

Conception Web

Partie Théorique

Questions vidéo 1 et 2

Répondre aux questions suivantes :

1. C'est quoi le web ?
2. C'est quoi le HTML ?
3. C'est quoi l'hypertexte ?
4. C'est quoi un navigateur ?
5. Où est-ce que le navigateur va chercher les pages web ?
6. C'est quoi un serveur Web ?
7. Est-ce qu'on a besoin d'utiliser un serveur web local pour accéder à une page web enregistrée localement ?
8. Comment faire pour publier notre site web et le rendre accessible à tout le monde ?
9. Que signifie le mot HTTP retrouvé dans une adresse d'une page web ?

Vidéo 4 et 5

Répondre aux questions suivantes :

1. Quels sont les concepts auxquels il faut faire attention en programmation ?
2. Peut-on utiliser MS Word pour éditer des pages HTML ?
3. Quelles sont les caractéristiques les plus importantes dans un éditeur de codes tel que Visual Studio Code ?

Questions vidéos 8 à 12

Répondre aux questions suivantes :

1. Que signifient les mots suivants retrouvés dans le langage HTML : texte, balise, référence ?
2. C'est quoi une balise ? C'est quoi un élément ?
3. Quelles sont les balises qui sont dans la vidéo et quelle est leur signification ?
4. Comment insérer un saut de ligne en HTML ?
5. Comment insérer une image ?
6. Comment appelle-t-on les éléments sans balise fermante ?
7. Comment insérer un paragraphe en HTML ?
8. Quelle est la différence entre `<p>...</p>` et `
` ?
9. Donner la signification des balises : mark, sub et sup ?
10. Comment ajouter un titre principal à un document HTML ?
11. Combien y-a-t-il de niveaux de titres en HTML ?

Questions vidéos 14 à 16

Répondre aux questions suivantes :

1. Quelle est la différence entre les éléments de bloc et les éléments en ligne ? Citer des exemples.
2. Quelle est la différence entre une liste ordonnée et une liste non ordonnée ?
3. Quelles sont les balises utilisées pour chacune des deux types ?
4. Comment ajouter des éléments dans une liste ?
5. Quelle est l'abréviation (nom plus court) d'une adresse Web ?

6. Donner la signification de chacune de ses parties ?
7. Quel est le protocole utilisé pour les pages enregistrées localement ?

Questions vidéos 17 à 20

Répondre aux questions suivantes :

1. Comment créer un lien hypertexte ?
2. Comment insérer une image ? Quelle est la signification de chacun des attributs ?
3. Quelle est la différence entre une adresse relative et une adresse absolue ?
4. Donner la structure minimale d'un document HTML ? Expliquer chacune de ses parties.

Réponses

1. C'est quoi le web ?

Le Web (ou World Wide Web) est un système de pages et de contenus interconnectés accessibles via Internet. Il utilise des technologies comme l'HTML, les navigateurs web et les protocoles pour permettre aux utilisateurs de consulter et d'échanger des informations.

2. C'est quoi le HTML ?

HTML (HyperText Markup Language) est le langage utilisé pour créer et structurer le contenu des pages web. Il permet de définir les titres, paragraphes, images, liens, tableaux, et bien plus, pour qu'ils soient affichés correctement par les navigateurs.

3. C'est quoi l'hypertexte ?

L'hypertexte est un système qui relie des mots ou des phrases (sous forme de liens cliquables) à d'autres documents, pages web ou sections. C'est l'une des bases du fonctionnement du Web, permettant la navigation d'une page à une autre.

4. C'est quoi un navigateur ?

Un navigateur web est un logiciel utilisé pour accéder et afficher les pages web. Il interprète les langages comme HTML, CSS et JavaScript pour présenter les informations de manière lisible. Exemples : Google Chrome, Firefox, Safari.

5. Où est-ce que le navigateur va chercher les pages web ?

Le navigateur envoie une demande (requête) au serveur web où la page est hébergée, grâce à l'adresse (URL) fournie. Le serveur répond en envoyant le fichier HTML et ses ressources associées (images, CSS, scripts), que le navigateur affiche.

6. C'est quoi un serveur Web ?

Un serveur web est un ordinateur (ou programme) qui stocke les fichiers des sites web et les transmet aux navigateurs des utilisateurs lorsqu'une requête est effectuée. Il agit comme un intermédiaire entre le site et les utilisateurs.

