

1 JavaScript (jQuery) 基礎問題

/ 60

3x20

1-1 JavaScript を外部ファイルとして読み込むための正しい HTML を選択してください。

- ☐ `<script href="script.js"></script>`
- ☐ `<script src="script.js"></script>`
- ☐ `<link src="script.js">`
- ☐ `<javascript src="script.js">`

1-2 jQuery 本体を読み込んだ後に、自作の JavaScript を読み込む正しい順番を選択してください。

- ☐ 自作 JavaScript → jQuery
- ☐ CSS → 自作 JavaScript → jQuery
- ☐ jQuery → 自作 JavaScript
- ☐ 順番は関係ない

1-3 HTML の読み込みが完了してから処理を実行する jQuery の記述として、正しいものを選択してください。

- ☐ `$(function () {});`
- ☐ `$(html () {});`
- ☐ `function.$( () {});`
- ☐ `$document(function {});`

1-4 ID が title の要素を選択する正しい記述を選択してください。

- ☐ `$(".title")`
- ☐ `$("title")`
- ☐ `$("#title")`
- ☐ `$("*title")`

1-5 クラス名が item の要素を選択する正しい記述を選択してください。

- ☐ `$("#item")`
- ☐ `$(".item")`
- ☐ `$("item")`
- ☐ `$("@item")`

1-6 すべての p 要素を選択する正しい記述を選択してください。

- ☐ \$("#p")
- ☐ \$(".p")
- ☐ \$("<p>")
- ☐ \$("p")

1-7 ID が button の要素にクリックイベントを設定する正しい記述を選択してください。

- ☐ \$("#button").on("click", function () {});
- ☐ \$("#button").click = function () {};
- ☐ \$("#button").event("click") {};
- ☐ \$("#button").onClick(function) {};

1-8 ID が name の入力欄から、入力された値を取得する正しい記述を選択してください。

- ☐ \$("#name").text()
- ☐ \$("#name").html()
- ☐ \$("#name").val()
- ☐ \$("#name").value()

1-9 ID が name の入力欄を空にする正しい記述を選択してください。

- ☐ \$("#name").val("")
- ☐ \$("#name").text("")
- ☐ \$("#name").empty("")
- ☐ \$("#name").remove("")

1-10 ID が message の要素に「完了」と表示する正しい記述を選択してください。

- ☐ \$("#message").val("完了")
- ☐ \$("#message").text("完了")
- ☐ \$("#message").show("完了")
- ☐ \$("#message").add("完了")

1-11 ID が box の要素を非表示にする正しい記述を選択してください。

- ☐ \$("#box").hide()
- ☐ \$("#box").hidden()
- ☐ \$("#box").none()
- ☐ \$("#box").close()

1-12 ID が box の要素を表示する正しい記述を選択してください。

- ☐ \$("#box").open()
- ☐ \$("#box").visible()
- ☐ \$("#box").show()
- ☐ \$("#box").display()

1-13 ID が box の要素に active クラスを追加する正しい記述を選択してください。

- ☐ \$("#box").class("active")
- ☐ \$("#box").addClass("active")
- ☐ \$("#box").addId("active")
- ☐ \$("#box").cssClass("active")

1-14 ID が box の要素から active クラスを削除する正しい記述を選択してください。

- ☐ \$("#box").deleteClass("active")
- ☐ \$("#box").remove("active")
- ☐ \$("#box").removeClass("active")
- ☐ \$("#box").emptyClass("active")

1-15 ID が box の要素に active クラスがあれば削除し、なければ追加する記述を選択してください。

- ☐ \$("#box").changeClass("active")
- ☐ \$("#box").switchClass("active")
- ☐ \$("#box").addClass("active")
- ☐ \$("#box").toggleClass("active")

1-16 ID が list の要素内の最後に、新しい li 要素を追加する正しい記述を選択してください。

- ☐ \$("#list").append(" 項目 ")
- ☐ \$("#list").afterList(" 項目 ")
- ☐ \$("#list").add(" 項目 ")
- ☐ \$("#list").htmlAdd(" 項目 ")

