**Welcome to HTML!**

**HTML stands for HyperText Markup Language. Unlike a scripting or programming language that uses scripts to perform functions, a markup language uses tags to identify content.**

* **What does a markup language use to identify content? Tags**
* **Le HTML (Language Markup HyperText) utilise des balises pour identifier le contenu.**

**Introduction**

**HTML (MARKUP HYPERTEXT Language) est le langage standard de création des pages web. Il permet de décrire la structure de la page. Alors que Le CSS (Cascading Style Sheet) est le langage de mise en forme de contenu HTML.**

**Il décrit la façon dont les éléments HTML seront affichées sur la page web.**

**La structure Web**

La capacité à coder en HTML est essentielle pour tout professionnel du web. L'acquisition de cette compétence devrait être le point de départ de toute personne qui apprend à créer du contenu pour le Web.

**Conception Web moderne**

**HTML:** Structure

**CSS:** Présentation

**JavaScript:** Comportement

**PHP ou similaire:** Backend

**CMS:** gestion de contenu

HTML est facile à apprendre. Alors n'attendez pas ! Plonge droit de dans!

Réorganisez le code pour entourer le texte "J'apprends le HTML sur SoloLearn !" avec des balises d'ouverture et de fermeture <p> :

<p> I am learning HTML on Sololearn! </p>

**Structure de base des documents HTML**

**The <html> tag**

Bien que différentes versions aient été publiées au fil des ans, les bases de HTML restent les mêmes.

La structure d'un document HTML a été comparée à celle d'un sandwich. Comme un sandwich a deux tranches de pain, le document HTML a des balises HTML d'ouverture et de fermeture.

Ces balises, comme le pain dans un sandwich, entourent tout le reste :

<html> …</html>

Tout dans un document HTML est entouré par la balise <html>.

Remplissez le champ vide pour créer une balise HTML d'ouverture : <html>

**The <head> Tag**

Immédiatement après la balise HTML d’ouverture, vous trouverez l’en-tête du document, qui est identifié par des balises d’ouverture et de fermeture d’en-tête.

L’en-tête d’un fichier HTML contient tous les éléments non visuels qui aident à faire fonctionner la page.

Essayez de créer un document HTML valide :

<html>   
**<head></head>**  
</html>

Les éléments de la section de tête seront discutés plus tard.

**The body Tag**

La balise body suit la balise head.

**Tous les éléments visuels et structurels sont contenus dans la balise body.** Les titres, les paragraphes, les listes, les citations, les images et les liens ne sont que quelques-uns des éléments qui peuvent être contenus dans la balise body.

<html>   
**<head></head>**

**<body>…</body>**  
</html>

La balise <body> définit le contenu principal du document HTML.

**The HTML File**

Les fichiers HTML sont des fichiers texte, vous pouvez donc utiliser n'importe quel éditeur de texte pour créer votre première page Web. Il existe de très bons éditeurs HTML disponibles ; vous pouvez choisir celui qui vous convient. Pour l'instant, écrivons nos exemples dans le Bloc-notes.

Vous pouvez exécuter, enregistrer et partager vos codes HTML sur notre Code Playground, sans installer de logiciel supplémentaire. N'oubliez pas de sauvegarder le fichier. Les noms de fichiers HTML doivent se terminer par .html ou .htm.

Quelle est la bonne extension pour les fichiers HTML ? .html

Où mettre la balise title ? entre les balises head

Création d'un blog

Remplissez les blancs pour ajouter un titre à la page html.

<html>   
**<head><title>My Blog</title></head>**

**<body>…</body>**  
</html>

**Quelles balises sont utilisées pour indiquer le headings ? <h1>…<h6>**

**Horizontal Line**  
Pour créer a horizontal line, on utilise la balise <hr>.

En HTML5, la balise <hr> définit une rupture thématique.

< !—Commentaires--->

Le navigateur n'affiche pas de commentaires, mais ils aident à documenter le code HTML et à ajouter des descriptions, des rappels et d'autres notes. <!-- Votre commentaire va ici -->

**The <p> Element**

Pour créer un paragraphe, saisissez simplement l'élément <p> avec ses balises d'ouverture et de fermeture :

Les navigateurs ajoutent automatiquement une ligne vide avant et après un paragraphe.

**Single Line Break**

Utilisez la balise <br /> pour ajouter une seule ligne de texte sans commencer un nouveau paragraphe :

L'élément <br /> est un élément HTML vide. L’élément br n'a pas de balise de fin.

Quelle balise est utilisée pour créer un saut de ligne sans espace supplémentaire entre les blocs de texte ? <br />

**Formatting Elements**

En HTML, il existe une liste d'éléments qui spécifient le style de texte.

Les éléments de mise en forme ont été conçus des types de pour afficher texte spéciaux : La balise <strong> est une balise de phrase. Il définit le texte important.

Quelles sont les deux balises ci-dessous qui rendent le texte visuellement en gras ? b et strong.

Les navigateurs affichent <strong> comme <b> et <em> comme <i>. Cependant, les significations de ces balises diffèrent : <b> et <i> définissent respectivement le texte en gras et en italique, tandis que <strong> et <em> indiquent que le texte est "important".

**HTML Elements**

Les documents HTML sont constitués d'éléments HTML.

Un élément HTML est écrit à l'aide d'une balise de début et d'une balise de fin, et avec le contenu entre les deux. Les documents HTML sont constitués d'éléments HTML imbriqués. Dans l'exemple ci-dessous, l'élément body comprend les balises <p>, la balise <br /> et le contenu, "Ceci est un paragraphe".

Certains éléments HTML (comme la balise <br />) n'ont pas de balises de fin.

Les éléments HTML généraux consistent en une balise d'ouverture, un contenu et une balise de fermeture.

Certains éléments sont assez petits. Étant donné que vous ne pouvez pas mettre de contenu dans une balise de rupture et que vous n'avez pas de balise de rupture d'ouverture et de fermeture, il s'agit d'un élément distinct et unique.

