

Syntaxe et intégration de JS

Table des matières

I. Contexte	3
II. Syntaxe de JavaScript et 2 méthodes possibles d'intégration en HTML	3
A. Syntaxe de JavaScript et 2 méthodes possibles d'intégration en HTML	3
B. Exercice : Quiz	8
III. Appliquer la convention et éviter une erreur courante	9
A. Appliquer la convention et éviter une erreur courante	9
B. Exercice : Quiz	11
IV. Essentiel	12
V. Auto-évaluation	12
A. Exercice	12
B. Test	13
Solutions des exercices	15

I. Contexte

Durée : 1h30

Prérequis : Connaître les bases du HTML/CSS

Contexte

Vous connaissez plusieurs caractéristiques essentielles de JavaScript. C'est un langage côté client, classé parmi les langages interprétés et qui va de pair avec le standard HTML/CSS. Mais une question se pose : comment intégrer un script JavaScript dans un document HTML ?

On le sait, un script JavaScript peut être intégré dans un code HTML, et va pouvoir, via le Document Object Model (DOM), accéder à une représentation des différents éléments de la page. JavaScript va donc nous permettre de modifier l'agencement des éléments HTML, donc d'ajouter de l'interactivité et du dynamisme aux sites web.

Il existe 2 méthodes permettant d'insérer des scripts JavaScript en utilisant la balise HTML **<script>**. Il va donc nous falloir utiliser la syntaxe HTML pour insérer un script. La première méthode consiste à directement intégrer le script dans le code HTML. La deuxième méthode revient à écrire notre script dans un fichier **.js** et à l'intégrer via un lien. L'intégration peut générer des erreurs si elle n'est pas faite au bon endroit dans le code HTML.

Nous ferons en premier lieu un point sur la syntaxe globale de JavaScript, puis nous aborderons ces 2 méthodes d'intégration d'un script JavaScript. Enfin, nous verrons comment intégrer d'une manière conventionnelle les scripts, et comment éviter une erreur liée à l'endroit où nous les intégrons. Prenez donc bien le temps de visualiser et de tester les exemples de codes qui vous permettront de bien assimiler la notion. Vous pourrez aussi visualiser les vidéos qui mettent en image les exemples pris dans ce cours, avec l'IDE Replit.

II. Syntaxe de JavaScript et 2 méthodes possibles d'intégration en HTML

A. Syntaxe de JavaScript et 2 méthodes possibles d'intégration en HTML

Avant de nous intéresser à l'intégration d'un script JavaScript dans un code HTML, parlons de la syntaxe générale de JavaScript.

Syntaxe générale de JavaScript

Pour faire simple, voici une liste non exhaustive de règles de syntaxe en JavaScript :

- La hiérarchie des instructions est représentée via le système d'indentation (tab),
- Chaque ligne d'instructions se termine par un « ; »,
- Les paramètres d'une fonction sont inscrits entre parenthèses et séparés par des virgules : (**paramètre 1, paramètre 2**),
- Le corps d'une fonction est inscrit entre accolades : **{corps d'une fonction}**,
- Les commentaires sont précédés des caractères : « // »,
- Etc.

Voici quelques règles basiques de syntaxe, mais vous pourrez découvrir les autres au fur et à mesure de votre apprentissage de JavaScript.

Intégration d'un script directement dans un document HTML (intégration interne)

La première méthode que nous pouvons utiliser pour intégrer un script JavaScript est de l'écrire directement dans notre document HTML (intégration interne). Pour cela, nous allons utiliser la balise **<script>**. Prenons un exemple d'un script HTML tout simple :

Exemple

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9
10 <style>
11   * {
12     font-family: Verdana, Geneva, Tahoma, sans-serif;
13     font-size: large;
14   }
15
16   header {
17     display: flex;
18     flex-direction: row;
19     justify-content: center;
20     align-items: center;
21     background-color: aquamarine;
22
23   }
24
25   article {
26     border: 1px black solid;
27     padding: 0% 2% 0% 2%;
28     text-align: justify;
29     margin: 2% 0% 2% 0%;
30   }
31
32   footer {
33     background-color: aquamarine;
34     height: 50px;
35
36   }
37 </style>
38
39 <body>
40   <header>
41     <h1>Blog</h1>
42   </header>
43   <section>
44     <article>
45       <p>
46         Voici
47       </p>
48     </article>
49     <article>
50       <p>

```

