



Conception de pages Web (HTML/CSS)

Introduction à HTML

Enzo Doyen

enzo.doyen@unistra.fr

2026 - LGA1DMT2

Informations organisationnelles

4 séances en salle **4S04** de **8 h à 11 h** ;

- ♦ du **29 janvier** au **19 février**
- ♦ **2** examens de **30 minutes** : les **12** et **19 février**.
 - Chaque épreuve est de coefficient 1 et compte pour 50 % de la note finale.

Informations organisationnelles

Page Moodle

Page « LGA1DMT2 — Introduction à la réalisation de pages Web avec HTML et CSS (2025-2026) » sur Moodle :

<https://moodle.unistra.fr/course/view.php?id=20825>

Contact

En cas de problèmes, questions :
enzo.doyen@unistra.fr

Objectifs du cours

- ♦ apprendre à utiliser le langage de balisage HTML pour créer des pages Web ;
- ♦ comprendre les bases du langage de style CSS pour mettre en forme ces pages ;
- ♦ connaître les principes de base de la conception de pages Web ;
- ♦ apprendre à utiliser des outils de développement Web.



I. Présentation de HTML

HTML, c'est quoi ?

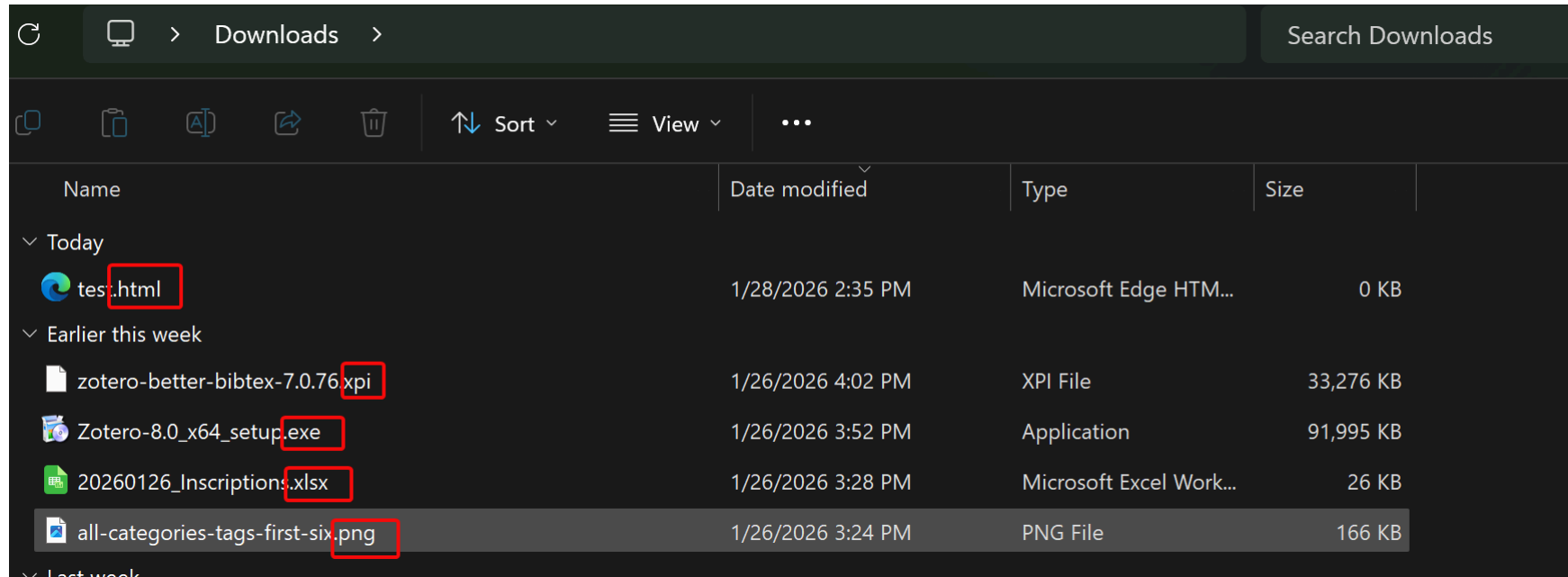
- ♦ **H**yper **T**ext **M**arkup **L**anguage : langage de balisage standard pour structurer le contenu d'une page Web ;
- ♦ Créé en 1991 par Tim Berners-Lee au CERN ;
- ♦ Plusieurs versions disponibles ; la dernière en date et la plus utilisée est HTML5 (2014).

