

MODELER VOS PAGES WEB

https://www.w3schools.com/js/js_htmlDOM.asp

DÉFINITION DE DOM

DOM (Modèle Objet de Document ou *Document Object Model*) est le modèle d'accès aux différentes composantes des documents HTML ou encore XML.

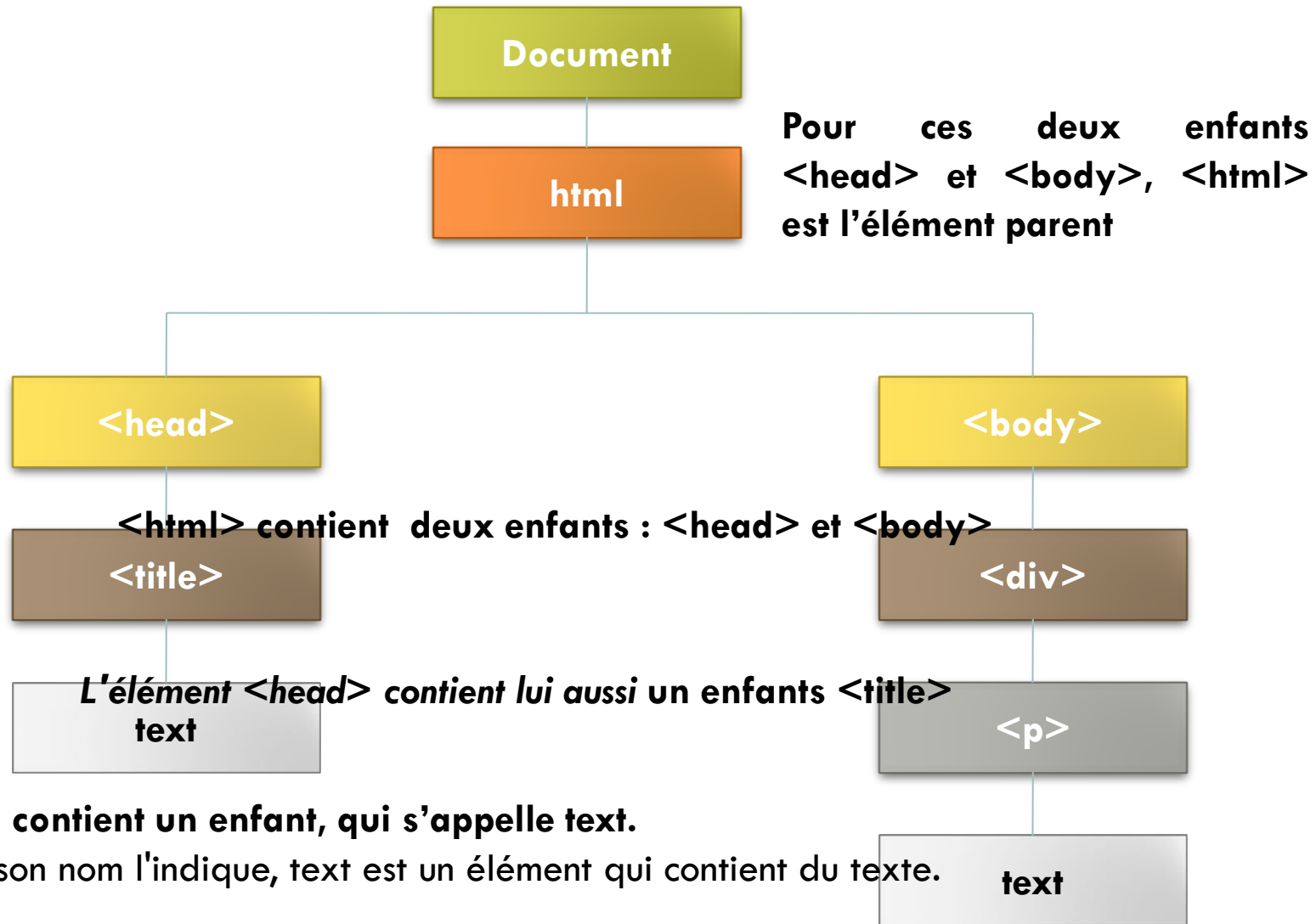
DOM permet de fournir un schéma d'accès pour atteindre les constituants (formulaire, champ...) d'une page web balisée.

