*Chapitre 1*

# *Introduction à JavaScript*

Le JavaScript (JS) est un langage de programmation, un langage de script, principalement employé dans les pages Web interactives pour le rendu côté client, mais depuis quelques années, son utilisation côté serveur s'est également répandue, avec Node.js par exemple. Javascript est un langage de programmation. Il s'agit d'une implémentation d'ECMAscript. À la base, il s'exécute sur l'ordinateur client et permet de rendre les pages Web interactives. Avec Javascript on peut accéder aux objets du DOM et les gérer librement, traiter les formulaires et créer des animations complexes. Les objets sont chacun équipés de constructeurs permettant de créer leurs propriétés notamment une propriété de prototypage.

***OBJECTIFS***

- Découvrir les origines de JavaScript.
- Découvrir comment on peut utiliser JavaScript sur une page Web.

# *Notes*

*Leçon 1-A*

# *Bref historique et définition*

JavaScript apparait comme l'un des langages les plus utilisés aujourd'hui sur le Web. L'histoire la plus courte qu'on raconte sur ce langage est qu'il a été créé en dix jours à peine par Brendan Eich pour le compte de Netscape en 1995.

***OBJECTIFS***

À la fin de la présente leçon, vous serez en mesure :

- de définir c'est quoi le JavaScript.

Dans le cours ***Développement Web I***, vous avez découvert l'interaction entre le HTML et le CSS afin de présenter des pages Web bien conçues et agréables à regarder. Ces deux langages, bien qu'ils soient différents, interagissent ensemble afin de structurer et présenter des pages Web statiques agréables à visiter. Lorsque Tim Berners-Lee a inventé le premier navigateur Web en 1990, les pages Web qu'il affichait manquaient encore de dynamisme et d'interactions plus conviviales.

- ***HTML*** est un langage de balises utilisé pour structurer et donner du sens au contenu d'une page Web.
- ***CSS*** est un langage de règles de style utilisé pour metre en forme le contenu HTML.
- ***JavaScript*** est un langage de programmation qui permet de créer du contenu mis-à-jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, et tout ce à quoi on peut penser.

Bon, peut-être que JavaScript ne peut pas tout faire mais, vous êtes en mesure de faire bien des choses avec ce langage.

Les trois couches se superposent naturellement afin de donner des pages Web qui, en plus d'être agréables à première vue, permettent aux internautes de pouvoir interagir avec vos pages Web.

### *JavaScript, c'est quoi ?*

JavaScript est un langage de programmation interprété par le navigateur. Le JavaScript est un ***langage client***, c'est-à-dire qu'il est exécuté par le navigateur de l'utilisateur lorsque la page Web est chargée ou lors d'une interaction de l'utilisateur sur votre page. Il a pour but de dynamiser les sites Internet.

### *Origine du langage JavaScript*

JavaScript a été créé en 1995 par Brendan Eich. Il a été standardisé sous le nom d'ECMAScript en juin 1997 par Ecma International dans le standard ECMA-262. Le standard ECMA-262 en est actuellement à sa 8$^{ième}$ édition. JavaScript n'est depuis qu'une implémentation d'ECMAScript, celle mise en œuvre par la fondation Mozilla. L'implémentation d'ECMAScript par Microsoft, dans Internet Explorer, jusqu'à sa version 9, se nomme JScript, tandis que celle d'Adobe Systems se nomme ActionScript.

Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies au cœur du World Wide Web. Le langage JavaScript permet des pages web interactives, et à ce titre est une partie essentielle des applications Web. Une grande majorité des sites Web l'utilisent, et la majorité des navigateurs Web disposent d'un moteur JavaScript dédié pour l'interpréter.

### *Que peut-on faire avec JavaScript*

La plupart du code Javascript se trouve dans des pages Web, et sert donc à dire comment la page Web doit réagir. Cela marche ainsi :

1. L'utilisateur clique sur un lien ou saisi une adresse dans son navigateur.
2. Son navigateur charge la page Web. Il voit le texte, les couleurs, les images.
3. Si la page Web contient du code Javascript, le navigateur lit le code Javascript et suit les instructions du code.