Donc, HTML est vraiment un script avec des éléments dans des éléments.

**HTML Attributes**

**Les attributs fournissent des informations supplémentaires sur un élément ou une balise, tout en les modifiant. La plupart des attributs ont une valeur, la valeur modifie l’attribut.**

Les attributs sont toujours spécifiés dans la balise de début et ils apparaissent dans des paires name="value".

**Le Role d’un attribut en HTML est de modifier la balise.**

Exemple : Quel attribut est utilisé pour aligner le contenu d’un élément à droite, au centre ou à gauche ? **attribut : Align.**

**The <img> Tag**

La balise <img> est utilisée pour insérer une image. Il ne contient que des attributs et n'a pas de balise fermante. L'URL (adresse) de l'image peut être définie à l'aide de l'attribut src.

La syntaxe de l'image HTML ressemble à ceci : **<img src="image.jpg" />**

L'attribut alt spécifie un texte alternatif pour une image.

Dans le cas où l'image ne peut pas être affichée, l'attribut alt spécifie un texte alternatif qui décrit l'image en mots. L'attribut alt est obligatoire.

Le chargement des images prend du temps. L'utilisation de grandes images peut ralentir votre page, alors utilisez-les avec précaution.

**HTML Ordered Lists**

Une liste ordonnée commence par la balise <ol> et chaque élément de la liste est défini par la balise <li>.

Les éléments de la liste seront automatiquement marqués de numéros.

**HTML Unordered List**

Une liste non ordonnée commence par la balise <ul>.

Les éléments de la liste seront marqués par des puces.

**Creating a Table**

Les tableaux sont définis à l'aide de la balise <table>. Les tableaux sont divisés en lignes de tableau avec la balise <tr>. Les lignes de tableau sont divisées en colonnes de tableau (données de tableau) avec la balise <td>. Voici un exemple de tableau avec une ligne et trois colonnes.

**<table>   
<tr>   
<td></td>   
<td></td>   
<td></td>   
</tr>   
</table>**

Les balises de données de table <td> agissent comme des conteneurs de données dans la table. Ils peuvent contenir toutes sortes d'éléments HTML, tels que du texte, des images, des listes, d'autres tableaux, etc.

Quelle balise est utilisée pour créer des colnnes en un row ? td

**The border and colspan Attributes**

Une bordure peut être ajoutée à l'aide de l'attribut border : <table **border**="2">

Quel attribut est utilisé pour développer une cellule pour deux cellules ou plus ? colspan

<tr> <td colspan=’’2’’>Orange</td> </tr>

Pour qu'une cellule s'étende sur plusieurs lignes, utilisez l'attribut rowspan.

**The <a> Tag**

Les liens font également partie intégrante de chaque page Web. Vous pouvez ajouter des liens vers du texte ou des images qui permettront à l'utilisateur de cliquer dessus afin d'être dirigé vers un autre fichier ou une autre page Web.

En HTML, les liens sont définis à l'aide de la balise <a>.

Utilisez L**'attribut href pour définir l'adresse de destination du lien**: <a **href**=""></a>

Pour lier une image à un autre document, imbriquez simplement la balise <img> dans les balises <a>.

**Creating Your First Link**

Les liens peuvent être absolus ou relatifs.

Quel attribut de la balise de lien contient l'emplacement de l'URL vers lequel vous essayez d'établir un lien ? href

**The target Attribute**

L'attribut cible spécifie où ouvrir le document lié. En donnant une valeur \_blank à votre attribut, le lien s'ouvrira dans une nouvelle fenêtre ou un nouvel onglet :

<a href=’’http://www.sololearn.com’’ \_blank>Learn Playing</a>, Un lien visité est souligné et violet.

**Types of Elements**

En HTML, la plupart des éléments sont définis comme des éléments de niveau bloc ou en ligne.

Les éléments de niveau bloc commencent à partir d'une nouvelle ligne.

Par exemple : <h1>, <form>, <li>, <ol>, <ul>, <p>, <pre>, <table>, <div>, etc.

Les éléments en ligne sont normalement affichés sans saut de ligne.

Par exemple : <b>, <a>, <strong>, <img>, <input>, <em>, <span>, etc.

L'élément <div> est un élément de niveau bloc qui est souvent utilisé comme conteneur pour d'autres éléments HTML.

Lorsqu'il est utilisé avec certains styles CSS, l'élément <div> peut être utilisé pour styliser des blocs de contenu.

**Question:** Lequel de éléments suivants sont des éléments de niveau bloc : h1, div.

D'autres éléments peuvent être utilisés soit comme **des éléments de niveau de blocs** ou **des éléments en ligne**. Cela inclut les éléments suivants:

Applet - Applet Java incorporé

Iframe - cadre en ligne

INS - Texte inséré

Carte - Carte Image

Objet - Objet intégré

Script - Script dans un document HTML

Vous pouvez insérer des éléments en ligne à l'intérieur des éléments de bloc. Par exemple, vous pouvez avoir plusieurs éléments <span> à l'intérieur d'un élément <div>.

Les éléments en ligne ne peuvent contenir aucun élément de niveau de bloc.

**The <form> Element**

Les formulaires HTML sont utilisés pour collecter des informations de l'utilisateur.

Les formulaires sont définis à l'aide de l'élément <form> avec ses balises d'ouverture et de fermeture:

<body> **<form></form>**</body>

**Utilisez L'attribut action pour pointer vers une page Web qui se chargera une fois que l'utilisateur soumet le formulaire.**

**form action**="http://www.sololearn.com">   
<**/form**>

**Habituellement, le formulaire est soumis à une page Web sur un serveur Web.**

Quel attribut contient l'adresse URL de la page Web chargée après la soumission d'un formulaire ? action

**The method and name Attributes**

L'attribut method spécifie la méthode HTTP (GET ou POST) à utiliser lors de la soumission des formulaires (voir ci-dessous pour la description):

<form action="url" **method="GET"**>

<form action="url" **method="POST"**>

Lorsque vous utilisez GET, les données du formulaire seront visibles dans l'adresse de la page. Utilisez POST si le formulaire met à jour des données ou contient des informations sensibles (mots de passe).