Le modèle DOM est un arbre constitué d'objet **Node** (nœuds). Il s'agit d'une structure arborescente avec un nœud de type racine et des nœuds enfants.

Chaque élément dans le HTML, il a un objet qu'il lui correspond dans le DOM .

Le DOM est une interface de programmation, qu'on appelle aussi une **API** (pour **Application Programming Interface**).

DOM permet d'accéder aux différents éléments de documents HTML via le JavaScript. C'est grâce au DOM que nous allons pouvoir modifier des éléments HTML (afficher ou masquer, ajouter, déplacer ou même supprimer).



OBJET WINDOW

- L'objet **window** est un **objet global** qui représente **la fenêtre du navigateur**.
- L'objet **document** est un **sous-objet** de **window** et représente **la page Web**.

L'objet **document** stocke l'arbre DOM de la page.

C'est grâce à lui que l'on va **pouvoir accéder aux éléments HTML de la page Web**.

Note importante:



alert() n'est pas vraiment une fonction. Il s'agit en réalité d'une **méthode appartenant à l'objet window**. Mais l'objet window est dit implicite.

EXEMPLE

Formulaire

Nom :

e-mail :

Message :

Page >> : Modele_DOM.html x

▼ top

file://

C:/MyDisk/IUT Aix-

Modele_DOM.htm

style.css

```
12 <!-- Balise meta de gestion de l'accentuation UTF-8 -->
13 <meta content="text/html; charset=UTF-8" http-equiv="Content-Type">
14
15 <link href="style.css" rel="stylesheet">
16 </head>
17 <body>
18 <h2> <center> Formulaire </center> </h2>
19
20 <form action="/ma-page-de-traitement" method="post">
21   <div>
22     <label for="name">Nom :</label>
23     <input type="text" id="name" name="user_name">
24   </div>
25
26   <div>
27     <label for="mail">e-mail&nbsp;</label>
28     <input type="email" id="mail" name="user_mail">
29   </div>
30
31   <div>
32     <label for="msg">Message :</label>
33     <textarea id="msg" name="user_message"></textarea>
34   </div>
35 </form>
36 </body>
37 </html>
```

Ligne 38, colonne 25 Couverture : N/A

Console Couverture Problèmes Affichage

▶ alert: f alert()

▶ atob: f atob()

▶ blur: f blur()

▶ btoa: f btoa()

▶ caches: CacheStorage {}

▶ cancelAnimationFrame: f cancelAnimationFrame()

EXEMPLE

Formulaire

Nom :

e-mail :

Message :

Éléments Console Sources Réseau Performances Mémoire Appli Sécurité Lighthouse >>

Page >> Modele_DOM.html x

▼ top
▼ file://
C:/MyDisk/IUT Aix-en-Provence/Anne%20Universitaire%202022-2023/R4.A.10%20Complement%20Web%20(JS)/Ar
Modele_DOM.html
style.css

```
12 <!-- Balise meta de gestion de l'accentuation UTF-8 -->
13 <meta content="text/html; charset=UTF-8" http-equiv="Content-Type">
14
15 <link href="style.css" rel="stylesheet">
16 </head>
17 <body>
18 <h2> <center> Formulaire </center> </h2>
19
20 <form action="/ma-page-de-traitement" method="post">
21   <div>
22     <label for="name">Nom :</label>
23     <input type="text" id="name" name="user_name">
24   </div>
25
26   <div>
27     <label for="mail">e-mail<\/label>
28     <input type="email" id="mail" name="user_mail">
29   </div>
30
31   <div>
32     <label for="msg">Message :</label>
33     <textarea id="msg" name="user_message"></textarea>
34   </div>
35 </form>
36 </body>
37 </html>
```

Ligne 38, colonne 25 Couverture : N/A

Console Couverture Problèmes Affichage

top Filtre Niveaux par défaut Aucun problème

```
crossOriginIsolated: false
crypto: Crypto {subtle: SubtleCrypto}
customElements: CustomElementRegistry {}
devicePixelRatio: 1
document: document
  location: Location {ancestorOrigins: DOMStringList, href: 'file:///C:/MyDisk/IUT%20Aix-en-Provence/Anne%20Uni...2%20-%20%20M
    URL: "file:///C:/MyDisk/IUT%20Aix-en-Provence/Anne%20Universitaire%202022-2023/R4.A.10%20Complement%20Web%20(JS)/Ar
  activeElement: body
```

INTERFACE NODE

Pour parcourir les différents niveaux de l'arborescence DOM, l'interface de Node est utilisée. Elle possède des propriétés et des méthodes.