Généralement le code Javascript dans une page Web sert à :

- Faire bouger, apparaitre ou disparaitre des éléments de la page, un titre, un menu, un paragraphe, une image…
- Mettre à jour des éléments de la page sans recharger la page, changer le texte, recalculer un nombre, etc.
- Demander au serveur une nouvelle partie d'une page et l'insérer dans la page en cours, sans la recharger.
- Attendre que l'utilisateur pose un geste, comme cliquer sur la page, saisir au clavier, bouger la souris, et réagir au geste posé.

Le code JavaScript sert donc à donner du dynamisme à la page. Sans lui, la page ressemble à une page de livre, un peu animée, grâce à l'utilisation du langage CSS, mais qui ne change pas beaucoup.

Certains sites Web ne pourraient tout simplement pas fonctionner sans JavaScript. C'est le cas de Facebook, Youtube ou Twitter qui utilisent le langage pour presque tous leurs affichages. La page de recherche de Google, en revanche, peut fonctionner sans JavaScript.

### *Récapitulation*

- JavaScript est un langage de programmation qui a été créé en 1995 par Brendan Eich.
- JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page Web.
- À chaque fois qu'une page Web fait plus que simplement afficher du contenu statique, JavaScript a de bonnes chances d'être impliqué.
- JavaScript est la troisième couche des technologies standards du Web, les deux premières, HTML et CSS, ont été couvertes dans le cours ***Développement Web I***.
- L'utilisation de ces trois langages, HTML, CSS et JavaScript, sur une même page Web peut apporter de très beaux résultats pour l'utilisateur qui visite votre page Web.

# *Notes*

*Leçon 1-B*

# *Où écrit-on le code JavaScript ?*

L'insertion de code JavaScript dans vos pages Web peut se faire de trois façons différentes :

- À l'intérieur de l'élément head d'une page HTML.
- À l'intérieur de l'élément body d'une page HTML.
- En utilisant un fichier .js inclus dans les dossiers utilisés par votre site Web.

Dans cette leçon, nous allons examiner ensemble ces trois méthodes d'insertion d'instructions JavaScript dans vos pages HTML.

***OBJECTIFS***

À la fin de la présente leçon, vous serez en mesure :

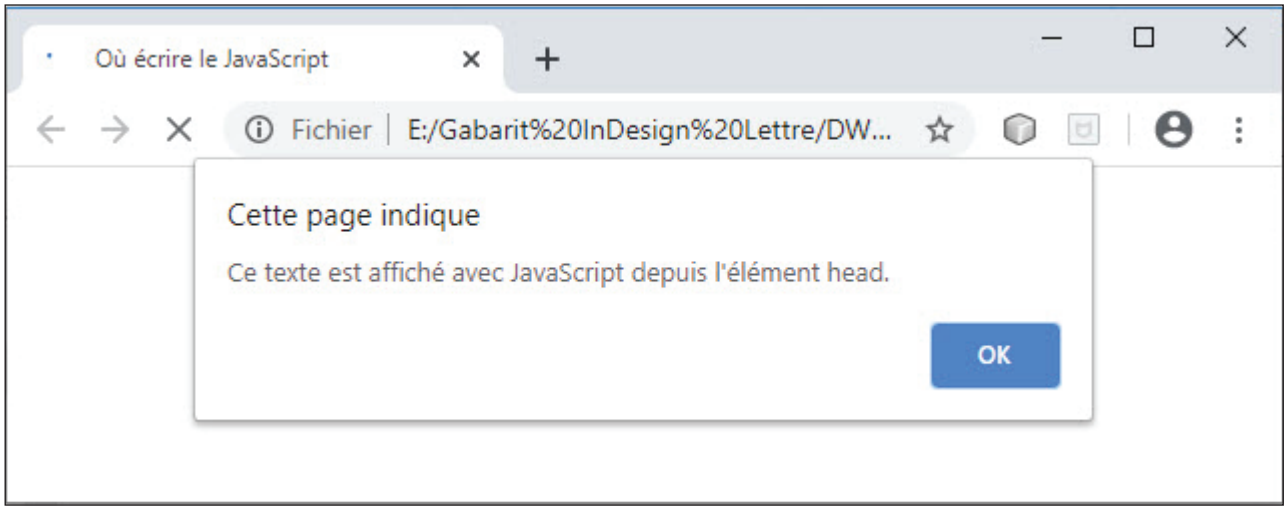
- de connaitre les façons d'insérer des éléments JavaScript dans vos pages HTML.
- de comprendre l'ordre d'exécution des éléments JavaScript dans une page HTML.