POST offre une meilleure sécurité car les données soumises ne sont pas visibles dans l'adresse de la page.

Pour prendre en compte les entrées de l'utilisateur, vous avez besoin des éléments de formulaire correspondants, tels que les champs de texte. L'élément <input> a de nombreuses variantes, selon l'attribut type. Il peut s'agir d'un texte, d'un mot de passe, d'une radio, d'une URL, d'un envoi, etc.

**L'attribut name spécifie un nom pour un formulaire.**

Si nous changeons le type d'entrée en radio, cela permet à l'utilisateur de sélectionner un seul choix parmi un certain nombre de choix.

Le type "radio" permet à l'utilisateur de sélectionner plus d'une option. La balise <input> n'a pas de balise de fin. Le bouton Soumettre soumet un formulaire à son attribut action.

**Une fois le formulaire soumis, les données doivent être traitées sur le serveur à l'aide d'un langage de programmation, tel que PHP.**

**Question:** Quelle valeur pour l'attribut type transforme la balise d'entrée en bouton d'envoi ? submit

Pour prendre en charge plusieurs lignes d'entrée, vous devez utiliser l'élément suivant : textarea

**HTML Colors!**

Les couleurs HTML sont exprimées en valeurs hexadécimales. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Comme vous pouvez le voir, il y a 16 valeurs, de 0 à F. Zéro représente la valeur la plus basse et F représente la plus élevée.

**Question:** De quels caractères est composé le système hexadécimal ? 0-F

**HTML Color Model**

Les couleurs sont affichées dans des combinaisons de lumière rouge, verte et bleue (RVB). Les valeurs hexadécimales sont écrites à l'aide du symbole hashtag (#), suivi de trois ou six caractères hexadécimaux.

Comme le montre l'image ci-dessous, les cercles se chevauchent, formant de nouvelles couleurs :

**Question:** quel color model utilise le HTML ? RGB

**Background and Font Colors**

L'attribut bgcolor peut être utilisé pour changer la couleur d'arrière-plan de la page Web. Cet exemple produirait un fond bleu foncé avec un titre blanc.

L'attribut color spécifie la couleur du texte à l'intérieur d'un élément <font>.

**The <frame> Tag**

Une page peut être divisée en cadres à l'aide d'un document cadre spécial. La balise <frame> définit une fenêtre spécifique (frame) dans un <frameset>. Chaque <frame> dans un <frameset> peut avoir différents attributs, tels que la bordure, le défilement, la possibilité de redimensionner, etc.

L'élément <frameset> spécifie le nombre de colonnes ou de lignes dans le jeu de cadres, ainsi que le pourcentage ou le nombre de pixels d'espace que chacun d'eux occupe.

**framesetcols**="100, 25%, \*"><**/frameset**>   
<**frameset rows**="100, 25%, \*"><**/frameset**>

La balise <frameset> n'est pas prise en charge dans HTML5.

**Question:** De quels attributs la balise frameset a-t-elle besoin pour indiquer sa taille ? lignes et cols

**Working with Frames**

Utilisez l'attribut <noresize> pour spécifier qu'un utilisateur ne peut pas redimensionner un élément <frame> :

<frame **noresize**="**noresize**">

Le contenu du cadre doit être défini à l'aide de l'attribut **src**. Enfin, l'élément **<noframes>** permet aux navigateurs qui ne prennent pas en charge les cadres d'afficher la page. L'élément peut contenir une page alternative, complétée par une balise body et tout autre élément.

<frameset cols="25%,50%,25%">   
<frame src="a.htm" />   
<frame src="b.htm" />   
<frame src="c.htm" />   
<noframes>Frames not supported!</noframes>   
</frameset>

La balise <frame> n'est pas prise en charge dans HTML5.

**Blog Project**

Pour finaliser notre blog, nous utiliserons un cadre pour intégrer une vidéo YouTube. Nous allons également créer une section Suivez-moi qui comprend des liens à la fin de la page.

TÂCHE:

1. Finalisez votre page de blog.

2. Partagez votre création de code avec la communauté, obtenez des commentaires et recevez des votes positifs !

**Question:** Les sections du projet de blog sont créées à l'aide de la balise suivante : div

**Question:** Lorsque vous formatez du texte, pouvez-vous obtenir le même résultat en utilisant différentes balises ? Oui

**Question:** Que contient l'attribut href ? L'URL de la page à transférer.

**Question:** Quelle balise contient les balises de cellule en plus de la balise de tableau ? <tr>

**New in HTML5**

Form

- La spécification Web Forms 2.0 permet de créer des formulaires plus puissants et des expériences utilisateur plus attrayantes.

- Des sélecteurs de dates, des sélecteurs de couleurs et des commandes numériques pas à pas ont été ajoutés.

- Les types de champs de saisie incluent désormais l'e-mail, la recherche et l'URL.

- Les méthodes de formulaire PUT et DELETE sont désormais prises en charge.

API intégrée (interfaces de programmation d'applications)

- Glisser déposer

- Audio et vidéo

- Applications Web hors ligne

- Histoire

- Stockage local

- Géolocalisation

- Messagerie Web

Vous en apprendrez plus sur ces nouvelles fonctionnalités dans les leçons à venir. Appuyez sur le bouton Continuer pour commencer !

**Question:** Quel élément/fonctionnalité n'est pas nouveau dans HTML5 ? Image

**The List of Content Models**

En HTML, les éléments appartenaient généralement au modèle de contenu au niveau du bloc ou en ligne. HTML5 introduit sept principaux modèles de contenu.

- Metadata  
- Embedded  
- Interactive  
- Heading  
- Phrasing  
- Flow  
- Sectioning

Les modèles de contenu HTML5 sont conçus pour rendre la structure de balisage plus significative à la fois pour le navigateur et le concepteur Web.

