

# Javascript et HTML

## 1. Présentation de Javascript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs.

JavaScript a été créé en 1995 par l'américain Brendan Eich. Il a été standardisé sous le nom d'ECMAScript en juin 1997 par Ecma International dans le standard ECMA-262.

Brendan Eich Mozilla Foundation official photo  
Brendan Eich, né en 1961 à Pittsburgh en Pennsylvanie

## 2. Fonctionnement de Javascript

Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies 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, on peut donc se passer des serveurs pour le faire fonctionner.

### Les balises

Les navigateurs décryptant le HTML, vont identifier le code encapsulé entre les deux balises, l'une ouvrante, l'autre fermante : `<script> </script>`

### Place des balises

Cette balise peut se placer n'importe où sur la page HTML mais il est d'usage de placer les scripts en fin de page, juste avant la balise `</body>`.

On peut aussi faire figurer le code dans un fichier externe comme nous le verrons.

## 2a. Un premier exemple, le fameux "Hello, World!"

Recopiez le code suivant sur une page html nommée index.html et ouvrez-le avec votre navigateur.

```
<!DOCTYPE html>
<html>
  <head>
    <title>TD Javascript</title>
  </head>
```

```
<body>
  <h1>Mes premiers exemples</h1>
  <p></p>
  <script>
    alert('Hello world! Dans une nouvelle fenêtre');
    document.write('Hello world! sur la page');
  </script>
</body>
</html>
```

On obtient le résultat suivant :

[https://math93.com/images/pdf/NSI/premiere/html-premiere/exemple\\_javascript-1.html](https://math93.com/images/pdf/NSI/premiere/html-premiere/exemple_javascript-1.html)

## Instructions et point-virgule

Les instructions Javascript doivent être suivies d'un point-virgule ;

## Les fonctions et méthodes : exemple, la boîte de dialogue `alert()`, et `document.write("texte")`

- `alert()` est une instruction simple, appelée fonction, qui permet d'afficher une boîte de dialogue contenant un message. Ce message est placé entre apostrophes, elles-mêmes placées entre les parenthèses de la fonction `alert()`.
- `document.write()` est une méthode (et une fonction) , qui permet d'afficher un texte ou une variable sur le document. Ce message est placé entre apostrophes, elles-mêmes placées entre les parenthèses de la fonction `document.write()`.
- Une fonction se compose de son nom suivi d'un couple de parenthèses (une ouvrante et une fermante).
- Comme en Python, entre les parenthèses se trouvent les arguments, que l'on appelle aussi paramètres de la fonction. Ceux-ci contiennent des valeurs qui sont transmises à la fonction. Dans le cas de la fonction `alert()` ici, ce sont les mots « Hello world! » qui sont passés en paramètre.  
`alert('Hello world!');` // c'est une première fonction en Javascript  
`document.write('Hello world!');` // c'est une autre fonction (méthode) en Javascript

## Les commentaires

- Comme dans tout langage, il est important de commenter ses scripts.

Les commentaires s'écrivent : derrière le symbole // et sur une seule ligne ;ou alors précédé de /\* et se terminant par \*/ s'ils sont sur plusieurs lignes.

## Les Erreurs

En cas d'erreurs sur le Javascript on peut regarder s'il y a un message d'erreurs dans la console qui est accessible via **Ctrl+shift+I**

On peut écrire dans la console avec la fonction `console.log()`

## 2b. Le JavaScript externe

Il est possible, d'écrire le code JavaScript dans un fichier externe, portant l'extension `.js`. Ce fichier est ensuite appelé depuis la page Web au moyen de l'élément `<script>` et de son attribut `src` qui contient l'URL du fichier `.js`.

### Exercice 1A

- Créer dans le même répertoire, un fichier **exemple1.js** et un fichier **index.html**.