- Les propriétés

nodeName,.nodeType , nodeValue , parentNode , attributes, childNodes , firstChild , lastChild , nextSibling ...

- Les méthodes

appendChild () , cloneNode () , deleteData () , getAttribute () , removeChild () , replaceChild () , setAttribute () , setAttributeNode () ...

EXEMPLE

Formulaire

Nom :

e-mail :

Message :

```
<body>
  <h2> <center> Formulaire </center> </h2>
  <form action="/ma-page-de-traitement" method="post">
    <div>
      <label for="name">Nom :</label>
      <input type="text" id="name" name="user_name">
    </div>
    <div>
      <label for="mail">e-mail&nbsp;  </label>
      <input type="email" id="mail" name="user_mail">
    </div>
    <div>
      <label for="msg">Message :</label>
      <textarea id="msg" name="user_message"></textarea>
    </div>
  </form>
  <script>
    console.log(document.body.children);
  </script>
</body>
</html>
```

Ligne 38, colonne 13

Couverture : N/A

Console

HTMLCollection(3) [h2, form, script]

- 0: h2
- 1: form
- 2: script

length: 3

[[Prototype]]: HTMLCollection

EXEMPLE

Formulaire

Nom :

e-mail :

Message :

The screenshot displays a web browser window with a form titled "Formulaire" and its corresponding DOM structure in the developer tools. The form contains three input fields: "Nom", "e-mail", and "Message". The DOM structure is shown in the "Sources" tab, highlighting the HTML elements and the JavaScript code that logs the form data.

```
17 <body>
18 <h2> <center> Formulaire </center> </h2>
19
20 <form action="/ma-page-de-traitement" method="post">
21   <div>
22     <label for="name">Nom :</label>
23     <input type="text" id="name" name="user_name">
24   </div>
25
26   <div>
27     <label for="mail">e-mail<\/label>
28     <input type="email" id="mail" name="user_mail">
29   </div>
30
31   <div>
32     <label for="msg">Message :</label>
33     <textarea id="msg" name="user_message"><\/textarea>
34   </div>
35 <\/form>
36
37 <script>
38   console.log(document.body.children[1].children);
39 <\/script>
40 <\/body>
```

The DOM structure is shown in the "Sources" tab, highlighting the HTML elements and the JavaScript code that logs the form data. The DOM structure is shown in the "Sources" tab, highlighting the HTML elements and the JavaScript code that logs the form data.

Console: Ligne 38, colonne 13

Couverture : N/A

Console: Couverture Problèmes Affichage

Filtrer Niveaux par défaut Aucun problème

HTMLCollection(3) [div, div, div] 3

- ▶ 0: div
- ▶ 1: div
- ▶ 2: div

length: 3

[[Prototype]]: HTMLCollection

EXEMPLE

```
15 <link href="style.css" rel="stylesheet">
16 </head>
17 <body>
18 <h2> <center> Formulaire </center> </h2>
19
20 <form action="/ma-page-de-traitement" method="post">
21 <div>
22 <label for="name">Nom :</label>
23 <input type="text" id="name" name="user_name">
24 </div>
25
26 <div>
27 <label for="mail">e-mail<nbsp>:</label>
28 <input type="email" id="mail" name="user_mail">
29 </div>
30
31 <div>
32 <label for="msg">Message :</label>
33 <textarea id="msg" name="user_message"></textarea>
34 </div>
35 </form>
36
37 <script>
38 console.log(document.body.children[0]);
39 </script>
40 </body>
```