**Question:** Combien de modèles de contenu propose HTML5 ? 7

**Content Models**

Metadata: Content that sets up the presentation or behavior of the rest of the content. These elements are found in the head of the document.  
Elements: <base>, <link>, <meta>, <noscript>, <script>, <style>, <title>  
  
Embedded: Content that imports other resources into the document.  
Elements: <audio>, <video>, <canvas>, <iframe>, <img>, <math>, <object>, <svg>  
  
Interactive: Content specifically intended for user interaction.  
Elements: <a>, <audio>, <video>, <button>, <details>, <embed>, <iframe>, <img>, <input>, <label>, <object>, <select>, <textarea>  
  
Heading: Defines a section header.  
Elements: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <hgroup>  
  
Phrasing: This model has a number of inline level elements in common with HTML4.  
Elements: <img>, <span>, <strong>, <label>, <br />, <small>, <sub>, and more.

The same element can belong to more than one content model.

Contenu du flux : Contient la majorité des éléments HTML5 qui seraient inclus dans le flux normal du document.

Sectionner le contenu : définit la portée des en-têtes, du contenu, de la navigation et des pieds de page.

Éléments : <article>, <aside>, <nav>, <section>

Les différents modèles de contenu se chevauchent dans certains domaines, selon la manière dont ils sont utilisés.

**Question:** Quel modèle de contenu contient presque tous les autres ? Flow

**Page Structure in HTML5**

Une structure de page HTML5 générique ressemble à ceci :

1 : <header>

2: <nav>

3: <article>

4: <section>

5: <footer>

Vous n'aurez peut-être pas besoin de certains de ces éléments, selon la structure de votre page.

**The <header> Element**

En HTML4, nous définirions un en-tête comme celui-ci : <div id="header">

En HTML5, une simple balise <header> est utilisée à la place. L'élément <header> est approprié pour être utilisé à l'intérieur de la balise body.

Notez que le <header> est complètement différent de la balise <head>.

L'élément header est approprié pour utiliser... …dans la balise body.

**The <footer> Element**

L'élément footer de page est également largement utilisé. Généralement, nous nous référons à une section située tout en bas de la page Web en tant que footer. **<footer></footer>**

Les informations suivantes sont généralement fournies entre ces balises :

- Contact Information  
- Privacy Policy  
- Social Media Icons  
- Terms of Service  
- Copyright Information  
- Sitemap and Related Documents

Question : Quelle balise spécifie l'élément footer ? <footer>

**The <nav> Element**

Cette balise représente une section d'une page qui renvoie à d'autres pages ou à certaines sections de la page. Ce serait une section avec des liens de navigation.

Tous les liens d'un document ne doivent pas se trouver à l'intérieur d'un élément <nav>. L'élément <nav> est destiné uniquement aux principaux blocs de liens de navigation. Typiquement, l'élément <footer> a souvent une liste de liens qui n'ont pas besoin d'être dans un élément <nav>.

Réorganisez les sections pour créer une structure de page HTML5 générique :

1: <head>

2 : <header>

3 : <nav>

4 : <footer>

**The <article> Element**

L'article est un élément de contenu autonome et indépendant qui peut être utilisé et distribué séparément du reste de la page ou du site. Il peut s'agir d'un message sur un forum, d'un article de magazine ou de journal, d'une entrée de blog, d'un commentaire, d'un widget ou d'un gadget interactif ou de tout autre élément de contenu indépendant. L'élément <article> remplace l'élément <div> qui était largement utilisé dans HTML4, avec un identifiant ou une classe.

Lorsqu'un élément <article> est imbriqué, l'élément interne représente un article lié à l'élément externe. Par exemple, les commentaires d'un article de blog peuvent être des éléments <article> imbriqués dans l'élément <article> représentant l'article de blog.

**Question:** Quel élément était généralement utilisé en HTML4 à la place de la balise article ? div

**The <section> Element**

<section> est un conteneur logique de la page ou de l'article. Les sections peuvent être utilisées pour diviser le contenu d'un article. Par exemple, une page d'accueil peut avoir une section pour présenter l'entreprise, une autre pour les actualités et encore une autre pour les coordonnées.

Chaque <section> doit être identifiée, généralement en incluant un titre (élément h1-h6) en tant qu'enfant de l'élément <section>.

S'il est logique de syndiquer séparément le contenu d'un élément <section>, utilisez un élément <article> à la place.

**Question:** L'élément section doit être utilisé uniquement à l'intérieur d'un élément article. False

Lorsqu'une balise <aside> est utilisée en dehors d'une balise <article>, son contenu doit être lié au contenu environnant.

L'élément aside est utilisé pour définir : secondary content.

**Audio on the Web**

Avant HTML5, il n'existait aucune norme pour la lecture de fichiers audio sur une page Web.

L'élément HTML5 <audio> spécifie une norme pour l'intégration de l'audio dans une page Web.

+ Il existe deux manières différentes de spécifier l'URL du fichier source audio. Le premier utilise l'attribut source.

+ La deuxième méthode utilise l'élément <source> à l'intérieur de l'élément <audio>.

Plusieurs éléments <source> peuvent être liés à différents fichiers audio. Le navigateur utilisera le premier format reconnu.

Le texte entre les balises <audio> et </audio> s'affichera dans les navigateurs qui ne prennent pas en charge l'élément <audio>.

**Question:** Le texte entre les balises audio est : pour prendre en charge plusieurs types audio.

**Attributes of <audio>**

contrôles

Spécifie que les commandes audio doivent être affichées (telles qu'un bouton de lecture/pause, etc.)

lecture automatique

Lorsque cet attribut est défini, l'audio commence à jouer dès qu'il est prêt, sans demander la permission du visiteur.

boucle

Cet attribut est utilisé pour que l'audio soit rejoué à chaque fois qu'il est terminé.

Actuellement, il existe trois formats de fichiers pris en charge pour l'élément <audio> : MP3, WAV et OGG.