- Dans le fichier **exemple1.js** on écrit le script : `alert('Hello world!');`

- dans le fichier `index.html`., recopiez le code suivant et testez-le.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mes programmes en Javascript</title>
  </head>
  <body>
    <script src="exemple1.js"></script>
  </body>
</html>
```

On obtient le résultat suivant :

[https://math93.com/images/pdf/NSI/premiere/html-premiere/exemple\\_javascript-2.html](https://math93.com/images/pdf/NSI/premiere/html-premiere/exemple_javascript-2.html)

### Exercice 1B

Pour tous les exercices suivants, vous allez compléter la page `index.html`, en créant des liens vers les différents exercices JavaScript de ce TD.

Vous nommerez chaque fichier exercice sous la forme : "VotreNom-Javascript-Exercice2.html".

Cela va par exemple donner :

```
<!DOCTYPE html>
<html>
  <head>
    <title>TD javascript</title>
  </head>
  <body>
    <h1>Mes premiers exemples</h1>
    <script src="exemple1.js"></script>
    <p></p>
    <ol>
      <li>Exercice 2 : <a href="VotreNom-Javascript-Exercice2.html"> lien sur l'exercice 2</a> </li>
      <li>Exercice 3 : <a href="VotreNom-Javascript-Exercice3.html"> lien sur l'exercice 3</a> </li>
      <li>Exercice 4 : <a href="VotreNom-Javascript-Exercice4.html"> lien sur l'exercice 4</a> </li>
    </ol>
  </body>
</html>
```

## 3. Programmer en Javascript

### 3a. Les variables

Résumé du cours

- Une variable est un moyen pour stocker une valeur.
- Mot clé `var`  
On utilise les mots clés : `var` , `const` et `let` pour déclarer une variable, et on utilise `=` pour affecter une valeur à la variable.

Pour appréhender les différences entre `var`, `const` et `let`. <https://code-garage.fr/blog/quelles-sont-les-differences-entre-var-let-et-const-en-javascript/>

- Variables typées dynamiquement  
Les variables sont typées dynamiquement, ce qui veut dire que l'on n'a pas besoin de spécifier le type de contenu que la variable va contenir.

- Opérations  
Grâce à différents opérateurs, on peut faire des opérations entre les variables.

Voici la liste des principaux opérateurs avec les compatibilités des navigateurs :

[https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets\\_globaux/Math](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/Math)

- Concaténation

L'opérateur **+** permet de concaténer des chaînes de caractères, c'est-à-dire de les mettre bout à bout.

- **Interaction avec `prompt()`**

La fonction `prompt()` permet d'interagir avec l'utilisateur.

Tout ce qui est écrit dans le champ de texte de `prompt()` est récupéré sous forme d'une chaîne de caractères, que ce soit un chiffre ou non.

Si vous utilisez l'opérateur **+**, vous ne ferez pas une addition mais une concaténation !

Pour convertir la chaîne de caractères en nombre entier on utilise la fonction `parseInt()`, et en flottant, la fonction `parseFloat()`

Un exemple `<script>`

```
var a, b, produit,somme;
a = prompt('Donnez un nombre');
b= prompt('Donnez un autre nombre :');
a=parseFloat(a) ; // On convertit la chaîne de caractère a
en flottant
b=parseFloat(b); // On convertit la chaîne de caractère b
en flottant
produit = a*b;
somme=a+b;
alert('Le produit est '+ produit+', et la somme est '+
somme);
</script>
```

On obtient le résultat suivant : Exemple 3 .

<https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-Exemple3.html>

## Exercice 2

Ecrire un script Javascript qui pour un triangle ABC rectangle en A, demande la longueur AB, la longueur AC et affiche une valeur approchée de l'hypoténuse BC. Vous nommerez votre fichier : "VotreNom-Javascript-Exercice2.html" et devez faire figurer le lien vers ce fichier sur votre page web : exercices JavaScript (avec votre nom).

Aide : `<script>a = Math.sqrt(2) // a = la racine carrée de 2 (enfin une valeur approchée)</script>`

## 3b. Les conditions

Résumé du cours

- Une condition retourne une valeur booléenne : true ou false.
- En JavaScript on utilise les accolades pour délimiter les blocs d'instructions :