### *Première méthode : à l'intérieur de l'élément head d'une page HTML*

Examinez le prochain bloc de code.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=<device-width>, initial-scale=1.0">
    <title>Où écrire le JavaScript</title>
    <script>
        alert("Ce texte est affiché avec JavaScript depuis l'élément head.");
    </script>
</head>
    <body>
        <h1>On peut écrire le JavaScript dans...</h1>
        <ul>
            <li>L'élément head d'un fichier HTML.</li>
            <li>L'élément body d'un fichier HTML.</li>
            <li>Un fichier .js séparé.</li>
        </ul>
    </body>
</html>
```

Ce premier exemple insère un élément ***<script>*** à l'intérieur de l'élément ***head*** de notre page HTML. Cet élément ***script*** contient une seule et unique instruction, la méthode ***alert()*** qui est utilisée afin d'afficher une boite de dialogue à l'écran, tout comme nous sommes en mesure de le faire avec tout autre langage informatique. Lorsque nous demandons l'affichage de notre page Web dans notre navigateur, nous obtenons le résultat suivant affiché à l'écran.



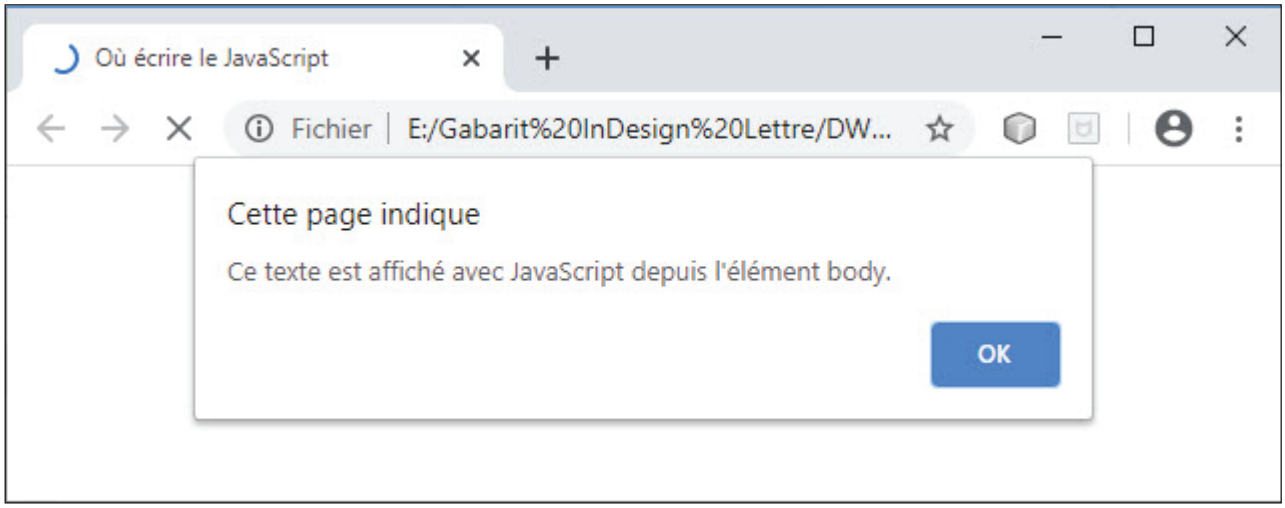

L'affichage complet du contenu de notre page Web sera fait après l'exécution du script, comme le montre la prochaine figure.


***Deuxième méthode : à l'intérieur de l'élément body d'une page HTML***

Examinez le prochain bloc de code.

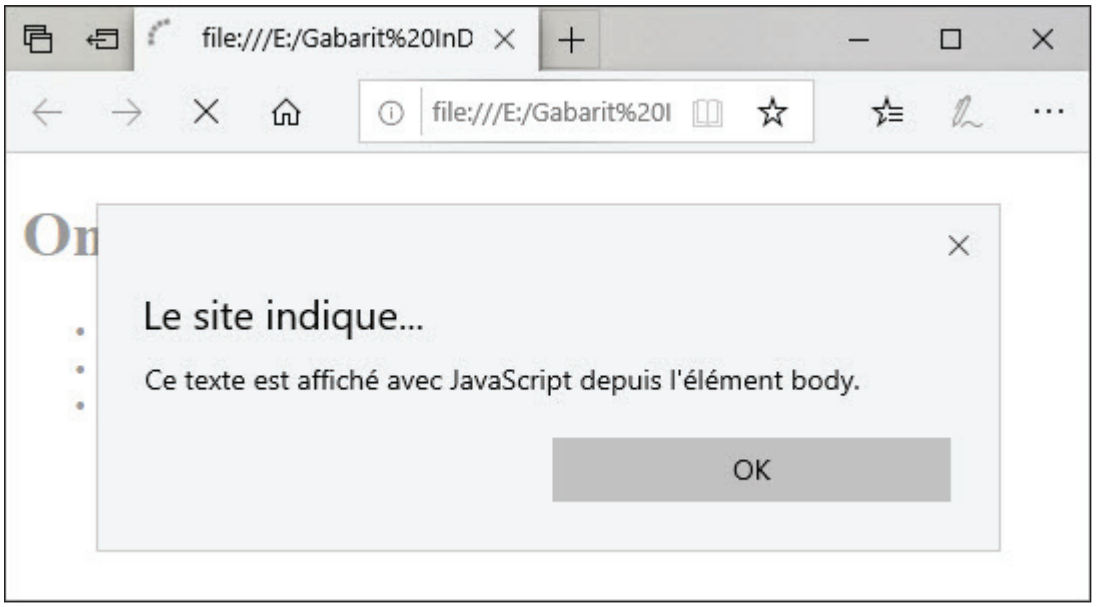
```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=<device-width>, initial-scale=1.0">
    <title>Où écrire le JavaScript</title>
</head>
    <body>
        <h1>On peut écrire le JavaScript dans...</h1>
        <ul>
            <li>L'élément head d'un fichier HTML.</li>
            <li>L'élément body d'un fichier HTML.</li>
            <li>Un fichier .js séparé.</li>