**Videos in HTML**

L'élément vidéo est similaire à l'élément audio. Vous pouvez spécifier l'URL de la source vidéo à l'aide d'un attribut dans un élément vidéo ou à l'aide d'éléments source à l'intérieur de l'élément vidéo.

Un autre aspect que les éléments audio et vidéo ont en commun est que les principaux navigateurs ne prennent pas tous en charge les mêmes types de fichiers. Si le navigateur ne prend pas en charge le premier type de vidéo, il essaiera le suivant.

<video controls autoplay loop>

<source src= ‘’video.mp4’’ type= ‘’video/mp4’’ />

<source src= ‘’video.ogg’’ type= ‘’video/ogg’’ />

Video Not Supported by your browser

</video>

**Attributes of <video>**

Un autre aspect partagé par les éléments audio et vidéo est que chacun a des attributs de contrôle, de lecture automatique et de boucle.

Actuellement, il existe trois formats vidéo pris en charge pour l'élément <video> : MP4, WebM et OGG.

Réorganisez le code pour créer une balise vidéo valide qui tentera de lire le fichier mp4 en premier.

<video controls>

<source src=’’a.mp4’’ type=’’video/mp4’’ />

<source src=’’a.ogg’’ type=’’video/ogg’’ />

Video is Not Supported

</video>

**Progress Bar**

L'élément **<progress>** permet de créer des barres de progression sur le Web.

L'élément de progression peut être utilisé dans les titres, les paragraphes ou n'importe où ailleurs dans le corps.

**Attributs d'élément de progression**

Valeur : spécifie la proportion de la tâche qui a été achevée.

Max : spécifie la quantité de travail requise par la tâche au total.

Utilisez la balise <progress> conjointement avec JavaScript pour afficher dynamiquement la progression d'une tâche.

**HTML5 Web Storage**

Avec le stockage Web HTML5, les sites Web peuvent stocker des données sur l'ordinateur local d'un utilisateur.

Avant HTML5, nous devions utiliser des **cookies JavaScript** pour obtenir cette fonctionnalité.

**Les avantages du stockage Web**

- Plus sécurisé

- Plus rapide

- Stocke une plus grande quantité de données

- Les données stockées ne sont pas envoyées à chaque requête du serveur

Le stockage local est par domaine. Toutes les pages d'un domaine peuvent stocker et accéder aux mêmes données.

Question : Avant HTML5, les données d'application étaient stockées dans : cookies.

**Types of Web Storage Objects**

Il existe deux types d'objets de stockage Web :

**- sessionStorage()**

**- stockage local()**

**Local vs Session**

- Le stockage de session est détruit une fois que l'utilisateur ferme le navigateur

- Le stockage local stocke les données sans date d'expiration

Vous devez être familiarisé avec JavaScript de base afin de comprendre et d'utiliser l'API.

**Working with Values**

La syntaxe du stockage Web pour le stockage local et de session est très simple et similaire.

Les données sont stockées sous forme de paires clé/valeur.

**Storing a Value:**

localStorage.setItem("key1", "value1");

JS

Getting a Value:

//this will print the value   
alert(localStorage.getItem("key1"));

JS

Removing a Value:

localStorage.removeItem("key1");

JS

Removing All Values:

localStorage.clear();

La même syntaxe s'applique au stockage de session, avec une différence : au lieu de localStorage, sessionStorage est utilisé.

Faites glisser et déposez les options ci-dessous pour effacer toutes les valeurs stockées dans le localStorage. Puis mémorisez "a" à l'aide de la touche "b".

localStorage.clear() ;

localStorage.setItem("b", "a") ;

**What is the Geolocation API?**

En HTML5, l'API de géolocalisation est utilisée pour obtenir la position géographique de l'utilisateur.

Comme cela peut compromettre la confidentialité de l'utilisateur, l'option n'est disponible que si l'utilisateur l'approuve.

La géolocalisation est beaucoup plus précise pour les appareils avec GPS, comme les smartphones et autres.

Avec l’API de géolocalisation, vous pouvez obtenir « user location ».

**Using HTML Geolocation**

La méthode principale de l'API de géolocalisation est getCurrentPosition, qui récupère l'emplacement géographique actuel de l'appareil de l'utilisateur.

navigator.geolocation.getCurrentPosition();

Paramètres:

showLocation (obligatoire) : définit la méthode de rappel qui récupère les informations de localisation.

ErrorHandler (facultatif) : définit la méthode de rappel appelée lorsqu'une erreur se produit lors du traitement de l'appel asynchrone.

Options (facultatif) : Définit un ensemble d'options pour récupérer les informations de localisation.

Vous devez être familiarisé avec JavaScript de base afin de comprendre et d'utiliser l'API.

**Question:** Quel choix est le paramètre obligatoire de la méthode getCurrentPosition() ? showLocation

**Presenting Data**

L'emplacement de l'utilisateur peut être présenté de deux manières : Geodetic et Civic.

1. La façon géodésique de décrire la position fait directement référence à la latitude et à la longitude.

2. La représentation civique des données de localisation est présentée dans un format qui est plus facilement lu et compris par la personne moyenne.

Chaque paramètre a à la fois une représentation Geodetic et Civic :



La méthode getCurrentPosition() renvoie un objet si elle réussit. Les propriétés de latitude, de longitude et de précision sont toujours renvoyées.

**Question:** quelles sont les 2 types connues de présenter des données spécifiques à un lieu ? Geodetic et Civic.

**Making Elements Draggable**

La fonction glisser-déposer vous permet de "saisir" un objet et de le faire glisser vers un emplacement différent. Pour rendre un élément déplaçable, il suffit de définir l'attribut draggable sur true :

<img **draggable**="**true**" />

Tout élément HTML peut être déplaçable.

L'API pour le glisser-déposer HTML5 est basée sur les événements.