1-17 クリックされた要素自身を jQuery で選択する正しい記述を選択してください。

- ☐ \$("this")
- ☐ \$("#this")
- ☐ \$(this)
- ☐ \$(".this")

1-18 選択した HTML 要素を画面から削除する正しいメソッドを選択してください。

- ☐ .hide()
- ☐ .remove()
- ☐ .delete()
- ☐ .clear()

1-19 変数 score が 80 以上であることを判定する正しい条件式を選択してください。

- ☐ if (score ==> 80)
- ☐ if (score = 80)
- ☐ if (score <= 80)
- ☐ if (score >= 80)

1-20 処理を途中で終了し、以降の処理を実行しないために使用するものを選択してください。

- ☐ return
- ☐ continueAll
- ☐ end
- ☐ stopFunction

2 タスク管理アプリ作成問題

/ 40

本設問は簡易的なウェブアプリ「タスク管理アプリ」の作成を前提としています。

4x10

別紙「タスク管理アプリ」の仕様に基づいて解答してください。

2-1 「追加」ボタンをクリックしたときに処理を実行する記述として、正しいものを選択してください。

- ☐ \$("#addButton").on("click", function () {});
- ☐ \$(".addButton").on("change", function () {});
- ☐ \$("#addButton").on("load", function () {});
- ☐ \$(".addButton").clickEvent(function () {});

2-2 ID が taskInput の入力欄から値を取得し、前後の空白を取り除く記述として、正しいものを選択してください。

- ☐ \$("#taskInput").text().clear()
- ☐ \$("#taskInput").html().space()
- ☐ \$("#taskInput").val().trim()
- ☐ \$("#taskInput").trim().value()

2-3 変数 `taskText` が空の場合に処理を行う条件式として、正しいものを選択してください。

- ☐ `if (taskText = "")`
- ☐ `if (taskText === "")`
- ☐ `if (taskText !== "")`
- ☐ `if (taskText >= "")`

2-4 新しい `li` 要素を作成し、`task-item` クラスを追加する記述として、正しいものを選択してください。

- ☐ `$("#li").addClass("#task-item")`
- ☐ `$("#li").class("task-item")`
- ☐ `$("<li>").addId("task-item")`
- ☐ `$("<li>").addClass("task-item")`

2-5 作成したタスク要素 `$taskItem` を、ID が `taskList` の要素内の最後に追加する記述を選択してください。

- ☐ `$("#taskList").append($taskItem)`
- ☐ `$("#taskList").remove($taskItem)`
- ☐ `$("#taskList").text($taskItem)`
- ☐ `$("#taskList").beforeClass($taskItem)`

2-6 「追加」 ボタンをクリックしたときに処理を実行する記述として、正しいものを選択してください。

- ☐ `$(".complete-button").load("click", function () {});`
- ☐ `$("#taskList").on("click", ".complete-button", function () {});`
- ☐ `$("#taskList").on(".complete-button", "click") {};`
- ☐ `$(".complete-button").addEvent("complete") {};`

2-7 クリックされた「完了」 ボタンが含まれている、`.task-item` を取得する記述を選択してください。

- ☐ `$(this).find(".task-item")`
- ☐ `$(this).children(".task-item")`
- ☐ `$(this).closest(".task-item")`
- ☐ `$(this).next(".task-item")`

2-8 タスクの完了・未完了を切り替えるため、`completed` クラスを追加または削除する記述を選択してください。

- ☐ `$taskItem.toggleClass("completed")`
- ☐ `$taskItem.remove("completed")`
- ☐ `$taskItem.text("completed")`
- ☐ `$taskItem.toggleId("completed")`

2-9 「削除」 ボタンをクリックしたタスクだけを削除する記述として、正しいものを選択してください。

- ☐ \$("#taskList").remove()
- ☐ \$(".task-item").remove()
- ☐ \$(this).remove("#taskList")
- ☐ \$(this).closest(".task-item").remove()

2-10 未完了のタスク数を取得する記述として、正しいものを選択してください。

- ☐ \$("#taskList").length
- ☐ \$("#taskList .task-item").not(".completed").length
- ☐ \$(".completed").length
- ☐ \$("#remainingCount").val()