```
51         "Votre texte"
52     </p>
53 </article>
54 </section>
55
56 <footer></footer>
57 </body>
58
59 </html>
```

Ici, on a directement inséré le code css dans le fichier HTML pour faire plus simple (dans la réalité, il est mieux de créer des fichiers **.css** et de les intégrer via un lien dans la balise **<head>**). Maintenant, écrivons un script JavaScript très simple qui déclenche une alerte. Pour cela, nous allons insérer une balise **<script>** :

Exemple

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8 </head>
9
10 <style>
11     * {
12         font-family: Verdana, Geneva, Tahoma, sans-serif;
13         font-size: large;
14     }
15
16     header {
17         display: flex;
18         flex-direction: row;
19         justify-content: center;
20         align-items: center;
21         background-color: aquamarine;
22     }
23
24
25     article {
26         border: 1px black solid;
27         padding: 0% 2% 0% 2%;
28         text-align: justify;
29         margin: 2% 0% 2% 0%;
30     }
31
32     footer {
33         background-color: aquamarine;
34         height: 50px;
35     }
36
37 </style>
38
39 <body>
40     <header>
```

```

41     <h1>Blog</h1>
42 </header>
43 <section>
44     <article>
45         <p>
46             "Votre texte"
47         </p>
48     </article>
49     <article>
50         <p>
51             "Votre texte"
52         </p>
53     </article>
54 </section>
55
56 <footer></footer>
57
58 <script>
59     //script JavaScript
60     alert("Hello World !");
61 </script>
62
63 </body>
64
65 </html>

```

Si on exécute ce code, on peut voir qu'il fonctionne, l'alerte se déclenche.

Maintenant, voyons comment insérer un lien vers un fichier **.js**.

Intégration d'un script via un lien dans un document HTML (intégration externe)

Il faut simplement utiliser la balise **<script>** et définir l'attribut **src** sur l'adresse du fichier **.js**. On va donc, dans notre exemple, créer un dossier « *scripts* » dans lequel on va créer un fichier « **script.js** ». Puis dans le fichier **script.js**, on insère l'instruction :

```
1 alert("Hello World !");
```

Maintenant, on peut modifier le fichier **.html** pour qu'il intègre le script :

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8 </head>
9
10 <style>
11     * {
12         font-family: Verdana, Geneva, Tahoma, sans-serif;
13         font-size: large;
14     }
15
16     header {
17         display: flex;
18         flex-direction: row;

```