<!DOCTYPE HTML>   
<html>   
<head>   
<script>   
function allowDrop(ev) {   
ev.preventDefault();   
}   
function drag(ev) {   
ev.dataTransfer.setData("text", ev.target.id);   
}   
function drop(ev) {   
ev.preventDefault();   
var data = ev.dataTransfer.getData("text");   
ev.target.appendChild(document.getElementById(data));   
}   
</script>   
</head>   
<body>   
<div id="box" **ondrop**="drop(event)"   
**ondragover**="allowDrop(event)"   
style="border:1px solid black;   
width:200px; height:200px"></div>   
<img id="image" src="sample.jpg" **draggable**="true"   
**ondragstart**="drag(event)" width="150" height="50" alt="" />   
</body>   
</html>

What to Drag :

Lorsque l'élément est glissé, l'attribut ondragstart appelle une fonction, drag(event), qui spécifie quelles données doivent être glissées.

La méthode dataTransfer.setData() définit le type de données et la valeur des données glissées :

function drag(ev) {   
ev.dataTransfer.setData("text", ev.target.id);   
}

Dans notre exemple, le type de données est "texte" et la valeur est l'ID de l'élément déplaçable ("image").

Where to drag:

L'événement ondragover spécifie où les données glissées peuvent être déposées. Par défaut, les données et les éléments ne peuvent pas être déposés dans d'autres éléments. Pour autoriser un drop, il faut empêcher la gestion par défaut de l'élément.

Cela se fait en appelant la méthode event.preventDefault() pour l'événement ondragover.

Do the Drop:

Lorsque les données glissées sont supprimées, un événement de suppression se produit.

Dans l'exemple ci-dessus, l'attribut ondrop appelle une fonction, drop(event) :

function drop(ev) {   
ev.preventDefault();   
var data = ev.dataTransfer.getData("text");   
ev.target.appendChild(document.getElementById(data));   
}

La méthode preventDefault() empêche la gestion par défaut des données par le navigateur (la valeur par défaut est open as link on drop). Les données glissées sont accessibles avec la méthode dataTransfer.getData(). Cette méthode renverra toutes les données qui ont été définies sur le même type dans la méthode setData().

Les données glissées sont l'ID de l'élément glissé ("image").

À la fin, l'élément glissé est ajouté à l'élément déposé, à l'aide de la fonction appendChild().

Une connaissance de base de JavaScript est nécessaire pour comprendre et utiliser l'API.

**Question:** Combien de fois HTML5 peut-il être renvoyé ? Multiple

**Drawing Shapes**

SVG signifie Scalable Vector Graphics et est utilisé pour dessiner des formes avec un balisage de style HTML.

Il offre plusieurs méthodes pour dessiner des chemins, des boîtes, des cercles, du texte et des images graphiques.

SVG n'est pas basé sur les pixels, il peut donc être agrandi à l'infini sans perte de qualité.

**Inserting SVG Images**

Une image SVG peut être ajoutée au code HTML avec juste une balise d'image de base qui inclut un attribut source pointant vers l'image :

<img **src**="image.**svg**" alt="" height="300" />

SVG définit les graphiques vectoriels au format XML.

**Drawing a Circle**

Pour dessiner des formes avec SVG, vous devez d'abord créer une balise d'élément SVG avec deux attributs : largeur et hauteur. **svg width**="1000" **height**="1000">**</svg>**

Pour créer un cercle, ajoutez une balise <circle> :

- cx pousse le centre du cercle plus à droite de l'écran.

- cy pousse le centre du cercle plus bas depuis le haut de l'écran.

- r définit le rayon.

- fill détermine la couleur de notre cercle.

- stroke ajoute un contour au cercle.

Chaque élément et chaque attribut des fichiers SVG peut être animé.

**Other Shape Elements**

<rect> définit un rectangle:

<svg width="2000" height="2000">

     <rect width="300" height="100" x="20" y="20" fill="green" />

</svg>

Le code suivant dessinera un rectangle rempli de vert.

<line> définit un segment de ligne :

<svg width="400" height="410">

     <line x1="10" y1="10" x2="200" y2="100"

          style="stroke:#000000; stroke-linecap:round; stroke-width:20" />

</svg>

(x1, y1) définissent les coordonnées de départ(x2, y2) définissent les coordonnées de fin.

<polyline> définit des formes construites à partir de plusieurs définitions de lignes :

Les points sont les coordonnées de la polyligne. Le code ci-dessous dessinera une coche noire :

<svg width="2000" height="500">

    <polyline style="stroke-linejoin:miter; stroke:black; stroke- width:12; fill:none;"

    points="100 100, 150 150, 200 100" />

</svg>

Les attributs width et height de l'élément <rect> définissent la hauteur et la largeur du rectangle.

**<ellipse> and <polygon>**

Ellipse

L'<ellipse> est similaire au <cercle>, à une exception près :

Vous pouvez modifier indépendamment les axes horizontal et vertical de son rayon, en utilisant les attributs rx et ry.

<svg height="500" width="1000">

    <ellipse cx="200" cy="100" rx="150" ry="70" style="fill:green" />

</svg>

Polygone

L'élément <polygon> est utilisé pour créer un graphique avec au moins trois côtés. L'élément polygone est unique car il ferme automatiquement la forme pour vous.

<svg height="500" width="1000">

    <polygon points="100 100, 200 200, 300 0" style="fill:green; stroke:black;" />

</svg>

Polygon vient du grec. "Poly" signifie "plusieurs" et "gon" signifie "angle".

**Shape Animations**

Les animations SVG peuvent être créées à l'aide de l'élément <animate>. L'exemple ci-dessous crée un rectangle qui changera de position en 3 secondes et répétera ensuite l'animation deux fois :

<svg height="250" width="1000">

    <rect width="150" height="150" fill="orange">

      <animate attributeName="x" from="0" to="150" dur="3s" fill="freeze"

               repeatCount="2"/>

    </rect>

</svg>

attributeName : spécifie quel attribut sera affecté par l'animation

from : spécifie la valeur de départ de l'attribut

to : spécifie la valeur de fin de l'attribut

dur : spécifie la durée d'exécution de l'animation (durée)

fill : spécifie si la valeur de l'attribut doit ou non revenir à sa valeur initiale lorsque l'animation est terminée (Valeurs : "remove" réinitialise la valeur ; "freeze" conserve le "to value")

repeatCount : spécifie le nombre de répétitions de l'animation

Dans l'exemple ci-dessus, le rectangle change son attribut x de 0 à 300 en 3 secondes.