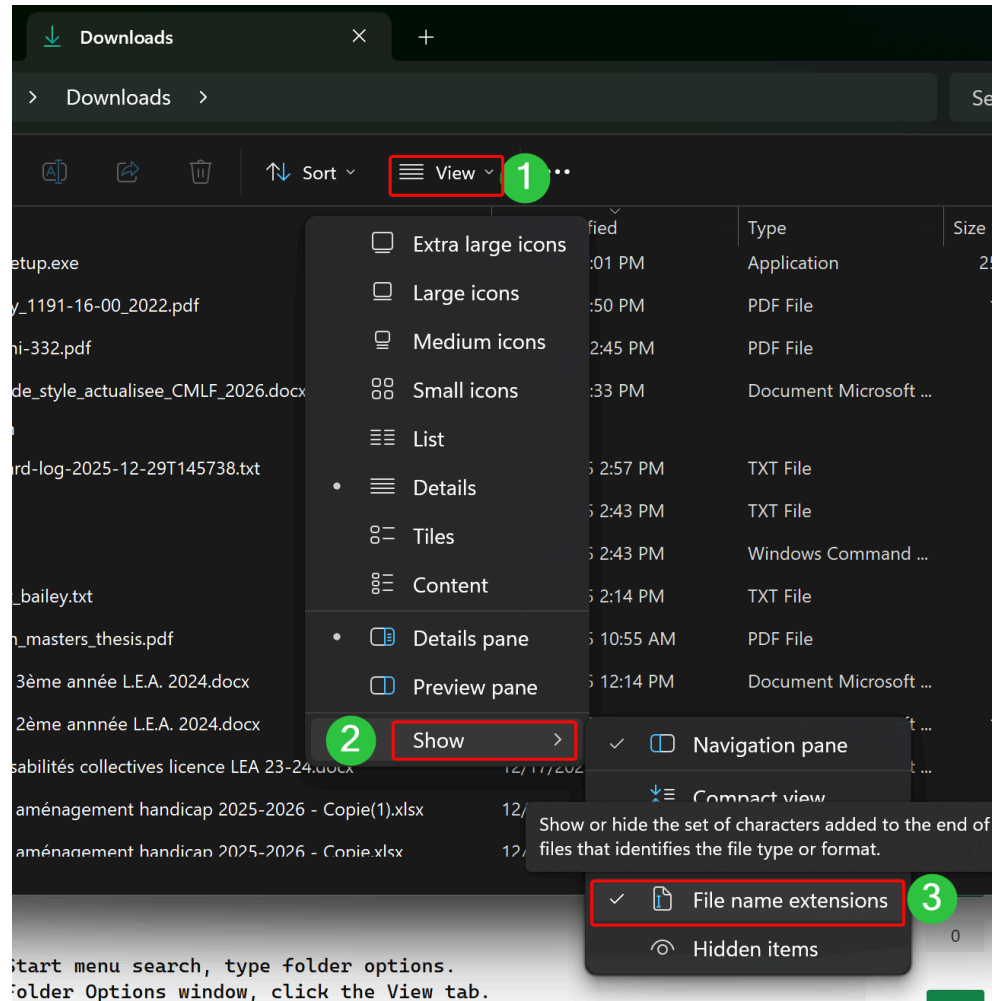
HTML, c'est quoi ?

- ♦ HTML décrit la **structure** des pages Web en utilisant des **balises**.
 - Ces balises sont utilisées pour définir les éléments de la page (titre, paragraphe, image, etc.).

Écrire des fichiers HTML

- Un fichier HTML est un fichier texte avec l'extension `.html` (au lieu de `.txt`);





Écrire des fichiers HTML

- ♦ Un fichier HTML est un fichier texte avec l'extension `.html` (au lieu de `.txt`);
- ♦ Pour modifier ces fichiers, on peut utiliser un éditeur de texte spécialisé (Notepad++, Sublime Text, Visual Studio Code).

HTML : syntaxe de base

Texte

<p>contenu</p>

*Balise
ouvrante*

*Balise
fermante*

HTML : syntaxe de base

Chaque élément HTML est défini par une balise de début et une balise de fin.

On ouvre une balise avec `<nom-de-la-balise>...`

... et on la ferme avec `</nom-de-la-balise>`.

Certaines balises demandent d'avoir un contenu (du texte ou d'autres balises imbriquées) ; pour d'autres, ce n'est pas nécessaire.

