

Exercise walk-through

2.4

Adding Interactivity with JavaScript

GAC Computing

You must be able to

- explain the DOM and an event
- select an element after the page is loaded
- keep basic page functions usable without JavaScript

1

The method

Method — Respond to a click

`document.querySelector('#menu')` finds the menu in the **DOM** 文档对象模型. A click is an **event** 事件. An event listener runs a function when that click happens.

Make the menu a working list in HTML first. JavaScript may then hide or show it. If the script fails, visitors still have a usable menu.

2

Practice

Practice 2.1 ■ 1 mark

Define the DOM.

Solution 2.1

Worked steps

- The loaded web page represented as a tree of objects that code can read and change. **B1**

Practice 2.2 ■ 1 mark

State one example of a browser event.

Solution 2.2

Worked steps

- Click, submit, key press, or scroll. **B1**

Practice 2.3 ■ 2 marks

Explain why a script that runs before the HTML is parsed may fail.

Solution 2.3

Method: Respond to a click — p.4

Worked steps

- The elements may not exist in the DOM yet. **B1**
- The selector finds nothing, so later code cannot change the element. **A1**

3

Exam-style

Exam-style 3.1 ■ 4 marks

Explain how a click listener can make a menu button show or hide a panel.

Solution 3.1

Method: Respond to a click — p.4

Worked steps

- Select the button and panel from the DOM. **B1**
- Add a listener for the button's click event. **A1**
- The listener runs a function that changes the panel's hidden state. **A1**
- The browser waits for the event; the code does not need a loop. **B1**

Exam-style 3.2 ■ 4 marks

Explain why client-side form validation is useful but cannot protect a system by itself.

Solution 3.2

Worked steps

- It gives immediate feedback to a user before submission. **B1**
- A visitor can bypass or change browser-side code. **A1**
- Therefore it cannot be trusted for security. **A1**
- The server must validate the same rules again. **B1**