        </ul>
        <script>
            alert("Ce texte est affiché avec JavaScript depuis l'élément body.");
        </script>
    </body>
</html>
```

Dans ce nouveau bloc de code, nous avons déplacé notre élément script à la fin de notre élément body, tout juste au-dessus de sa balise de fermeture. La même méthode est utilisée afin d'afficher une boite de dialogue comme nous l'avons fait précédemment, seul le texte qui sera affiché a été modifié. Lorsque nous demandons l'affichage de notre page Web dans notre navigateur, nous obtenons le résultat de la prochaine figure affiché à l'écran.




Tout comme précédemment, le contenu de notre page sera affiché après un clic sur le bouton ***OK*** de la boite de dialogue affichée.

***Notez*** que ce résultat ne sera pas nécessairement le même selon le navigateur que vous utilisez. Jusqu'à présent, les résultats affichés proviennent de l'utilisation du navigateur ***Google Chrome***. Examinez la prochaine figure issue de l'affichage de la même page en utilisant le navigateur ***Microsoft Edge***.




Dans cette dernière figure, on peut remarquer que le contenu de notre page Web est affiché en arrière-plan de notre boite de dialogue. Certain navigateur interprète le code ***JavaScript*** dans l'ordre d'apparition du contenu de notre page Web. Dans ce dernier exemple, notre code apparait à la fin du contenu de notre élément ***body*** et, il est exécuté après le chargement de tous les éléments se trouvant juste avant notre élément ***script***.

### *Troisième méthode : dans un fichier .js séparé*

Premièrement, nous devons ajouter notre fichier ***.js*** à l'intérieur de notre projet Web. Dans cet exemple, nous avons nommé notre fichier ***script.js*** puis, nous avons saisi la ligne de code suivante à l'intérieur.