Pour répéter l'animation indéfiniment, utilisez **la valeur "indefinite"** pour l'attribut **repeatCount**.

**Paths**

L'élément <path> est utilisé pour définir un path.

Les commandes suivantes sont disponibles pour les données de path :

M : passer à

L : ligne vers

H : ligne horizontale

V : ligne verticale

C : courbe vers

S : courbe lisse

Q : courbe de Bézier quadratique

T : courbe de Bézier quadratique lisse

A : Arc elliptique

Z : chemin proche

Définissez un path à l'aide de l'attribut d :

<svg height="500" width="1000">

    <path d="M 0 0 L200 200 L200 0 Z" style="stroke:#000; fill:none;" />

</svg>

M place notre "stylo virtuel" vers le bas à la position 0, 0. Il se déplace ensuite de 200 px vers le bas et vers la droite, puis remonte jusqu'à la position 200, 0. La commande Z ferme la forme, ce qui entraîne une hypoténuse :

Toutes les commandes ci-dessus peuvent également être exprimées avec des lettres minuscules. Lorsque des lettres majuscules sont utilisées, cela indique la position absolue ; les minuscules indiquent la position relative.

Question : Quelle forme est indiquée par le chemin suivant ?

<path d="M0 0 L0 100 L100 100 L100 0 Z" /> Square

**The <canvas> Element**

**Le canevas HTML est utilisé pour dessiner des graphiques qui incluent tout, des lignes simples aux objets graphiques complexes.**

L'élément <canvas> est défini par :

**<canvas** id="canvas1" width="200" height="100">   
<**/canvas**>

**L'élément <canvas> n'est qu'un conteneur pour les graphiques. Vous devez utiliser un script pour dessiner les graphiques (généralement JavaScript).**

L'élément <canvas> doit avoir un attribut id pour qu'il puisse être référencé par JavaScript :

<html>   
<head></head>   
<body>   
**<canvas id**="canvas1"   
width="400" height="300">**</canvas>**  
<script>   
var can = document.getElementById("canvas1");   
var ctx = can.getContext("2d");   
</script>   
  
</body>   
</html>

getContext() renvoie un contexte de dessin sur le canevas.

Une connaissance de base de JavaScript est nécessaire pour comprendre et utiliser le canevas.

Question : vous pouvez dessiner le canvas en utilisant le JavaScript.

**Canvas Coordinates**

Le canvas HTML est une grille à deux dimensions.

Le coin supérieur gauche du canevas a les coordonnées (0,0).

La coordonnée X augmente vers la droite.

La coordonnée Y augmente vers le bottom du canvas.

L'élément <canvas> n'est qu'un conteneur pour les graphiques.

Question : X et Y sont ...coordonnées à partir du coin supérieur gauche.

**Drawing Shapes**

La méthode fillRect(x, y, w, h) dessine un rectangle "rempli", dans lequel w indique la largeur et h indique la hauteur. La couleur de remplissage par défaut est le noir.

Un rectangle noir de 100\*100 pixels est dessiné sur le canevas à la position (20, 20) :

La propriété fillStyle est utilisée pour définir une couleur, un dégradé ou un motif pour remplir le dessin. L'utilisation de cette propriété vous permet de dessiner un rectangle rempli de vert.

var can = document.getElementById("canvas1")

var ctx = can.getContext("2d")

ctx.fillStyle = "rgba(0, 0, 0, 1)"

ctx.fillRect(20, 20, 100, 100)

Le canevas prend en charge diverses autres méthodes de dessin :

Tracer une ligne

moveTo(x,y) : définit le point de départ de la ligne.

lineTo(x,y) : définit le point de fin de la ligne.

Dessiner un cercle

beginPath() : démarre le dessin.

arc(x,y,r,start,stop) : définit les paramètres du cercle.

stroke() : Termine le dessin.

Dégradés

createLinearGradient(x,y,x1,y1) : Crée un dégradé linéaire.

createRadialGradient(x,y,r,x1,y1,r1) : crée un dégradé radial/circulaire.

Dessiner du texte sur la toile

Police : Définit les propriétés de la police du texte.

fillText(text,x,y) : dessine du texte "rempli" sur le canevas.

strokeText(text,x,y) : dessine du texte sur le canevas (pas de remplissage).

Il existe de nombreuses autres méthodes visant à aider à dessiner des formes et des images sur le canvas.

**Canvas vs. SVG**

Canvas

- Les éléments sont dessinés par programmation

- Le dessin est fait avec des pixels

- Les animations ne sont pas intégrées

- Haute performance pour les opérations de dessin basées sur les pixels

- Dépend de la résolution

- Pas de support pour les gestionnaires d'événements

- Vous pouvez enregistrer l'image résultante au format .png ou .jpg

- Bien adapté aux jeux à forte intensité graphique

SVG

- Les éléments font partie du DOM (Document object model) de la page

- Le dessin est fait avec des vecteurs

- Des effets, tels que des animations sont intégrés

- Basé sur la syntaxe XML standard, qui offre une meilleure accessibilité

- Indépendant de la résolution

- Prise en charge event handlers

- Ne convient pas aux applications de jeu

- Idéal pour les applications avec de grandes zones de rendu (par exemple, Google Maps)

Vous pouvez en fait utiliser à la fois SVG et canvas sur la même page, si nécessaire. Cependant, vous ne pouvez pas simplement dessiner SVG sur un canvas, ou vice-versa.

**Working with Canvas**

L'élément Canvas peut être transformé. A titre d'exemple, un texte est écrit sur la toile aux coordonnées (10, 30).