HTML : syntaxe de base

Les noms des balises correspondent à leur fonction (en anglais) :

- ♦ p : paragraphe ;
- ♦ h1, h2, h3, h4, h5, h6 : titres (*heading*) de niveau 1 à 6 ;
- ♦ img : image ;
- ♦ etc.

```
1 <h1>Titre de niveau 1.</h1>
```



Il existe des centaines de balises différentes ; liste complète ici : [https://
developer.mozilla.org/fr/docs/Web/HTML/Reference/Elements](https://developer.mozilla.org/fr/docs/Web/HTML/Reference/Elements)

HTML : syntaxe de base

Certaines balises n'ont pas de balise de fin : on les appelle des balises **autofermantes**.

C'est le cas de la balise `img`, ou du saut de ligne `
`.

```
1 <p>Paragraphe 1</p>
```



```
2 <br> <!-- commentaire : saut de ligne -->
```

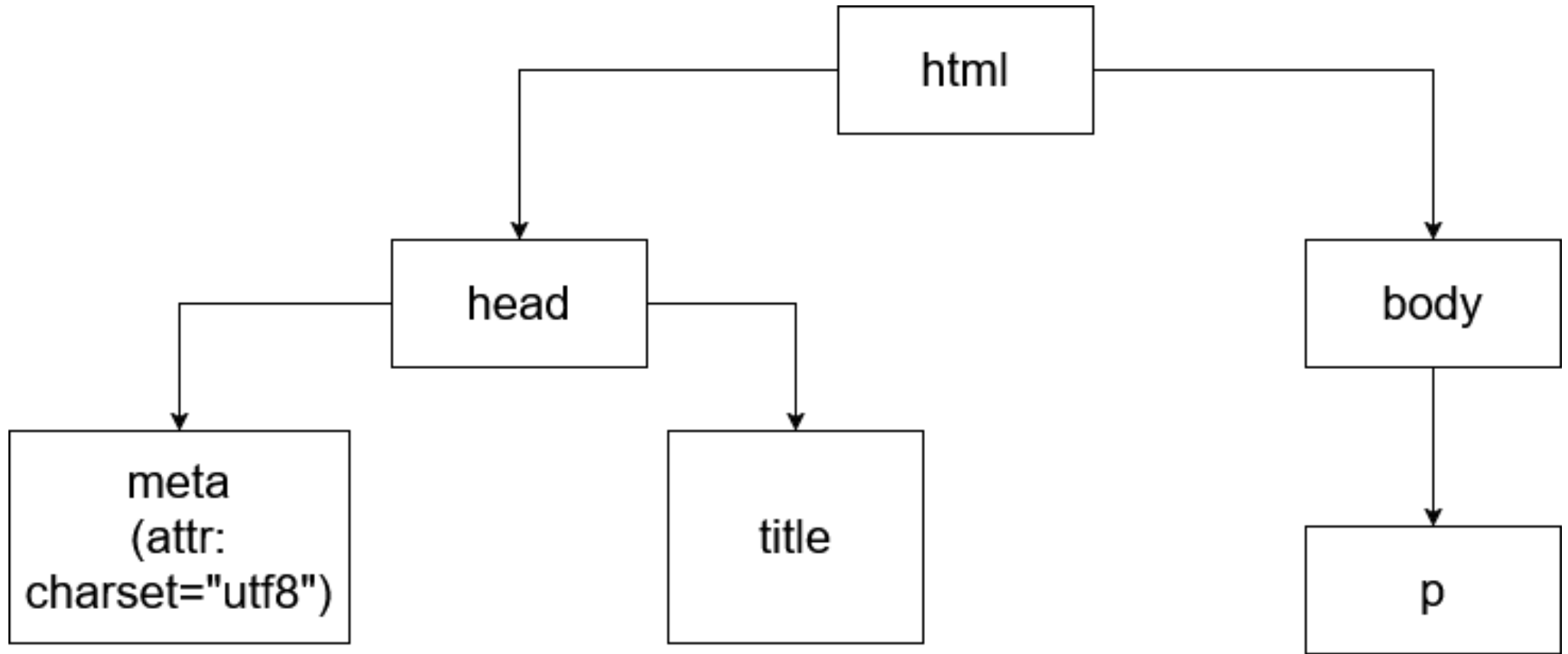
```
3 <p>Paragraphe 2</p>
```

Structure d'une page HTML5

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <title>Titre de la page</title>
6    </head>
7    <body>
8      <p>Contenu</p>
9    </body>
10 </html>
```

head : informations importantes pour l'affichage de la page, mais ne fait pas partie du contenu

body : contenu de la page



<!DOCTYPE html>

Déclaration du type de document

<html>

Élément racine du document HTML

<head>

Contient les métadonnées/informations pour le document

<title>

Titre de la page

<body>

Contient le contenu visible de la page

<p>

Un paragraphe de texte

Structure d'une page HTML5



Structure d'une page HTML5

Cf. modèle sur page Moodle : <https://moodle.unistra.fr/mod/page/view.php?id=744362>

→ Démonstration en direct

HTML : syntaxe de base

La plupart des balises peuvent avoir des **attributs**.

On les place dans la balise de début, après le nom de la balise.

Ils sont associés à une **valeur**, placée entre **guillemets anglais**, **sans espace** entre les guillemets.

HTML : syntaxe de base, balise `img`

Par exemple, la balise `img` a un attribut `src` qui contient l'URL de l'image à afficher.

```
1  <!-- notez la  
   balise autofermante -->
```



HTML : syntaxe de base, balise `img`

En supplément, cette même balise `img` peut aussi avoir :

- ♦ une balise `alt` qui contient un texte alternatif pour l'image ;
- ♦ une balise `width` et `height` pour définir la largeur et la hauteur de l'image.

```
1 
```



Pour avoir plus d'informations sur une balise, on peut consulter sa page

MDN : [https://developer.mozilla.org/fr/docs/Web/HTML/Reference/Elements/
img](https://developer.mozilla.org/fr/docs/Web/HTML/Reference/Elements/img)

Code source HTML

Il est possible de voir la structure HTML de n'importe quelle page en ouvrant la console de développement du navigateur (F12 ou CTRL + SHIFT + K).