```
19     justify-content: center;
20     align-items: center;
21     background-color: aquamarine;
22
23 }
24
25 article {
26     border: 1px black solid;
27     padding: 0% 2% 0% 2%;
28     text-align: justify;
29     margin: 2% 0% 2% 0%;
30 }
31
32 footer {
33     background-color: aquamarine;
34     height: 50px;
35 }
36 }
37 </style>
38
39 <body>
40     <header>
41         <h1>Blog</h1>
42     </header>
43     <section>
44         <article>
45             <p>
46                 “Votre texte”
47             </p>
48         </article>
49         <article>
50             <p>
51                 “Votre texte”
52             </p>
53         </article>
54     </section>
55
56     <footer></footer>
57
58     <script src = "scripts/script.js"></script>
59
60 </body>
61
62 </html>
```

En règle générale, lorsque nous intégrons des scripts JavaScript longs dans un code HTML, nous créons des fichiers **.js** que nous pouvons classer dans le projet, avant d'insérer un lien qui permet de les exécuter. C'est une méthode à privilégier. Elle permet par ailleurs d'isoler les scripts JavaScript, il sera donc plus simple de s'y retrouver. Comme vous pouvez le constater, la procédure est assez simple.

Complément Syntaxe des chemins relatifs

Dans notre exemple, nous avons utilisé l'attribut **src** en spécifiant un chemin relatif. Si le document HTML se trouve dans un sous dossier, il faudra sortir du sous-dossier pour se placer dans le dossier parent, pour ensuite se déplacer dans le sous-dossier « *scripts* ». Pour sortir du dossier et donc se placer dans le dossier parent, il faut commencer le chemin par : « *../* ». Cette inscription permet de se déplacer dans le dossier parent.

B. Exercice : Quiz

[solution n°1 p.17]

Question 1

Quelle balise permet d'insérer du JavaScript dans un code HTML ?

- ☐ <article>
- ☐ <script>
- ☐ <link>

Question 2

Quelle est l'extension d'un fichier script JavaScript ?

- ☐ .js
- ☐ .script
- ☐ .php

Question 3

Quelle méthode d'intégration de JS est préconisée lorsqu'on réalise un site web ?

- ☐ L'intégration « *interne* » via la balise <script> en écrivant le JS directement dans la balise
- ☐ .php
- ☐ L'intégration « *externe* » via la balise <script> avec src défini sur le chemin d'un fichier .js

Question 4

Dans un chemin relatif, pour sortir d'un répertoire et se placer dans le répertoire parent :

- ☐ Le chemin commence par : « ../ »
- ☐ Le chemin commence par : « ./ »
- ☐ Le chemin commence par : « / »

Question 5

En JavaScript, chaque ligne d'instructions se termine généralement par :

- ☐ Une virgule : « , »
- ☐ Un point-virgule : « ; »
- ☐ Rien

III. Appliquer la convention et éviter une erreur courante

A. Appliquer la convention et éviter une erreur courante

Vous connaissez maintenant la syntaxe très simple permettant d'intégrer un script JavaScript dans un document HTML. Mais où l'intégrer ? Dans le dernier exemple, vous avez constaté qu'on intègre le script à la fin de la balise **<body>**. Cependant, d'un point de vue conventionnel, il est conseillé d'insérer les scripts (avec l'attribut **src**) dans la balise **<head>**. Avec le script JavaScript que nous avons écrit, cela ne posera pas de problème. Nous allons au passage déplacer le contenu de la balise **<style>** dans un fichier **.css** à part, contenu dans un répertoire « *styles* » :

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="styles/style.css">
8   <script src="scripts/script.js"></script>
9   <title>Document</title>
10 </head>
11
12 <body>
13   <header>
14     <h1>Blog</h1>
15   </header>
16   <section>
17     <article>
18       <p>
19         "Votre texte"
20       </p>
21     </article>
22     <article>
23       <p>
24         "Votre texte"
25       </p>
26     </article>
27   </section>
28
29   <footer></footer>
30
31 </body>
32
33 </html>

```

Attention Ordre des métadonnées

Il est préférable, voire indispensable, de placer dans le **head** les liens vers les feuilles de style avant les scripts, pour afficher correctement et rapidement le visuel du site.

Dans ce cas-ci, ça ne pose pas de problème, mais admettons qu'on modifie notre HTML pour y insérer un bouton qui déclenchera une action :

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="styles/style.css">

```

```

8   <script src="scripts/script.js"></script>
9   <title>Document</title>
10  </head>
11
12  <body>
13    <header>
14      <h1 id="blog">Blog</h1>
15      <button id="btn">Bouton</button>
16    </header>
17
18    <section>
19      <article>
20        <p>
21          "Votre texte"
22        </p>
23      </article>
24
25      <article>
26        <p>
27          "Votre texte"
28        </p>
29      </article>
30
31    </section>
32
33    <footer></footer>
34
35  </body>
36
37 </html>

```

On a donc un bouton dans notre balise **<header>**, mais il nous faut modifier notre **script.js** pour qu'une action puisse se déclencher lors du clic. On le modifie ainsi :

```

1  const btn = document.getElementById("btn");
2
3  btn.onclick = () => alert("Bouton cliqué");

```

Ce script permet simplement de récupérer l'élément HTML **button** pour qu'il déclenche une alerte lorsqu'il est cliqué. Ne vous inquiétez pas, le but n'est pas de comprendre comment fonctionne ce script, mais simplement de voir ce qui se passe. Si on actualise notre page HTML depuis notre navigateur, on peut voir que rien ne se passe. D'ailleurs, si on affiche la console, une erreur est présente :

```

1  TypeError: null is not an object (evaluating 'btn.onclick = () => alert("Bouton cliqué");')

```

Mais pourquoi cela ne fonctionne pas ? Tout simplement parce que lors du traitement du fichier HTML, le script étant indiqué en début du fichier (dans la balise **<head>**), il est exécuté avant que le code HTML soit chargé intégralement. Or, le script JavaScript fait référence à un bouton ayant pour **id** : **"btn"**. Mais comme la page HTML n'est pas chargée, ce bouton n'existe pas encore, donc, la console indique que le script fait référence à un élément HTML qui n'existe pas. Comment résoudre ce problème ? Une solution très simple consiste à indiquer l'attribut **defer** de la balise **<script>** qui pour résumer simplement permet d'indiquer que le script doit être exécuté de manière différée, donc une fois que le document est analysé. Cet attribut doit être utilisé uniquement dans le cas où un attribut **src** est spécifié (donc pas dans le cas de la méthode n° 1). Avec cette intégration, tout fonctionne :

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">

