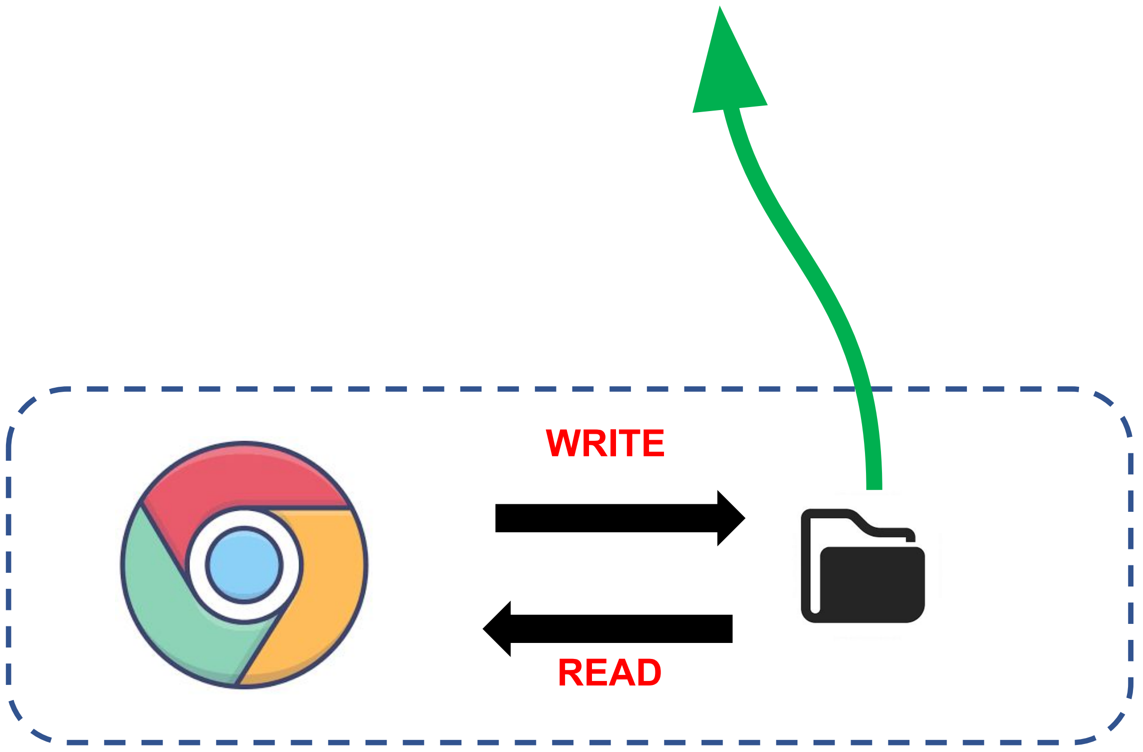
Now, let’s use the **Local storage** to **store the QUIZ** on the browser data