```
alert("Ce texte est affiché avec JavaScript depuis un fichier .js.");
```

Notez que nous n'avons pas utilisé de balise ***<script>***. Ce n'est pas nécessaire puisqu'il s'agit d'un fichier de script qui sera interprété comme tel lors de son appel. À l'intérieur de notre fichier HTML nous devons nous assurer de pointer vers notre fichier ***.js*** tout comme vous l'avez appris précédemment lors de l'utilisation des feuilles de styles en cascade. Nous devons donc ajouter l'instruction suivante à la fin de notre fichier HTML.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=<device-width>, initial-scale=1.0">
    <title>Où écrire le JavaScript</title>
</head>
    <body>
        <h1>On peut écrire le JavaScript dans...</h1>
        <ul>
            <li>L'élément head d'un fichier HTML.</li>
            <li>L'élément body d'un fichier HTML.</li>
            <li>Un fichier .js séparé.</li>
        </ul>
        <script src = "script.js"></script>
    </body>
</html>
```

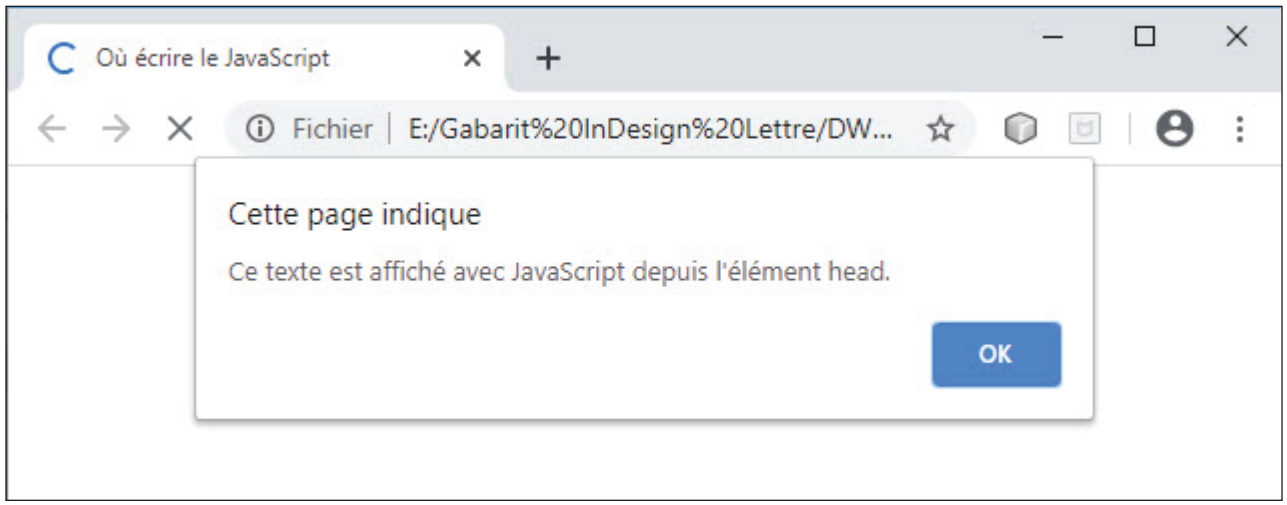
L'affichage de notre page Web se fera de la même façon que précédemment mais, avec un message différent.

### *Insertion de plusieurs scripts dans une page Web*

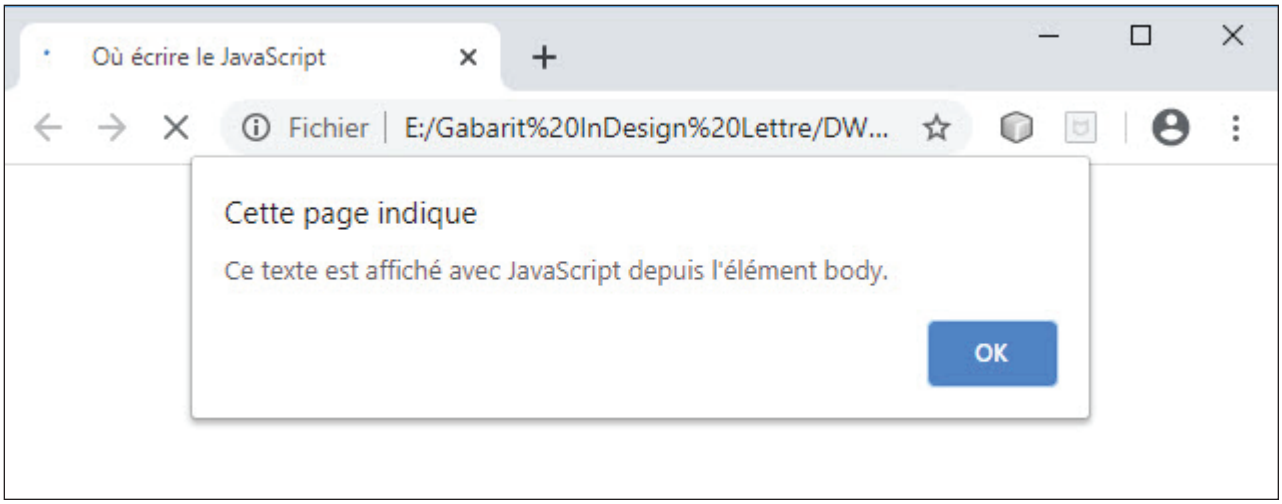
Il est aussi possible d'insérer différents éléments ***<script>*** dans la composition de notre page Web, comme le montre le prochain bloc de code.