```
if ( /* condition */ ) {
                                // Du code...
} else {
                                // Du code...
}
```

- Les opérateurs logiques en javascript

De nombreux opérateurs existent afin de tester des conditions et ils peuvent être combinés entre eux.

- test d'égalité `==` , supérieur à `>` , supérieur ou égal à `>=` , et différent de `!=`
- contenu et type égal à `===` , et contenu et type différent de `!==`
- le ET logique se traduit par supérieur à `&&`
- le OU logique par `||`
- la négation par le point d'exclamation `!` .
- La condition `if else` permet de combiner les conditions.
- Quand il s'agit de tester une égalité entre une multitude de valeurs, la condition `switch` est préférable.

## Exercice 3

Ecrire un script Javascript qui pour un triangle ABC, demande la longueur AB, la longueur AC et la longueur BC et affiche si ce triangle est rectangle (et en quel sommet) ou si il ne l'est pas.

Vous nommerez votre fichier : "VotreNom-Javascript-Exercice3.html" et devez faire figurer le lien vers ce fichier sur votre page web : exercices JavaScript (avec votre nom).

Vous devrez tester votre script avec :

- 1) ABC tel que AB = 3, AC = 4 et BC = 5 qui devrait afficher : "ABC est rectangle en A" ;

- 2) ABC tel que AB = 50, AC = 30 et BC = 40 qui devrait afficher : "ABC est rectangle en C" ;
- 3) ABC tel que AB = 6, AC = 10 et BC = 8 qui devrait afficher : "ABC est rectangle en B" ;
- 4) ABC tel que AB = 3, AC = 4 et BC = 6 qui devrait afficher : "ABC n'est pas rectangle".

### 3c. Les structures itératives (boucles)

#### Résumé du cours

- L'incrémement est importante au sein des boucles.
- Incrémenter ou décrémenter signifie ajouter ou soustraire une unité à une variable.

```
var x = 0;
```

```
x = x+1 ; // on ajoute 1 à x
alert(x); // Affiche : « 1 »
```

```
x++; // on ajoute 1 à x
alert(x); // Affiche : « 2 »
```

```
x--; // on enlève 1 à x
alert(number); // Affiche : « 1 »
```

- La boucle **while** permet de répéter une liste d'instructions tant que la condition est vérifiée.

```
<script>
    var N = 0,somme=0;
    while (N < 5) {
        N=N+1; // ou N++
        somme=somme+N }
    alert('somme= ' + somme + ' et N=' + N );
    // Affiche « somme = 15 et N = 5 »
</script>
```

Un test avec ce script : <https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-boucles-exemple1.html>

- La boucle **for** est une boucle utilisée pour répéter une liste d'instructions un certain nombre de fois.

```
<script>
    var somme=0;
    for (var N=1 ; N<=5 ; N=N+1 ) // syntaxe (var compteur=valeur départ ;
    compteur<= valeur fin ; incrément)
        { somme=somme+N }
    alert('somme= ' + somme + ' et N=' + N ); // Affiche « somme = 15 et N = 6
    » , la dernière valeur de N est 6 ce qui invalide la condition
</script>
```

Un test avec ce script : <https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-boucles-exemple2.html>

#### Exercice 4

Ecrire un script Javascript qui tente de faire deviner un entier de votre choix et affiche, plus petit, ou plus grand jusqu'à la victoire.

Commencer par fixer votre nombre à deviner puis essayer de le générer aléatoirement dans un deuxième temps.

Vous nommerez votre fichier : "VotreNom-Javascript-Exercice4.html" et devez faire figure le lien vers ce fichier sur votre page web : exercices JavaScript (avec votre nom).

Exemple attendu en vidéo : <https://www.youtube.com/watch?v=pMMvuSCB7WA>

### 3d. Les fonctions

#### Résumé du cours

- Il existe des fonctions natives en JavaScript dont voici une liste des fonctions JavaScript.  
On a par exemple : **alert()** , **prompt()** , **confirm()** , **parseInt()** , **parseFloat()** ,

...

- Il est aussi possible d'en créer, avec le mot-clé **function**.