Browsers allow you to save
Up to 10 Mb of data
locally



The local storage is a dictionary

local Storage = { key: value, key: value,… }



Set a value

```
localStorage.setItem("amount", 12);
```

GLOBAL
VARIABLE

KE
Y

VALU
E

Get a value

```
let amount = localStorage.getItem("amount");
```

KE

Clear all

```
localStorage.clear();
```

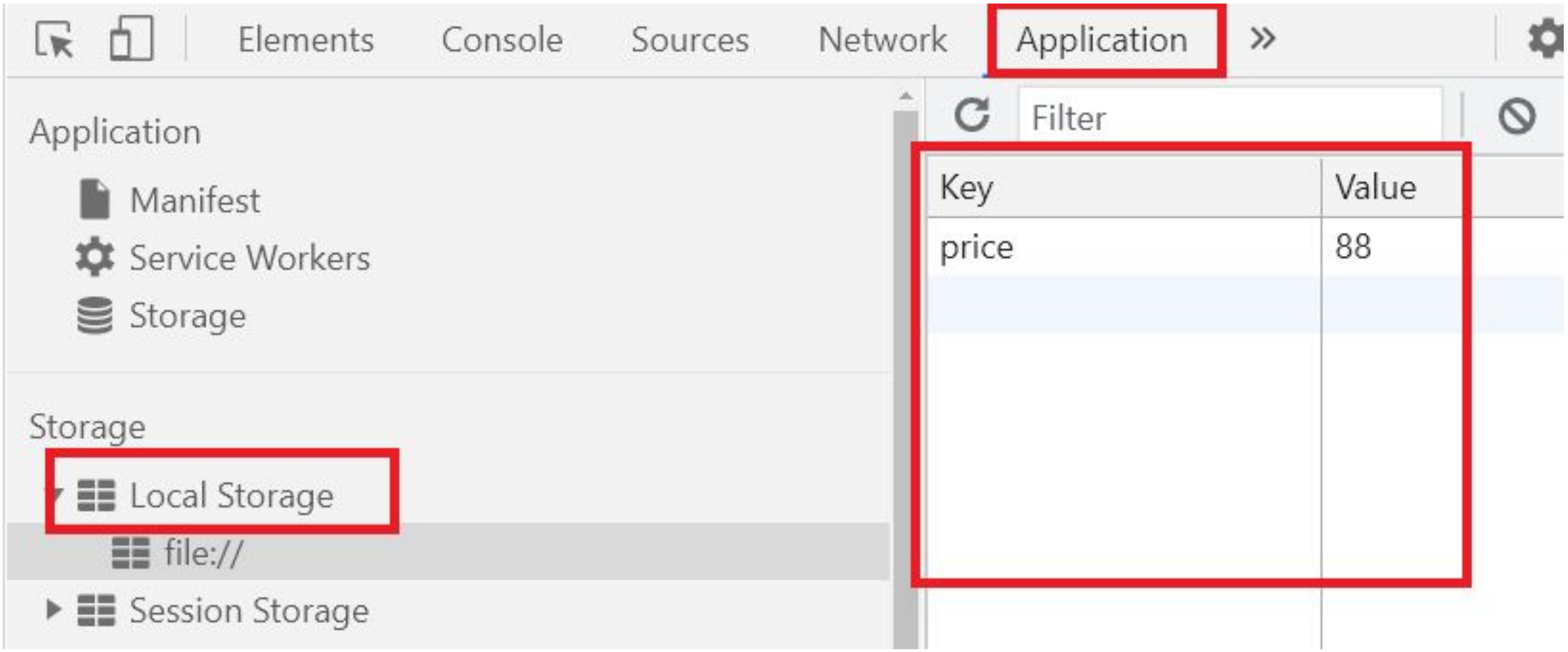
Clear only 1 item

```
localStorage.removeItem("amount");
```

KEY

DEMO

How to see your local storage in Chrome



Key	Value
price	88

30 MIN **STEP 4**

Save/load the QUIZ on local storage

TODO

✔ Code the function **saveQuestions**() to **save the list of questions** to the local storage

*You can use "**questions**“ as a key to store question in the local storage*

✔ Code the function **loadQuestions**() to **load the list** of questions from the local storage

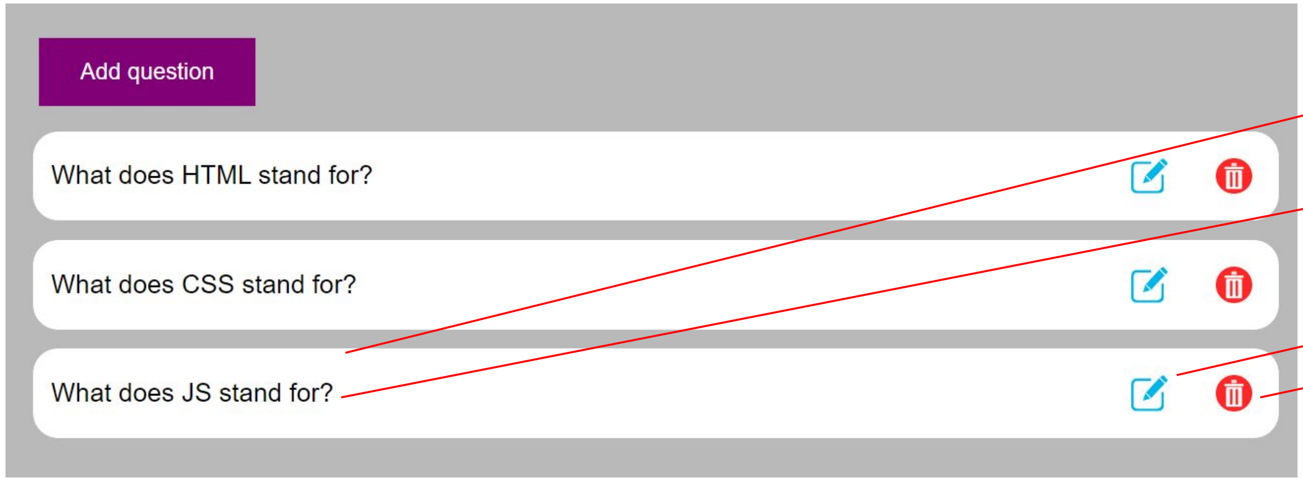
*Warning: if there is no data on local storage, you should not update the global variable **questions***