The screenshot displays the Chrome DevTools interface. At the top, there's a toolbar with icons for Inspector, Console, Debugger, Network, Style Editor, Performance, Memory, Storage, Accessibility, and Application. Below the toolbar is a search bar labeled "Search HTML". The main area shows the expanded HTML document tree. The root node is `<!DOCTYPE html>`. Its child is `<html itemscope="" itemtype="http://schema.org/SearchResultsPage" lang="en-FR">`, which has two children: `<head>` and `<body>`. The `<body>` node is selected and highlighted in blue. It contains several nested elements: `<style>`, `<div jscontroller="nmfH9e" data-et="30" hidden="true" jsaction="rcuQ6b:npT2md">`, `<noscript>`, another `<style>`, and a `<script nonce="">`. A breadcrumb at the bottom left indicates the current selection: `html > body#gsr.srp`.

Points importants à retenir : imbrication

HTML permet d'**imbriquer** des balises, comme nous l'avons vu dans le modèle HTML auparavant.

<!DOCTYPE html>

Déclaration du type de document

<html>

Élément racine du document HTML

<head>

Contient les métadonnées/informations pour le document

<title>

Titre de la page

<body>

Contient le contenu visible de la page

<p>

Un paragraphe de texte

Points importants à retenir : imbrication

Dans ce cas, il est important de fermer les balises dans l'ordre : la **dernière balise ouverte** doit être **fermée en premier** :

```
1 <html>
2   <head>
3     <title>Mon titre</title>
4   </head>
5 </html>
6 ...
```



Points importants à retenir : casse

Les balises HTML sont **toujours écrites en minuscules**.



Correct :

```
1 <p>Bonjour</p> <!-- Paragraphe -->
```

 HTML



Incorrect :

```
1 <P>Bonjour</P>
```

 HTML

Points importants à retenir : aération du code

Il est vivement conseillé d'**aérer son code** lorsque l'on écrit du HTML : on met généralement un élément par ligne pour les éléments dits « block », tels que `div`, `p`, `section`, `article`, `header`, `footer`, `ul`, `li`, `h1`–`h6`.

Lorsque des éléments sont imbriqués, on **indente** le code (c'est-à-dire que l'on rajoute des espaces au début d'un élément imbriqué) pour faciliter la lecture.

Points importants à retenir : aération du code



Correct :

```
1 <ul> <!-- Liste non ordonnée -->
2   <li>Pomme</li> <!-- Élément de liste -->
3   <li>Banane</li>
4   <li>Orange</li>
5 </ul>
```



Incorrect :

```
1 <ul><li>Pomme</li><li>Banane</
  li><li>Orange</li></ul>
```



Points importants à retenir : aération du code

Les éléments dits « inline » (span, a, strong, b, em, i) peuvent rester sur la même ligne :

```
<p>Ceci est un paragraphe <strong>très  
1 important</strong>.</p> <!-- strong = gras  
-->
```





II. Balises HTML de base (à utiliser pour les exercices)

Balises textuelles

Paragraphe