```

```
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="styles/style.css">
8   <script src="scripts/script.js" defer></script>
9   <title>Document</title>
10  </head>
11
12  <body>
13    <header>
14      <h1 id="blog">Blog</h1>
15      <button id="btn">Bouton</button>
16    </header>
17
18    <section>
19      <article>
20        <p>
21          "Votre texte"
22        </p>
23      </article>
24
25      <article>
26        <p>
27          "Votre texte"
28        </p>
29      </article>
30
31    </section>
32
33    <footer></footer>
34
35  </body>
36
37 </html>
```

B. Exercice : Quiz

[solution n°2 p.18]

Question 1

Dans quelle balise HTML est-il conventionnel d'intégrer le JS ?

- ☐ <body>
- ☐ <header>
- ☐ <head>

Question 2

Quel attribut permet d'exécuter un script après le chargement de la page HTML ?

- ☐ src
- ☐ defer
- ☐ integrity

Question 3

Si un script est intégré dans le **<head>** (sans attribut **defer**) et qu'il contient des références vers des éléments HTML :

- ☐ Il est probable qu'il y ait des erreurs dues à l'exécution du script avant le chargement des éléments HTML
- ☐ Aucune erreur ne devrait avoir lieu

Question 4

L'attribut **defer** ne doit pas être utilisé avec l'intégration interne d'un script.

- ☐ Vrai
- ☐ Faux

Question 5

Dans la balise **<head>**, il est important de placer :

- ☐ Les liens vers les feuilles de style (.css) avant les balises `<script>`
- ☐ Les liens vers les feuilles de style (.css) après les balises `<script>`

IV. Essentiel

Nous avons vu comment intégrer un script JavaScript dans une page HTML. Nous avons compris que c'est la balise **<script>** qui permet d'intégrer du JS. Il existe 2 méthodes permettant d'intégrer des scripts JS.

La méthode interne revient à écrire directement le code JavaScript dans le code HTML, à l'intérieur d'une balise **<script>**. Cette méthode peut être pratique pour des courts extraits de codes, mais n'est pas recommandée car le code HTML peut vite devenir illisible.

La méthode externe quant à elle revient à écrire le script dans un fichier **.js** à part. On utilise la balise **<script>** dans le code HTML et l'attribut **src** pour spécifier un chemin vers le fichier. Cette méthode est préconisée, car elle permet de rendre plus lisible le code HTML et par ailleurs d'organiser ses scripts dans des fichiers classés dans des répertoires.

D'un point de vue conventionnel, c'est dans la balise **<head>** qu'on va intégrer nos scripts (via des balises scripts avec bien évidemment la méthode externe). Il vaut mieux intégrer les scripts après les liens vers les feuilles de style (.css). Cependant, dans le cas presque systématique où le script JS essaye de récupérer des éléments HTML, des erreurs vont se produire. En effet, les scripts vont être exécutés avant le chargement de la page. Donc pour éviter ce problème, nous pouvons utiliser l'attribut **defer** qui permet d'exécuter un script après le chargement de la page.

V. Auto-évaluation

A. Exercice

Vous cherchez à réaliser un site de e-commerce. Vous avez déjà réalisé une maquette très simple de la page d'accueil de votre site, mais elle est statique. Vous aimeriez bien rajouter des animations visuelles grâce à JavaScript. Voici votre maquette :

[cf. Cas pratique - template.zip]

Question 1

[solution n°3 p.19]

Créez un répertoire « *scripts* » contenant un script JavaScript (un fichier **.js**). Ce script contiendra juste un **alert** avec la chaîne « *Site de e-commerce* ». Puis intégrez d'une manière conventionnelle le script dans votre doc HTML.

Question 2

[solution n°4 p.19]

Vous souhaitez que le script s'exécute après le chargement de la page HTML. Modifier l'intégration du script pour que cela soit possible. La modification ne sera pas très visible avec Replit, mais cette modification aura un effet considérable si le code JavaScript fait référence à des éléments HTML.

B. Test**Exercice 1 : Quiz**

[solution n°5 p.19]

Question 1

On utilise la balise **<link>** pour intégrer un script JS.