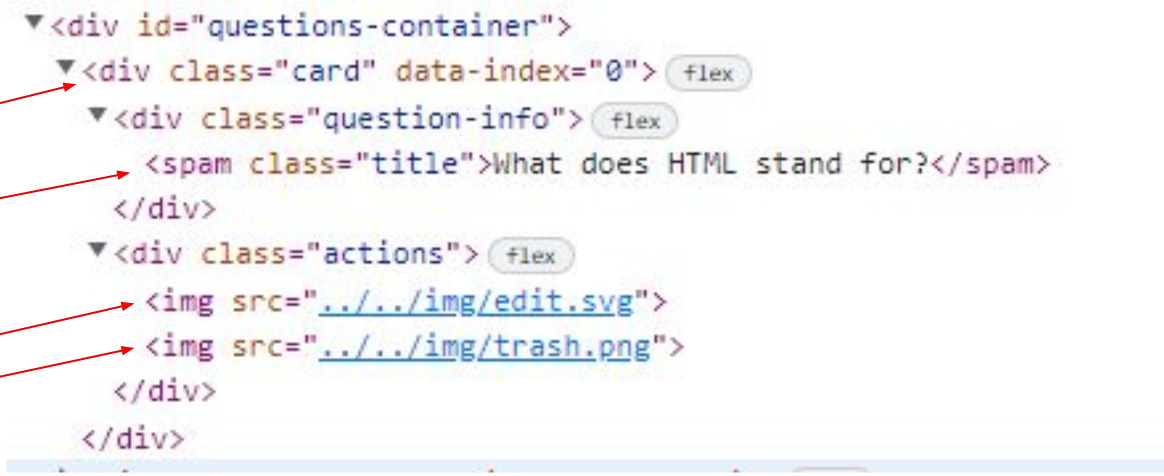
✔ Test your local storage save/load functions :

1. Launch saveQuestions() when the view PLAY is displayed
2. Check the questions are saved on the local storage
3. Replace saveQuestions() by loadQuestion() when starting to play quiz
4. Change 1 question on the local storage
5. Reload the page : check we load question from the local storage

45 MIN **STEP 5**

Edit quiz - List all questions





```
▼<div id="questions-container">
  ▼<div class="card" data-index="0"> flex
    ▼<div class="question-info"> flex
       <spam class="title">What does HTML stand for?</spam>
     </div>
    ▼<div class="actions"> flex
       <img src="../../img/edit.svg">
       <img src="../../img/trash.png">
     </div>
   </div>
```