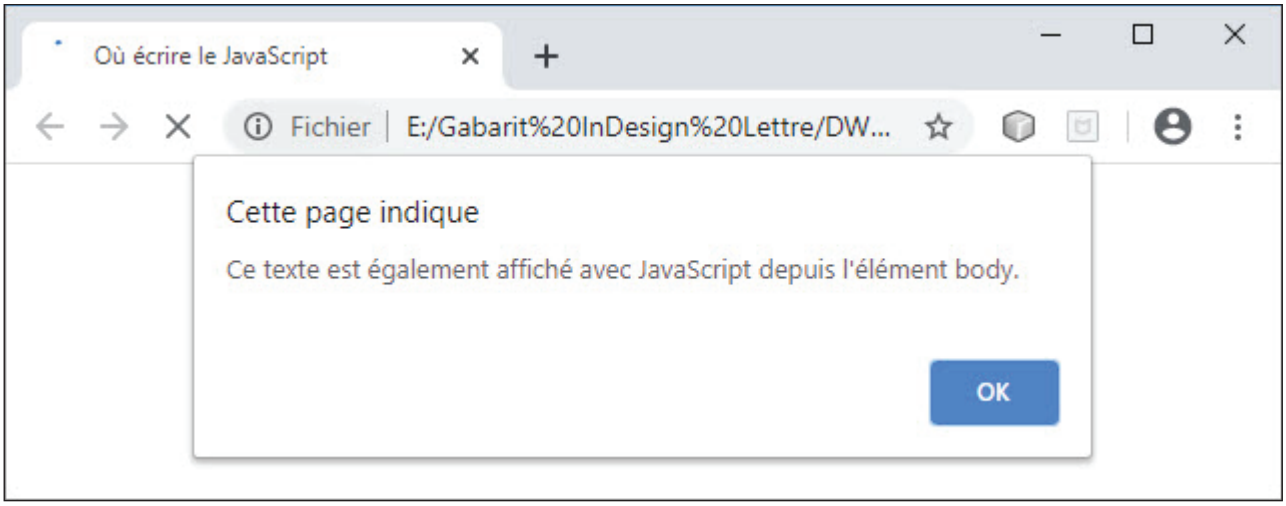
```html
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=<device-width>, initial-scale=1.0">
    <title>Où écrire le JavaScript</title>
    <script>
        alert("Ce texte est affiché avec JavaScript depuis l'élément head.");
    </script>
</head>
    <body>
        <script>
            alert("Ce texte est affiché avec JavaScript depuis l'élément body.");
        </script>
        <h1>On peut écrire le JavaScript dans...</h1>
        <ul>
            <li>L'élément head d'un fichier HTML.</li>
            <li>L'élément body d'un fichier HTML.</li>
            <li>Un fichier .js séparé.</li>
        </ul>
        <script>
            alert("Ce texte est également affiché avec JavaScript depuis l'élément body.");
        </script>
    </body>
</html>
```

Lorsque nous demandons l'affichage de notre page Web dans notre navigateur, nous obtenons le résultat de la prochaine figure affiché à l'écran.




Par la suite, les deux boites de dialogue suivantes s'affichent à tour de rôle à l'écran selon l'ordre d'apparition de nos scripts dans notre page HTML, comme le montre les prochaines figures.