```
1 <p>Paragraphe</p>
```



Titres (de <h1> à <h6>)

```
1 <h1>Titre de niveau 1</h1>
```



```
2 <h6>Titre de niveau 6</h6>
```


Balises textuelles

Texte en gras

```
1 <strong>Texte en gras</strong>
```



```
2 <b>Texte en gras</b>
```

Texte en italique

```
1 <em>Texte en italique</em>
```



```
2 <i>Texte en italique</i>
```

Listes

Deux types de listes : **ordonnées** (`ol`) et **non ordonnées** (`ul`).

Chaque élément de la liste est défini par une balise `li`.

Par défaut, pour les listes ordonnées, chaque élément est numéroté. Pour les listes non ordonnées, chaque élément est précédé d'un point.

Listes

Liste ordonnée (ol)

1 ``



2 `Élément 1`

3 `Élément 2`

4 ``

Listes

Liste non ordonnée (ul)

```
1 <ul>
2   <li>Élément 1</li>
3   <li>Élément 2</li>
4 </ul>
```



Images

Pour insérer une image, il faut utiliser l'attribut `src` pour indiquer le lien vers l'image.

Le lien vers l'image peut être soit un **chemin relatif** soit un **chemin absolu**.

- ♦ **chemin relatif** : par rapport à l'emplacement du fichier HTML ;
- ♦ **chemin absolu** : chemin complet du fichier sur l'ordinateur **ou** URL complète de l'image en ligne.

Images

Chemin relatif

```
1 
```



Chemin absolu

```
1 
```



```
2 
```

Remarques sur les liens relatifs et absolus

Il est **déconseillé** d'utiliser des liens absolus quand vous faites référence à une image qui est enregistrée sur votre ordinateur (et non en ligne), car **cela ne fonctionnera pas sur un autre ordinateur**.

Remarques sur les liens relatifs et absolus

Concernant les liens relatifs, on les appelle ainsi car ils sont **relatifs à l'endroit où se trouve le fichier sur l'ordinateur**.

Par exemple, si vous voulez inclure une image qui se nomme `chat.jpg` et qui se trouve dans le même dossier que votre fichier HTML, vous écrirez :

```
1 
```



Remarques sur les liens relatifs et absolus

Par contre, si l'image se trouve **dans un autre dossier** images qui est à l'intérieur du dossier dans lequel se trouve votre fichier HTML, il faudra d'abord spécifier le nom du dossier (dans l'exemple, images) suivi d'une barre oblique / :

```
1 
```



Remarques sur les liens relatifs et absolus

Et si l'image se trouve dans un dossier images **à l'extérieur du dossier**, il faudra utiliser `../` pour remonter d'un niveau :

```
1 
```



Images

D'autres attributs existent pour les images :

- ♦ `alt` : texte alternatif pour l'image ;
- ♦ `width` et `height` : largeur et hauteur de l'image (en pixels).

```
1 
```



Vidéos

Pour insérer une vidéo, on utilise la balise `video` et la balise enfant `source` pour indiquer le lien vers la vidéo.

```
1 <video controls muted>
```



```
2   <source src="video.mp4" type="video/mp4">
```

```
3 </video>
```

Liens

Les liens (<a>) ont toujours un attribut href dont la valeur est la page vers laquelle on veut rediriger.

Le lien peut être absolu (par exemple, adresse URL complète d'un site en ligne) ou relatif (pour inclure une autre page HTML enregistrée localement).

Liens

Lien vers un site Web (absolu)

```
1 <a href="https://www.unistra.fr">Lien vers  
  l'Université de Strasbourg</a>
```



Lien vers une autre page HTML locale (relatif)

```
1 <a href="page2.html">Lien vers la page 2</  
  a>
```



Liens

Adresse e-mail

Il est également possible de créer un lien qui ouvre un client de messagerie avec l'adresse e-mail déjà renseignée. On utilise `mailto:` suivi de l'adresse e-mail comme valeur de `href` :

```
1 <a href="mailto:adresse@gmail.com">Envoyer  
  un e-mail</a>
```



Liens

Ouverture dans un nouvel onglet

Pour ouvrir un lien dans un nouvel onglet, on ajoute l'attribut `target` avec la valeur `_blank` :

```
<a href="https://www.unistra.fr"  
1 target="_blank">Lien vers l'Université de  
Strasbourg</a>
```





III. Mise en pratique

Mise en pratique

La meilleure manière d'apprendre le HTML reste de pratiquer.

Vous retrouverez sur Moodle des exercices pratiques, et vous pourrez ainsi créer vos premières pages Web !