```
<script>
    function DireBonjour(x) {
        return ('Bonjour ' + x);
    }
    // L'instruction « return » suivie de la concaténation de 'Bonjour'
    // et de la variable chaîne x
    nom=prompt('Quel est votre prénom svp ?'); // On demande le
    prénom à l'utilisateur
```

```
alert(DireBonjour(nom)); // Ici on affiche la valeur retournée par la fonction DireBonjour()
```

```
</script>
```

Par exemple ici avec ce code on obtient :

<https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-Fonctions-Exemple1.html>

- Les variables déclarées avec var au sein d'une fonction ne sont accessibles que dans cette fonction.
- Il faut éviter le plus possible d'avoir recours aux variables globales.
- Une fonction peut recevoir un nombre défini ou indéfini de paramètres. Elle peut aussi retourner une valeur ou ne rien retourner du tout.
- Des fonctions qui ne portent pas de nom sont des **fonctions anonymes** et servent à isoler une partie du code.

### Exercice 5

Ecrire une fonction Javascript qui teste la présence de la lettre "e" minuscule dans une chaîne de caractères passée en paramètre. Cette fonction doit renvoyer vrai ou faux. Utilisez votre fonction après avoir demandé une chaîne de caractères via la fonction `prompt()`.

Vous nommerez votre fichier : "VotreNom-Javascript-Exercice5.html" et devez faire figure le lien vers ce fichier sur votre page web : exercices JavaScript (avec votre nom).

Aide : La fonction `.length` renvoie la longueur d'une chaîne C et la fonction `.charAt(position)` le caractère à la position précisée dans la chaîne C.

Voici un exemple d'utilisation :

<https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-Fonctions-Exemple2.html>

```
<script>
```

```
var C='abcdefg'; // C est une chaîne de caractères
long=C.length; // long est la longueur de la chaîne, ici 7
lettre=C.charAt(0); // lettre est le caractère de la chaîne en position 0 (le 1er caractère)
alert('Chaîne=' + C + ', longueur = ' + long + ', premier caractère = ' + lettre
);
```