Ligne 38, colonne 13

Console

```
<h2>
  <center> Formulaire </center>
</h2>
```

```
15 <link href="style.css" rel="stylesheet">
16 </head>
17 <body>
18 <h2> <center> Formulaire </center> </h2>
19
20 <form action="/ma-page-de-traitement" method="post">
21 <div>
22 <label for="name">Nom :</label>
23 <input type="text" id="name" name="user_name">
24 </div>
25
26 <div>
27 <label for="mail">e-mail<nbsp>:</label>
28 <input type="email" id="mail" name="user_mail">
29 </div>
30
31 <div>
32 <label for="msg">Message :</label>
33 <textarea id="msg" name="user_message"></textarea>
34 </div>
35 </form>
36
37 <script>
38 console.log(document.body.firstChild);
39 </script>
40 </body>
```

Ligne 38, colonne 13

Console

```
<h2>
  <center> Formulaire </center>
</h2>
```

EXEMPLE

Formulaire

Nom :

e-mail :

Message :

Éléments Console Sources Réseau Performances Mémoire Appli Sécurité Lighthouse

Page >> Modele_DOM.html x

topfile://C:/MyDisk/IUT Aix-Marseille 3/Travaux Pratiques/Modele_DOM.htmlstyle.css

```
17 <body>
18 <h2> <center> Formulaire </center> </h2>
19
20 <form action="/ma-page-de-traitement" method="post">
21   <div>
22     <label for="name">Nom :</label>
23     <input type="text" id="name" name="user_name">
24   </div>
25
26   <div>
27     <label for="mail">e-mail<\/label>
28     <input type="email" id="mail" name="user_mail">
29   </div>
30
31   <div>
32     <label for="msg">Message :</label>
33     <textarea id="msg" name="user_message"></textarea>
34   </div>
35 <\/form>
36
37 <script>
38   console.log(document.body.lastElementChild);
39 <\/script>
40
41 <\/body>
42 <\/html>
```

Ligne 38, colonne 13 Couverture : N/A

Console Couverture Problèmes Affichage

top Filtre

<script> console.log(document.body.lastElementChild); <\/script> Modele_DOM.html:38

EXEMPLE

Formulaire

Nom :

e-mail :

Message :

Éléments Console Sources Réseau Performances Mémoire Appli Sécurité Lighthouse

Page >> Modele_DOM.html x

top
file://
C:/MyDisk/IUT Aix-
Modele_DOM.ht
style.css

```
17 <body>
18 <h2> <center> Formulaire </center> </h2>
19
20 <form action="/ma-page-de-traitement" method="post">
21   <div>
22     <label for="name">Nom :</label>
23     <input type="text" id="name" name="user_name">
24   </div>
25
26   <div>
27     <label for="mail">e-mail<nbsp>:</label>
28     <input type="email" id="mail" name="user_mail">
29   </div>
30
31   <div>
32     <label for="msg">Message :</label>
33     <textarea id="msg" name="user_message"></textarea>
34   </div>
35 </form>
36
37 <script>
38   console.log(document.body.children[1].nextElementSibling);
39 </script>
40
41 </body>
42 </html>
```

Ligne 18, colonne 20 Couverture : N/A

Console Couverture Problèmes Affichage

top

Filtrer

Niveaux par défaut

Aucun problème

```
<script> console.log(document.body.children[1].nextElementSibling); </script>
```

Modele_DOM.html:38

EXEMPLE

Formulaire

Nom :

e-mail :

Message :

Éléments Console Sources Réseau Performances Mémoire Appli Sécurité Lighthouse

Page >> Modele_DOM.html x

top file:// C:/MyDisk/IUT Aix-M... Modele_DOM.htm style.css

```
15 <link href="style.css" rel="stylesheet">
16 </head>
17 <body>
18 <h2> <center> Formulaire </center> </h2>
19
20 <form action="/ma-page-de-traitement" method="post">
21   <div>
22     <label for="name">Nom :</label>
23     <input type="text" id="name" name="user_name">
24   </div>
25
26   <div>
27     <label for="mail">e-mail&nbsp;  </label>
28     <input type="email" id="mail" name="user_mail">
29   </div>
30
31   <div>
32     <label for="msg">Message :</label>
33     <textarea id="msg" name="user_message"></textarea>
34   </div>
35 </form>
36
37 <script>
38   console.log(document.body.children[1].parentElement);
39 </script>
40 </body>
41 </html>
```