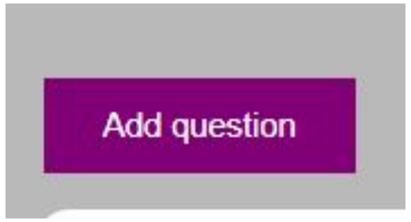
TODO

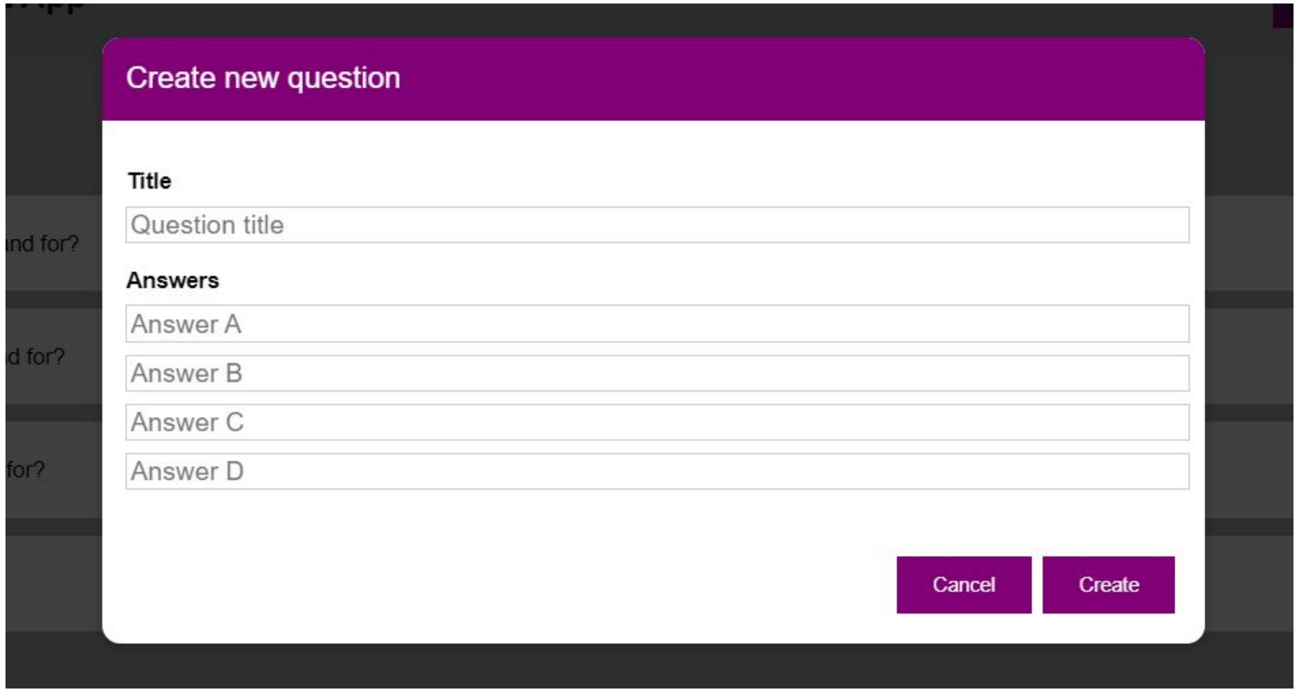
✔ Code the function **renderQuestions**() to **display all existing questions** with the right HTML/CSS attributes

✔ Complete the function **removeQuestion**() to remove the question when clicking on remove button

Warning: you need to understand how to use dataset to keep the index of the question on the question card

45 MIN **STEP 6** Create a dialog and add questions




Create new question

Title

Question title

Answers

Answer A

Answer B

Answer C

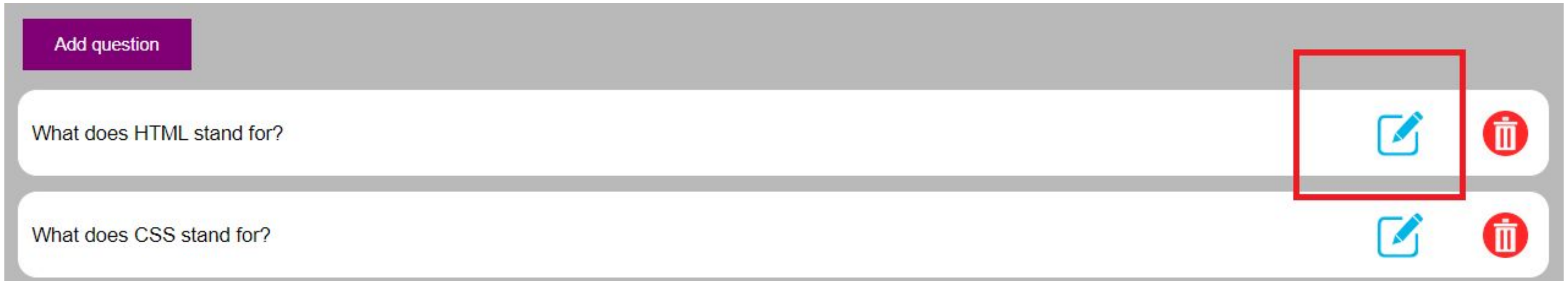
Answer D

TODO

- ✔ On EDIT html: call the right functions when clicking on buttons
- ✔ On EDIT JS: Code the functions **onCreate**() , **onAddQuestion**() , **onCancel**()

45 MIN **STEP 7**

Edit a question



What does HTML stand for?

What does CSS stand for?