```
</script>
```

Avant de poursuivre vers le chapitre 4

Sur les variables. Faire :

le QCM : <https://odyssey.sdlm.be/javascript/04/partie1/chapitre3/qcm.htm>

QCM didactiques :

<https://odyssey.sdlm.be/javascript/05/partie1/chapitre3/questionnaire.htm>

exercice 1 : Déclarer et initialiser une variable ;

<http://odyssey.sdlm.be/javascript/06/partie1/chapitre3/variable.htm>

exercice 2 : Déclarer et initialiser deux variables.

<http://odyssey.sdlm.be/javascript/07/partie1/chapitre3/variables.htm>

Sur les conditions. Faire :

le QCM : <http://odyssey.sdlm.be/javascript/08/partie1/chapitre4/qcm.htm>

La Question ouverte :

<http://odyssey.sdlm.be/javascript/10/partie1/chapitre4/questions.htm>

exercice 3 : Écrire une condition.

<http://odyssey.sdlm.be/javascript/11/partie1/chapitre4/condition.htm>

Sur les boucles. Faire :

le QCM <http://odyssey.sdlm.be/javascript/13/partie1/chapitre5/qcm.htm>

le questionnaire didactique

<http://odyssey.sdlm.be/javascript/14/partie1/chapitre5/questionnaire.htm>

exercice 4 : écrire une boucle while

<http://odyssey.sdlm.be/javascript/15/partie1/chapitre5/while.htm>

exercice 5 : écrire une boucle while qui exécute un `prompt()` .

<http://odyssey.sdlm.be/javascript/16/partie1/chapitre5/prompt.htm>

Sur les fonctions. Faire :

le QCM <http://odyssey.sdlm.be/javascript/17/partie1/chapitre6/qcm.htm>

le questionnaire didactique

<http://odyssey.sdlm.be/javascript/18/partie1/chapitre6/questionnaire.htm>

exercice 6 : Déclarer une fonction

<http://odyssey.sdlm.be/javascript/19/partie1/chapitre6/fonction.htm>

exercice 7 : Écrire une fonction pour comparer deux nombres

<http://odyssey.sdlm.be/javascript/20/partie1/chapitre6/fonction-max.htm>

## 4. Les événements en HTML

JavaScript permet de réagir à un événement lié à la page HTML :

- clic , utilisation du clavier, focus sur une zone ...

Pour réagir à un évènement de ce type, il faut commencer par agir au niveau HTML en ajoutant un attribut à la balise à laquelle on souhaite ajouter une interaction. La syntaxe est toujours la même :

**onEvènement = "Fonction"**

Les événements permettent de déclencher une fonction selon qu'une action s'est produite ou non.

Par exemple, on peut faire apparaître une fenêtre alert() lorsque l'utilisateur survole une zone d'une page Web, clique sur un texte...

Voici une liste de quelques évènements :

EventHandler :	Evènement
<b>Gestionnaire d'évènement (avec exemple)</b>	
onclick	Clic de la souris sur l'élément sur l'élément
ondblclick	Double-clic de la souris sur l'élément sur l'élément
onkeypress	Frapper (appuyer puis relâcher) une touche de clavier sur l'élément
onkeydown	Appuyer (sans relâcher) sur une touche de clavier sur l'élément
onmouseover	Faire entrer le curseur sur l'élément
onsubmit	Envoi d'un formulaire
input	Taper un caractère dans un champ de texte
<b>select</b>	Sélectionner le contenu d'un champ de texte (input,textarea, etc.)

On trouve de nombreux exemples sur cette page :

[https://www.w3schools.com/js/js\\_htmlDOM\\_events.asp](https://www.w3schools.com/js/js_htmlDOM_events.asp)

#### 4.1. Réponse à un click sur un élément en HTML

Il s'agit de l'interaction la plus simple. Un élément avec l'attribut **onclick** déclenche une fonction JavaScript

Dans cet exemple le clique sur le bouton avec le texte défini par **value= "Ouvrir une fenêtre alerte"**, déclenche la fonction **fenetre()** qui a été définie dans les scripts et qui affiche un message.

Les évènements 'onclick' peuvent se programmer en HTML ou en JavaScript dans le script lui-même avec une syntaxe un peu différente :

[https://www.w3schools.com/jsref/event\\_onclick.asp](https://www.w3schools.com/jsref/event_onclick.asp)

Déclencher une fonction JavaScript avec un click - Ouvrir une fenêtre **alert()** au click

```
<!DOCTYPE html>
<html lang="fr">
  <meta charset="utf-8">
  <head>
    <title>Les évènements en JavaScript : Exemple 1</title>
  </head>
  <body>
    <input type="button" value="Ouvrir une fenêtre alerte"
      onclick="fenetre('Bonjour')"/>
    <script>
      function fenetre(x) {alert(x + ' Madame ou Monsieur'); }
    </script>
  </body>
</html>
```

Voici un premier exemple : <https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-evenement-exemple1.html>

#### Créer un affichage dans à un endroit précis de la page au click

- Dans cet autre exemple, deux boutons, avec l'un on affiche la date et avec l'autre on modifie un paragraphe.

```
<body>
  <h2> Des boutons </h2>
  <h3>1. Un premier bouton qui affiche la date</h3>

  <!--un premier bouton-->
  <button type="button"
    onclick="document.getElementById('demo').innerHTML=Date()" >
    Cliquez sur le bouton 1 </button>
  <!-- -->

  <!--La balise id="demo" permet d'identifier un élément de la page-->
  <p id="demo">C'est ici</p>
  <!-- -->
```

```
<h3>2. Un autre bouton qui modifie du contenu HTML</h3>
<p> On va modifier le contenu de la balise HTML de paragraphe nommée
"ICI" (grâce à l'identifiant id="ICI"). Il change sous vos yeux. </p>
```

```
<!--La balise id="ICI" permet d'identifier un autre élément de la page-->
<p id="ICI" > Ici la balise HTML nommée ICI : On va changer ce texte. </p>
<!-- -->
```

```
<!--un autre bouton-->
<button type="button"
onclick="document.getElementById('ICI').innerHTML= 'On change' " >
Cliquez sur le bouton 2</button>
<!-- -->
```

```
</body>
```

Lien vers : évènements exemple 1bis :

<https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-evenement-exemple1bis.html>

Dans cet autre exemple, la couleur du fond d'écran change quand on clique sur le bouton.