La méthode translate(x,y) est utilisée pour déplacer le Canvas.

x indique jusqu'où déplacer la grille horizontalement et y indique jusqu'où déplacer la grille verticalement.

Dans cet exemple, le canevas est déplacé de 100 pixels vers la droite et de 150 pixels vers le bas. Canvas propose plusieurs méthodes pour dessiner des chemins, des boîtes, des cercles, du texte et ajouter des images.

**The rotate() Method**

La méthode rotate() est utilisée pour faire pivoter le canevas HTML5. La valeur doit être en radians et non en degrés. Voici un exemple qui dessine le même rectangle avant et après que la rotation soit définie :

ctx.fillStyle = "#FF0000"

ctx.fillRect(50, 50, 100, 100)

ctx.rotate((Math.PI / 180) \* 25)

ctx.fillStyle = "#0000FF"

ctx.fillRect(50, 50, 100, 100)

La rotation n'affectera que les dessins réalisés après la rotation.

Question : Le paramètre de rotation est : Radians.

**The scale() Method**

La méthode scale() met à l'échelle le dessin courant. Il prend deux paramètres :

- Le nombre de fois par lequel l'image doit être mise à l'échelle dans la direction X.

- Le nombre de fois par lequel l'image doit être mise à l'échelle dans la direction Y.

ctx.fillRect(50, 50, 100, 100)

ctx.font="bold 22px Tahoma"

ctx.textAlign="start"

ctx.fillText("start", 15, 30)

ctx.translate(100, 150)

ctx.fillText("After translate", 0, 0)

ctx.rotate(1)

ctx.fillText("After rotate", 0, 0)

ctx.scale(1.5, 4)

ctx.fillText("After Scale", 0, 20)

Cela redimensionnera le canevas 1,5 fois dans la direction X et 4 fois dans la direction Y :

Si vous mettez un dessin à l'échelle, tous les futurs dessins seront également mis à l'échelle.

Question : Quelle méthode est utilisée pour augmenter ou diminuer la taille du dessin courant ?

Scale Méthode

**HTML5 Forms**

HTML5 apporte de nombreuses fonctionnalités et améliorations à la création de formulaires Web. De nouveaux attributs et types d'entrée ont été introduits pour aider à créer de meilleures expériences pour les utilisateurs Web.

La création de formulaire se fait en HTML5 de la même manière qu'en HTML4 :

Utilisez l'attribut novalidate pour éviter la validation du formulaire lors des soumissions.

**New Attributes**

HTML5 a introduit un nouvel attribut appelé espace réservé « placeholder ». Sur les éléments <input> et <textarea>, cet attribut fournit une indication à l'utilisateur sur les informations pouvant être saisies dans le champ.

L'attribut autofocus met l'accent sur l'entrée souhaitée lors du chargement du formulaire.

L'attribut required indique au navigateur que l'entrée est requise.

**Forms with Required Fields**

L'attribut "required" est utilisé pour rendre les éléments d'entrée obligatoires.

Le formulaire ne sera pas soumis sans avoir rempli les champs obligatoires.

L'attribut de saisie autocomplete spécifie si un formulaire ou un champ de saisie doit avoir la saisie semi-automatique activée ou désactivée. Lorsque la saisie semi-automatique est activée, le navigateur complète automatiquement les valeurs en fonction des valeurs que l'utilisateur a saisies auparavant.

HTML5 a ajouté plusieurs nouveaux types d'entrée:

**- color  
- date  
- datetime  
- datetime-local  
- email  
- month  
- number  
- range  
- search  
- tel  
- time  
- url  
- week**

Nouveaux attributs d'entrée dans HTML5 :

**- autofocus  
- form  
- formaction  
- formenctype  
- formmethod  
- formnovalidate  
- formtarget  
- height and width  
- list  
- min and max  
- multiple  
- pattern (regexp)  
- placeholder  
- required  
- step**

**Les types d'entrée qui ne sont pas pris en charge par les anciens navigateurs Web se comporteront comme du texte de type d'entrée.**

**Creating a Search Box**

Le nouveau type d'entrée de recherche peut être utilisé pour créer un champ de recherche:

<input type="mysearch" name="searchitem" type="search" />

Vous devez vous rappeler de définir un nom pour votre entrée ; sinon, rien ne sera soumis.

**Search Options**

La balise <datalist> peut être utilisée pour définir une liste d'options prédéfinies pour le champ de recherche :

<input id="car" type="text" list="colors" />

   <datalist id="colors">

      <option value="Red">

      <option value="Green">

      <option value="Yellow">

   </datalist>

<option> définit les options dans une liste déroulante que l'utilisateur peut sélectionner.

L'ID de l'élément datalist doit correspondre à l'attribut list de la zone de saisie.

**Creating More Fields**

Certains autres nouveaux types d'entrée incluent l'e-mail, l'url et le tél :

Celles-ci sont particulièrement utiles lors de l'ouverture d'une page sur un appareil mobile moderne, qui reconnaît les types de saisie et ouvre un clavier correspondant correspondant au type du champ :

Ces nouveaux types facilitent la structuration et la validation des formulaires HTML.

Question : Lequel des types suivants n'est pas pris en charge pour la balise d'entrée ? Planet choix

**Module Quiz**

Question : Quel élément HTML5 est le bon pour lire des fichiers vidéo ? <video>

Question : L'élément <canvas> en HTML5 est utilisé pour : dessiner des graphiques

Question : Quelle balise contient la navigation ? <nav>

Question : sessionStorage stocke les données pour la durée de combien de session(s) ? 1

Question : Quelle forme résulte du code suivant ?

<svg largeur="100" hauteur="100">

<ligne x1="50" y1="0" x2="50" y2="100"

style="fill:black" />

<ligne x1="0" y1="50" x2="100" y2="50"

style="fill:black" />

</svg>

Signe plus

Question : Remplissez les blancs pour désactiver la saisie semi-automatique et exiger le champ du mot de passe.

<form autocomplete="off">

<input type="text" name="nom" />

<input type="mot de passe" nom="passer" required />

</form>