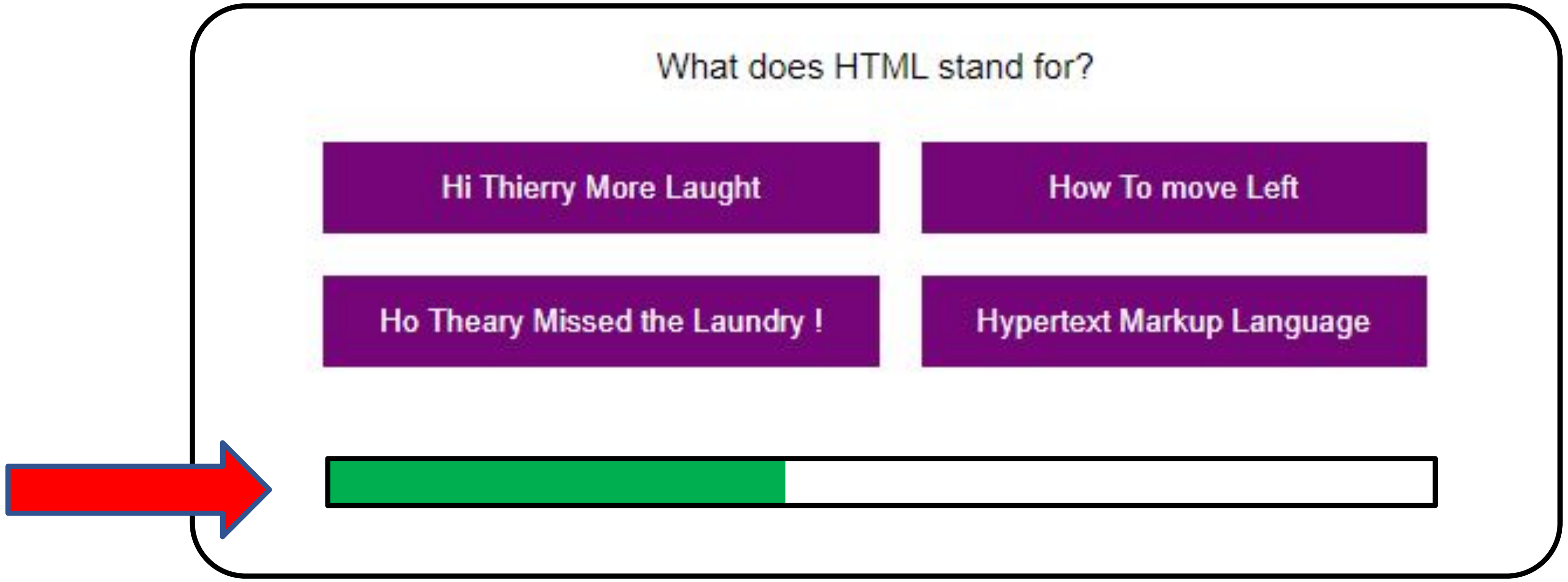
TODO

- ✔ On EDIT JS : Code the functions **editQuestion** (event)
  - ✔ *Keep the ID of the edited question using a global variable*
- ✔ On EDIT JS : Update the functions **onCreate** ()
  - ✔ *This function should manage to edit the edited question or create a new question, depending on the situation*