Toutes ces méthodes d'écriture du code JavaScript sont bonnes. Par contre, la méthode à privilégier, spécialement dans de gros projets de site Web est la méthode qui utilise un fichier de ***script***, un fichier ***.js***.

### *Ajouter des commentaires*

Vous avez appris l'importance d'ajouter des commentaires dans votre code informatique. JavaScript ne fait pas exception à la règle. Il est possible d'ajouter des commentaires dans vos scripts de deux façons soit :

- des commentaires sur une seule ligne.
- des commentaires en bloc sur plusieurs lignes.

Un commentaire sur une seule ligne est précédé de deux barres obliques. Ces deux barres obliques informent JavaScript que tout le texte qui suit doit être ignoré.

```
<script>
    // Ceci est un commentaire sur une seule ligne.
    alert("Ce texte est affiché avec JavaScript depuis l'élément head.");
</script>
```

Un bloc de commentaire sur plusieurs lignes doit débuter par la séquence /* et se terminer par la séquence */.

```
<script>
    /* Voici un bloc de commentaire
    de plusieurs lignes qui
    ne sera pas interprété.*/
    alert("Ce texte est affiché avec JavaScript depuis l'élément head.");
</script>
```

Jusqu'à présent, vous avez sûrement remarqué que chacune des instructions écrites dans notre élément ***script*** se terminait par un point-virgule. Apportons une petite précision sur l'utilisation de ce point-virgule. Il n'est pas nécessaire d'ajouter un point-virgule à la fin de chacune de nos instructions si ces dernières sont inscrites sur des lignes différentes. Par contre, si vous inscrivez plusieurs instructions sur la même ligne, vous devrez séparer vos instructions par un point-virgule. Ainsi, le prochain bloc de code fonctionnera correctement.

```
<script>
    /* L'utilisation de point-virgule
    n'est pas nécessaire si les
    instructions sont sur des lignes différentes.*/
    var message1 = "Ce texte est affiché avec JavaScript depuis l'élément head."
    var message2 = "Bienvenue dans le cours portant sur JavaScript."
    alert(message1)
    alert(message2)
</script>
```

Par contre, le prochain bloc de code ne sera pas exécuté puisque les différentes instructions se retrouvent sur la même ligne sans être séparée par des points-virgules.

```
<script>
    /* Ce script ne fonctionnera pas puisque
    les instructions ne sont pas séparées
    par des points-virgules.*/
    var prenom = "Marc" var age = 25 alert(prenom) alert(age)
</script>
```

***Remarquez*** que dans ce cas, l'éditeur de code Visual Studio Code signale qu'il y a des erreurs à l'aide d'une barre ondulée rouge.

Nous vous recommandons donc de prendre la bonne habitude de terminer vos instructions à l'aide d'un point-virgule afin d'éviter des erreurs d'exécution de vos scripts qui pourraient survenir.

### *Récapitulation*

- Les instructions JavaScript peuvent être écrites à trois endroits différents dans la composition d'une page Web. Nous pouvons les écrire à :
  - l'intérieur de l'élément head d'une page Web.
  - l'intérieur de l'élément body d'une page Web.
  - l'intérieur d'un fichier séparé portant l'extension .js.
- Les instructions JavaScript contenues dans une page Web doivent être incluses à l'intérieur d'un élément script.
- Il est possible d'inclure plus d'un élément script dans une page Web.
- Lorsqu'une page Web inclut plus d'un élément script, l'ordre d'exécution des scripts se fait selon l'ordre d'apparition dans la page en débutant par le ou les scripts inclus dans l'élément head de la page.
- Lors de l'utilisation d'un fichier de scripts séparé, vous devez en informer la page en utilisant l'attribut src de l'élément script de votre page.
- Pour de gros projets de site Web, il est recommandé d'utiliser un fichier de scripts séparé.
- Il est important de bien commenter son code JavaScript.
- Il n'est pas nécessaire de terminer vos différentes instructions avec un point-virgule tant qu'elles ne se trouvent pas sur la même ligne. Une bonne habitude à prendre est de terminer vos instructions à l'aide d'un point-virgule pour éviter les erreurs d'exécution qui pourraient survenir.

# *Notes*