- ☐ Vrai
- ☐ Faux

Question 2

Quel attribut de la balise **<script>** permet de spécifier le chemin d'un script ?

- ☐ src
- ☐ integrity
- ☐ defer

Question 3

Voici un code HTML :

```
1 1 <!DOCTYPE html>
2 2 <html lang="en">
3 3 <head>
4 4 <meta charset="UTF-8">
5 5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 7 <link rel="stylesheet" href="styles/style.css">
8 8 <title>Site de E Commerce</title>
9 9 </head>
10 10
11 11 <body>
12 12 <section id="presentation">
13 13 <article>
14 14 <p>
15 15 Présentation du site
16 16 </p>
17 17 </article>
18 18
19 19 <script src="script.js"></script>
20 20 </section>
21 21 </body>
22 22
23 23 </html>
```

L'intégration du script est conventionnelle.

- ☐ Vrai
- ☐ Faux

Question 4

Voici un code HTML :

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <script src="script.js" defer></script>
8   <link rel="stylesheet" href="styles/style.css">
9   <title>Site de E Commerce</title>
10 </head>
11
12 <body>
13   <section id="presentation">
14     <article>
15       <p>
16         Présentation du site
17       </p>
18     </article>
19
20 </body>
21
22 </html>

```

L'intégration est placée de la manière la plus optimale.

- ☐ Vrai
- ☐ Faux

Question 5

Voici un code HTML :

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="styles/style.css">
8   <title>Site de E Commerce</title>
9 </head>
10
11 <body>
12   <section id="presentation">
13     <article>
14       <p>
15         Présentation du site
16       </p>
17     </article>
18
19     <script>alert("Hello World !")</script>
20
21 </body>
22
23 </html>

```

L'intégration du script est :


- ☐ Interne
- ☐ Externe
- ☐ Ulérieur

Solutions des exercices

Exercice p. 8 Solution n°1**Question 1**

Quelle balise permet d'insérer du JavaScript dans un code HTML ?


- ☐ <article>
- ☒ <script>
- ☐ <link>

 La balise **<script>** permet d'intégrer du JavaScript dans un document HTML.

Question 2

Quelle est l'extension d'un fichier script JavaScript ?


- ☒ .js
- ☐ .script
- ☐ .php

 Un fichier script JavaScript a l'extension **.js**.

Question 3

Quelle méthode d'intégration de JS est préconisée lorsqu'on réalise un site web ?


- ☐ L'intégration « *interne* » via la balise <script> en écrivant le JS directement dans la balise
- ☐ .php
- ☒ L'intégration « *externe* » via la balise <script> avec src défini sur le chemin d'un fichier .js

 Il est préconisé d'intégrer les scripts JS d'une manière externe, c'est-à-dire en rédigeant le code dans un fichier **.js** et en l'intégrant dans le HTML via la balise **<script>** avec l'attribut **src** défini sur le chemin du fichier **.js**.

Question 4


Dans un chemin relatif, pour sortir d'un répertoire et se placer dans le répertoire parent :

- ☒ Le chemin commence par : « ../ »
- ☐ Le chemin commence par : « ./ »
- ☐ Le chemin commence par : « / »

 Dans un chemin relatif, pour sortir d'un répertoire courant et se placer dans le répertoire parent, il faut simplement que le chemin commence par « ../ ».

Question 5


En JavaScript, chaque ligne d'instructions se termine généralement par :

- ☐ Une virgule : « , »
- ☒ Un point-virgule : « ; »
- ☐ Rien
-  En JavaScript, chaque ligne d'instructions se termine généralement par un point-virgule : « ; ».

Exercice p. 11 Solution n°2


Question 1

Dans quelle balise HTML est-il conventionnel d'intégrer le JS ?

- ☐ <body>
- ☐ <header>
- ☒ <head>
-  D'un point de vue conventionnel, c'est dans la balise **<head>**, dédiée aux métadonnées, qu'il faut intégrer le script JavaScript (via un lien).


Question 2

Quel attribut permet d'exécuter un script après le chargement de la page HTML ?

- ☐ src
- ☒ defer
- ☐ integrity
-  L'attribut **defer** permet d'exécuter un script après le chargement d'une page HTML.