20 MIN **STEP 8** Add a progress bar

TODO

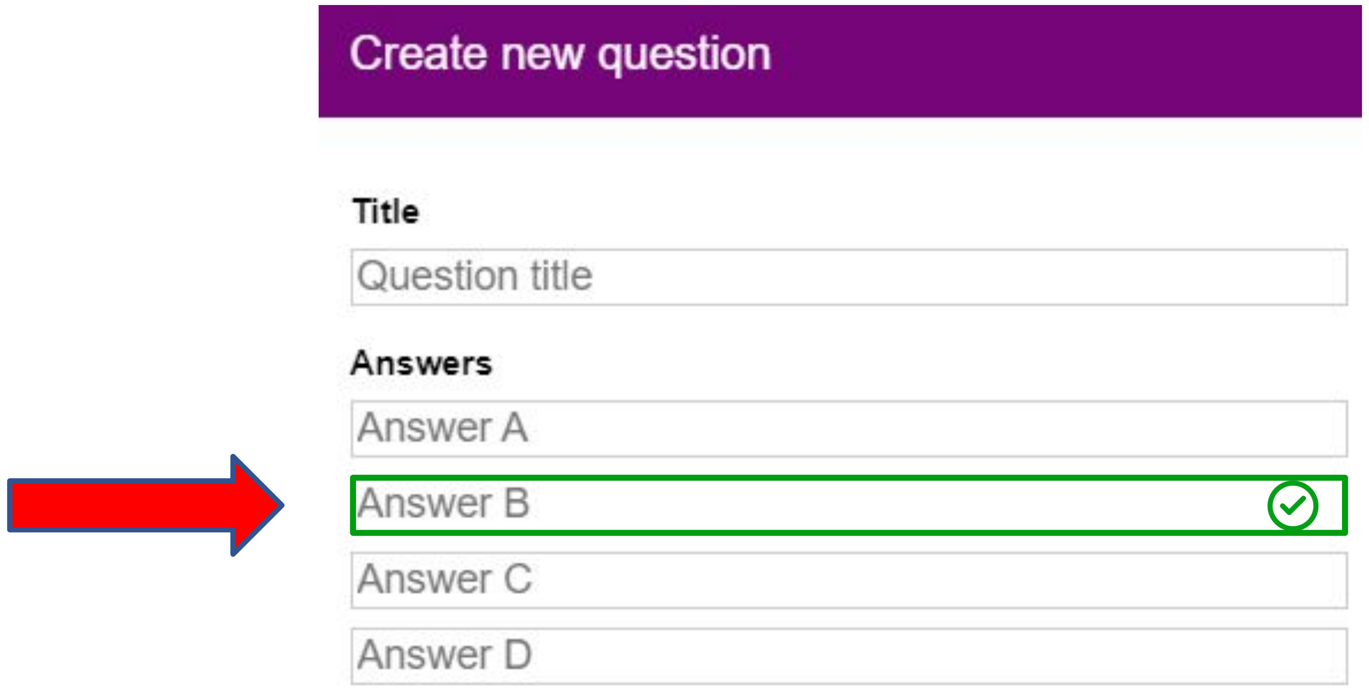
✔ Add a progress bar on the PLAY view to see the **progress** in %


What does HTML stand for?
Hi Thierry More Laught
How To move Left
Ho Theary Missed the Laundry !
Hypertext Markup Language

BONUS

BONUS 1

✔ On the DIALOG, add a way to select the **GOOD answer**


Create new question

Title

Question title

Answers

Answer A

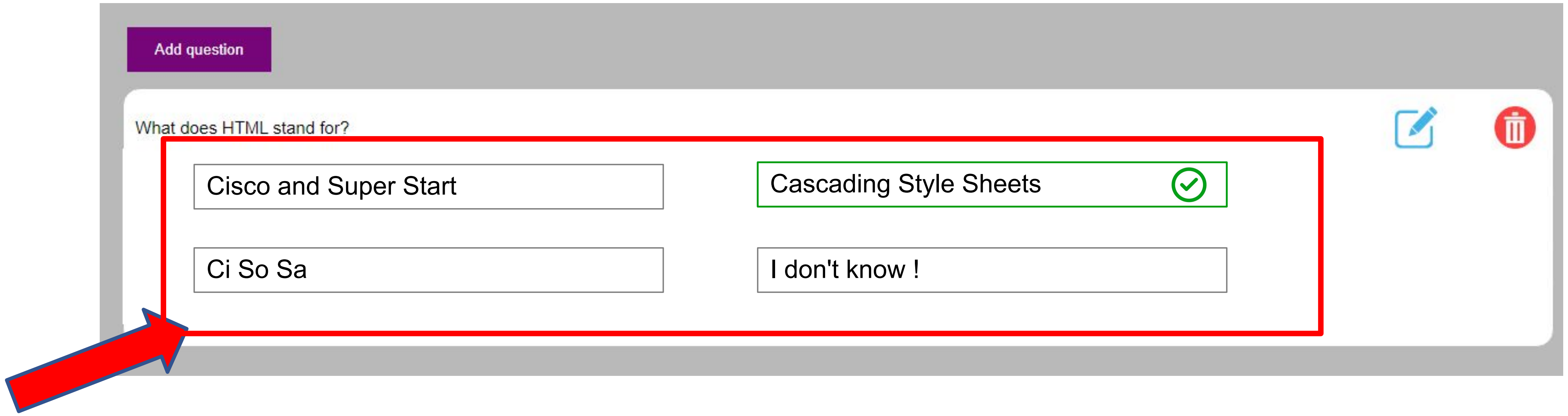
Answer B ✓

Answer C

Answer D

BONUS 2

✔ On the EDIT VIEW, display the **4 possible answers** and the **correct answer**



Add question

What does HTML stand for?

Cisco and Super Start

Cascading Style Sheets ✓

Ci So Sa

I don't know !