Ligne 38, colonne 31 Couverture : N/A

Console Couverture Problèmes Affichage

top Filtre

Niveaux par défaut Aucun problème

> <body>...</body>

Modele_DOM.html:38

ACCÈS AUX ÉLÉMENTS DE LA PAGE

L'objet document est la racine de tous les nœuds d'un script Web. Cet objet particulier a ses propres méthodes s'appliquant au document dans sa globalité :

- `getElementById()` qui permet d'accéder à un élément en connaissant son ID qui est simplement l'attribut id de l'élément.
- `getElementsByTagName()` qui permet de récupérer, sous la forme d'un tableau, tous les éléments de la famille. Elle retourne une collection d'éléments (utilisable de la même manière qu'un tableau). Pour accéder à chaque élément, il est nécessaire de parcourir le tableau avec une petite boucle.
- `getElementsByName()` est semblable à `getElementsByTagName()` et permet de ne récupérer que les éléments qui possèdent un attribut name que vous spécifiez. L'attribut name n'est utilisé qu'au sein des formulaires

ACCÈS AUX ÉLÉMENTS

- `querySelector()` et `querySelectorAll()` ont pour particularité de prendre pour paramètre un seul argument: une chaîne de caractères . Cette chaîne de caractères doit être un sélecteur CSS comme ceux que vous utilisez dans vos feuilles de style.

`querySelector()`, renvoie le premier élément trouvé correspondant au sélecteur CSS, tandis que `querySelectorAll()` va renvoyer tous les éléments (sous forme de tableau) correspondant au sélecteur CSS fourni.

EXEMPLE

```
<div id="menu">
  <div class="item">
    <span>Élément 1</span>
    <span>Élément 2</span>
  </div>

  <div class="publicite">
    <span>Élément 3</span>
    <span>Élément 4</span>
  </div>
</div>

<div id="contenu">
  <span>Introduction au contenu de la page...</span>
</div>

<script type="text/javascript">
  var query= document.querySelector('#menu .item span'), queryAll = document.querySelectorAll('#menu .item span');

  alert(query.innerHTML); //Affiche Élément 1
  alert(queryAll.length); // Affiche : "2"
  alert(queryAll[0].innerHTML + ' - ' + queryAll[1].innerHTML); //Affiche : "Élément 1 - Élément 2"
</script>
```


EXAMPLE

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <div id="menu">
    <div class="item">
      <span>Élément 1</span>
      <span>Élément 2</span>
    </div>

    <div class="publicite">
      <span>Élément 3</span>
      <span>Élément 4</span>
    </div>
  </div>

  <div id="contenu">
    <span>Introduction au contenu de la page...</span>
  </div>

  <script type="text/javascript">
    var query= document.querySelector('#menu .item span'), queryAll = document.querySelectorAll('#menu .item span');

    alert(query.innerHTML);//Affiche Élément 1
    alert(queryAll.length); // Affiche : "2"
    alert(queryAll[0].innerHTML + ' - ' + queryAll[1].innerHTML); //Affiche : "Élément 1 - Élément 2"
  </script>
</body>
</html>
```

L'HÉRITAGE DES PROPRIÉTÉS ET DES MÉTHODES

- Le Javascript voit les éléments HTML comme étant des objets. Chaque élément HTML possède des propriétés et des méthodes.
- En Javascript, un objet peut appartenir à plusieurs groupes.



Attention: Les éléments HTML ne possèdent pas tous les mêmes propriétés et méthodes.

- Tous les éléments HTML sont d'un même type: le type **Node**.
- L'objet **Node** apporte un certain nombre de propriétés et de méthodes qui pourront être utilisées depuis un de ses sous-objets.
- Les sous-objets héritent des propriétés et méthodes de leurs objets parents. Voilà donc ce que l'on appelle l'**héritage**.

EDITER LES ÉLÉMENTS HTML

LES ATTRIBUTS VIA L'OBJET ÉLÉMENT

Pour interagir avec les attributs, l'objet Element nous fournit deux méthodes, `getAttribute()` et `setAttribute()`.

- **getAttribute()** permet de récupérer un attribut. Elle prend en paramètre le nom de l'attribut.
- **setAttribute()** permet d'éditer un attribut. Elle prend en paramètre la nouvelle valeur à donner à l'attribut.

EXEMPLE

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <a id="myLink" href="https://www.edx.org/course/javascript-introduction"> lien modifié dynamiquement</a>

  <script>
    var link = document.getElementById('myLink');
    var href = link.getAttribute('href'); // On récupère l'attribut « href »

    alert(href);
    link.setAttribute('href', 'http://www.istic.rnu.tn/fr/'); // On édite l'attribut « href »
  </script>
</body>
</html>
```