Question 3

Si un script est intégré dans le **<head>** (sans attribut **defer**) et qu'il contient des références vers des éléments HTML :


- ☒ Il est probable qu'il y ait des erreurs dues à l'exécution du script avant le chargement des éléments HTML
- ☐ Aucune erreur ne devrait avoir lieu
-  Si un script est intégré en début de document HTML (dans le **<head>** par exemple si le code est correct d'un point de vue conventionnel), alors il risque de se produire des erreurs si le script contient des références vers les éléments HTML (via le Document Object Model) et que l'attribut **defer** n'est pas spécifié. En effet, si la page HTML n'a pas eu le temps de charger avant l'exécution du script, alors la console renverra que des objets HTML mentionnés n'existent pas.

Question 4

L'attribut **defer** ne doit pas être utilisé avec l'intégration interne d'un script.

☒ Vrai

☐ Faux


 L'attribut **defer** ne doit être utilisé que lorsque l'attribut **src** est spécifié, donc avec l'intégration externe d'un script.

Question 5

Dans la balise **<head>**, il est important de placer :

☒ Les liens vers les feuilles de style (.css) avant les balises **<script>**

☐ Les liens vers les feuilles de style (.css) après les balises **<script>**

 Il est préconisé de placer dans la balise **<head>** les liens vers les feuilles de style (c'est-à-dire les fichiers **.css**) avant les balises **<scripts>**.

p. 12 Solution n°3

Le projet suivant intègre correctement le script :

[cf. Cas pratique - 1.zip]

[cf.]

p. 12 Solution n°4

Il faut simplement spécifier dans l'intégration l'attribut **defer** :

[cf. Cas pratique - 2.zip]

[cf.]


Exercice p. 13 Solution n°5

Question 1

On utilise la balise **<link>** pour intégrer un script JS.

☐ Vrai

☒ Faux

 C'est la balise **<script>** qui est utilisée pour intégrer un script JS dans un code HTML.


Question 2

Quel attribut de la balise **<script>** permet de spécifier le chemin d'un script ?

☒ src

☐ integrity

☐ defer

 L'attribut **src** permet de spécifier le chemin d'un script avec la balise **<script>**.

Question 3

Voici un code HTML :

```

1 1 <!DOCTYPE html>
2 2 <html lang="en">
3 3 <head>
4 4 <meta charset="UTF-8">
5 5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 7 <link rel="stylesheet" href="styles/style.css">
8 8 <title>Site de E Commerce</title>
9 9 </head>
10 10
11 11 <body>
12 12 <section id="presentation">
13 13 <article>
14 14 <p>
15 15 Présentation du site
16 16 </p>
17 17 </article>
18 18
19 19 <script src="script.js"></script>
20 20 </section>
21 21 </body>
22 22
23 23 </html>

```

L'intégration du script est conventionnelle.

☐ Vrai

☒ Faux

 L'intégration du script est réalisée dans la balise **<body>**. Pour que l'intégration soit conventionnelle, il faut la placer dans la balise **<head>**.

Question 4

Voici un code HTML :

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <script src="script.js" defer></script>
8   <link rel="stylesheet" href="styles/style.css">
9   <title>Site de E Commerce</title>
10 </head>
11
12 <body>
13   <section id="presentation">
14     <article>
15       <p>
16         Présentation du site
17       </p>
18     </article>
19
20 </body>


```

```
21  
22 </html>
```

L'intégration est placée de la manière la plus optimale.

☐ Vrai

☒ Faux

 Le script est intégré avant le lien vers la feuille de style (.css). Il vaut mieux que ce soit l'inverse pour éviter des problèmes concernant l'affichage du visuel du site.

Question 5

Voici un code HTML :


```
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4     <meta charset="UTF-8">  
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7     <link rel="stylesheet" href="styles/style.css">  
8     <title>Site de E Commerce</title>  
9 </head>  
10  
11 <body>  
12     <section id="presentation">  
13         <article>  
14             <p>  
15                 Présentation du site  
16             </p>  
17         </article>  
18  
19         <script>alert("Hello World !")</script>  
20  
21 </body>  
22  
23 </html>
```

L'intégration du script est :

☒ Interne

☐ Externe

☐ Ulérieur

 L'intégration du script est interne puisqu'on insère le code JavaScript directement dans le document HTML, il n'y a pas de lien vers un fichier .js externe. Cette méthode peut être fonctionnelle pour des codes courts, mais n'est pas recommandée car elle peut vite rendre les codes HTML trop chargés et illisibles.