```
<body>
<input type="button" value="Changer de couleur"
onclick="ChangementCouleur()"/>
<script>
    function ChangementCouleur() {
        const couleur=document.body.style.backgroundColor;
        // on charge la couleur du fond, const pour déclarer
        variables (avec let) remplace var
        alert("Vous m'avez cliqué ! On change de couleur = " +
        couleur);
        if (couleur=='ivory')
        {document.body.style.backgroundColor = 'orange';}
        else {document.body.style.backgroundColor = 'ivory';}
        }
    </script>
</body>
```

Lien vers : évènements exemple 2,

<https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-evenement-exemple2.html>

#### 4.2. Choix dans une liste déroulante

On va décrire l'exemple 3 qui suit.

- On détecte un changement sur l'élément **select** via la méthode indiquée par l'attribut **onchange**, ici la méthode **selection()**.
- Dans la fonction **selection()**, on cible l'élément **select** par son identifiant (id="choix").
- Puis on accède au choix effectué à l'aide de **selecteur.selecteurIndex**.
- Enfin on affiche ici la valeur et le texte de l'option choisie.

```
<!DOCTYPE html>
<html lang="fr">
    <meta charset="utf-8">
    <head>
        <title>Les évènements en javaScript : Exemple 3</title>
    </head>
    <body>
        <select id="choix" name="lang" onchange="selection()">
            <option value="fr"> Français </option>
            <option value="zh"> Chinois </option>
            <option value="it"> Italien </option>
        </select>
        <script>
            function selection() {
                const selecteur = document.getElementById("choix");
                const monChoix=selecteur[selecteur.selectedIndex];
                alert('Vous avez choisi la valeur = ' + monChoix.value + '
                , soit le choix = ' + monChoix.text);}
            </script>
        </body>
    </html>
```

#### 4.3. Un petit formulaire

On présente ici comment récupérer une valeur entrée dans une case à l'aide d'un formulaire après validation d'un bouton.

```

<body>
<!-->
<h2> Les évènement en HTML/JavaScript : Un petit formulaire</h2>
<!-->

<!-->
Donnez un entier entre 1 et 4, je vais vous le traduire en anglais :
<FORM>
  <input type="text" name="Valeur_Reponse" id="Id_Reponse"
  value="" />
  <input type="button" value="Valider" onclick="MaFonction();" />
</FORM>

<!--On affichera le résultat ici-->
<p id='resultat'>S'écrit en anglais :</p>
<!-->

<script>
function MaFonction(){
  var reponse = document.getElementById("Id_Reponse").value; // Pour
  récupérer la valeur entrée

  if (reponse=='1')
    {document.getElementById('resultat').innerHTML= 'one';}
  else if (reponse=='2')
    {document.getElementById('resultat').innerHTML= 'two';}
  else if (reponse=='3')
    {document.getElementById('resultat').innerHTML= 'three';}
  else if (reponse=='4')
    {document.getElementById('resultat').innerHTML= 'four';}
  else
    {document.getElementById('resultat').innerHTML= 'Je ne
  comprend pas';} }
</script>
<!-->
</body>

```

Lien vers l'exemple 4 : <https://math93.com/images/pdf/NSI/premiere/html-premiere/Duffaud-Javascript-evenement-exemple4.html>

## Exercice 6

Concevoir une page avec un Quiz avec au moins 3 questions qui portent sur le programme de NSI :

- 1 questions Vrai/Faux avec avec deux boutons ou un QCM ;
- 1 question avec une liste déroulante (QCM) ;
- 1 question ouverte avec un texte ou une valeur à donner.

Contraintes :

- Vous devez aussi afficher le score final.
- Vous devez personnaliser votre page (nom, thème, couleur, page CSS ...)
- Vous nommerez votre fichier : "VotreNom-Javascript-Exercice6.html".

En vidéo : <https://www.youtube.com/watch?v=rbXg9w29Vps>