EDITER LES ÉLÉMENTS HTML

LES ATTRIBUTS ACCESSIBLES

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <a id="myLink" href="https://www.edx.org/course/javascript-introduction"> lien modifié dynamiquement</a>

  <script>
    var link = document.getElementById('myLink');
    var att_href = link.href;

    alert(att_href);
    link.href='http://www.istic.rnu.tn/fr/';
  </script>

</body>
</html>
```

EDITER LES ÉLÉMENTS HTML

LE CONTENU INNERHTML

- La propriété `innerHTML` a été créée par Microsoft pour les besoins d'Internet Explorer et elle est normalisée.
- `innerHTML` permet de récupérer le code HTML enfant d'un élément sous forme de texte.
- Si des balises sont présentes, `innerHTML` les retournera sous forme de texte.

EDITER LES ÉLÉMENTS HTML

RÉCUPÉRER DU HTML

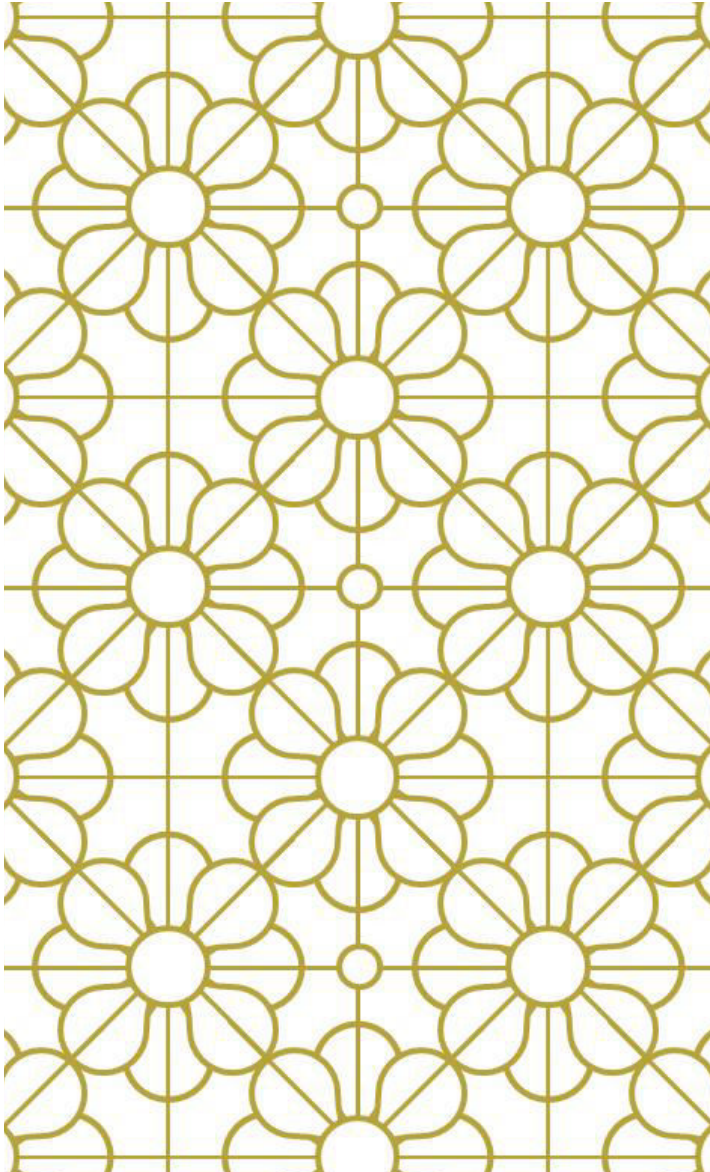
```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <html>
  <body>
    <h2> What can JavaScript do? </h2>
    <p id="id_1"> JavaScript Can chage HTML content </p>
    <script type="text/javascript">
      alert(document.getElementById('id_1').innerHTML);
    </script>
  </body>
</html>
</body>
</html>
```

EDITER LES ÉLÉMENTS HTML

AJOUTER OU ÉDITER DU HTML

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <html>
  <body>
    <h2> What can JavaScript do? </h2>
    <p id="id_1"> JavaScript Can chage HTML content  </p>

    <button type="button" onclick="document.getElementById('id_1').innerHTML='Hello JavaScript and innerHTML'"> CLIQUER </button>
    <button type="button" onclick="window.print()"> Imprimer </button>
  </body>
</html>
</body>
</html>
```

LES ÉVÉNEMENTS

https://www.w3schools.com/js/js_events_examples.asp

LE ÉVÉNEMENTS EN JAVASCRIPT

- 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.

- Il est possible d'ajouter **un événement** à un élément de votre page Web (par exemple, une balise `<div>`) pour faire en sorte de **déclencher** un code Javascript lorsque l'utilisateur fera une action sur l'élément en question.

LISTE DES ÉVÉNEMENTS

Nom de l'événement	Action pour le déclencher
<code>click</code>	Cliquer (appuyer puis relâcher) sur l'élément
<code>dblclick</code>	Double-cliquer sur l'élément
<code>mouseover</code>	Faire entrer le curseur sur l'élément
<code>mouseout</code>	Faire sortir le curseur de l'élément
<code>mousedown</code>	Appuyer (sans relâcher) sur le bouton gauche de la souris sur l'élément
<code>mouseup</code>	Relâcher le bouton gauche de la souris sur l'élément
<code>mousemove</code>	Faire déplacer le curseur sur l'élément
<code>keydown</code>	Appuyer (sans relâcher) sur une touche de clavier sur l'élément
<code>keyup</code>	Relâcher une touche de clavier sur l'élément
<code>keypress</code>	Frapper (appuyer puis relâcher) une touche de clavier sur l'élément
<code>focus</code>	« Cibler » l'élément
<code>blur</code>	Annuler le « ciblage » de l'élément
<code>change</code>	Changer la valeur d'un élément spécifique aux formulaires (<code>input</code> , <code>checkbox</code> , etc.)
<code>select</code>	Sélectionner le contenu d'un champ de texte (<code>input</code> , <code>textarea</code> , etc.)

EVÈNEMENTS SPÉCIFIQUES

Pour l'élément `<form>`, il existe deux événements spécifiques.

Nom de l'événement	Action pour le déclencher
<code>submit</code>	Envoyer le formulaire
<code>reset</code>	Réinitialiser le formulaire

EXEMPLE

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>
    <h3> Code de JavaScript </h3>
    <p id="id_1"> TEST CODE  JavaScript </p>
    <button type="button" onclick="myFunction()"> CLIQUER ICI </button>
    <script src="myFile.js"></script>
  </body>
</html>
```

```
function myFunction()
{
  document.getElementById("id_1").innerHTML="Code JavaScript dans un fichier externe!";
}
```

EXEMPLE

```
<html>
  <head>
    <script>
      function HideElement()
      {
        document.getElementById("id_1").style.display="none";
      }
    </script>
  </head>
  <body>
    <p id="id_1"> Hello World! </p>
    <button type="button" onclick="HideElement()"> Cliquer ici </button>
  </body>
</html>
```

Hello World!

Cliquer ici

Cliquer ici

EXEMPLE

```
<html>
  <head>
    <script>
      function showHideElement()
      {
        document.getElementById("id_1").style.display="block";
      }
    </script>
  </head>
  <body>
    <h2 id="id_1" style="display:none"> Bonjour tout le MONDE! </h2>
    <button type="button" onclick="showHideElement()"> Cliquer ICI </button>
  </body>
</html>
```

Cliquer ICI

Bonjour tout le MONDE!

Cliquer ICI

EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <p id="id_1"> JavaScript can change HTML attributes</p>
    

    <button type="button" onclick="document.getElementById('lamp').src='lampe_on.png'"> Turn ON </button>
    <button type="button" onclick="document.getElementById('lamp').src='lampe_off.png'"> Turn OFF </button>

  </body>
</html>
```