7. Est-ce qu'on a besoin d'utiliser un serveur web local pour accéder à une page web enregistrée localement ?

Non, ce n'est pas obligatoire. Vous pouvez ouvrir un fichier HTML directement dans un navigateur sans passer par un serveur local. Cependant, un serveur web local est nécessaire pour tester certaines fonctionnalités avancées, comme les scripts ou les requêtes réseau.

8. Comment faire pour publier notre site web et le rendre accessible à tout le monde ?

Pour publier un site web :

- **Achetez un nom de domaine** (ex. : www.mon-site.com).
 - **Choisissez un hébergeur web** (ex. : OVH, Bluehost, etc.).
 - **Transférez vos fichiers du site** sur le serveur de l'hébergeur à l'aide d'un outil comme FTP.
 - Votre site sera accessible en ligne via son URL.
-

9. Que signifie le mot HTTP retrouvé dans une adresse d'une page web ?

HTTP (HyperText Transfer Protocol) est un protocole utilisé pour la communication entre les navigateurs et les serveurs web. Il définit comment les informations sont transmises sur le Web. Son successeur sécurisé est HTTPS, qui ajoute une couche de chiffrement pour protéger les données.

10. Concepts importants en programmation

Voici les concepts essentiels à maîtriser pour devenir un bon programmeur :

a. Structure et logique

- **Les algorithmes** : Savoir décomposer un problème en étapes claires et logiques.
- **Les structures de contrôle** : Comprendre les boucles (**for**, **while**) et les conditions (**if**, **else**).
- **Les structures de données** : Connaître les tableaux, listes, dictionnaires, etc.

b. Lisibilité du code

- **Clarté et commentaires** : Écrire du code propre avec des noms de variables explicites et des commentaires.
- **Indentation** : Suivre une structure uniforme pour rendre le code lisible.

c. Réutilisation et modularité

- **Fonctions et méthodes** : Réduire la duplication en créant des blocs de code réutilisables.
- **DRY (Don't Repeat Yourself)** : Éviter les répétitions inutiles dans le code.

d. Débogage et tests

- **Détection des erreurs** : Savoir utiliser les outils de débogage.
- **Tests unitaires** : Tester les parties critiques du code pour prévenir les bugs.

e. Optimisation

- **Performance** : Éviter des solutions inefficaces ou lentes.
- **Gestion de la mémoire** : Comprendre comment gérer les ressources efficacement.

f. Versionnement

- **Utilisation d'outils comme Git** : Suivre et gérer les versions du code, surtout en équipe.
-

11. Peut-on utiliser MS Word pour éditer des pages HTML ?

Techniquement, oui, mais ce n'est pas recommandé. MS Word n'est pas conçu pour coder et risque d'ajouter des éléments indésirables ou non standard dans votre code (par ex. balises superflues ou styles inadaptés).

Pour éditer une page HTML efficacement :

- Utilisez un éditeur de texte simple comme **Notepad** (Windows) ou **TextEdit** (Mac).
 - Pour une meilleure expérience, privilégiez des éditeurs spécialisés comme **Visual Studio Code**, **Sublime Text**, ou **Atom**.
-

12. Caractéristiques importantes dans un éditeur comme Visual Studio Code

VS Code est un éditeur de code très populaire. Voici les fonctionnalités clés qui le rendent puissant :

a. Interface utilisateur personnalisable

- Support des **thèmes** (mode clair/sombre).
- **Disposition flexible** : Disposez les panneaux selon vos besoins.

b. Syntaxe et auto-complétion

- Mise en évidence de la **syntaxe** pour de nombreux langages.
- **Suggestions automatiques** pour accélérer le codage.

c. Extensions et plugins

- Supporte des milliers d'extensions pour ajouter des fonctionnalités (ex. : Prettier, ESLint).
- Intégration avec des outils de versionnement comme **Git**.

d. Débogage intégré

- Permet de tester et de corriger le code directement dans l'éditeur.

e. Terminal intégré

- Vous pouvez exécuter des commandes sans quitter l'environnement.

f. Gestion de projets

- Facilité pour ouvrir et naviguer dans les dossiers de projet.
- **Explorateur de fichiers** pour une gestion rapide.

g. Performances légères

- VS Code est rapide, même avec des projets complexes, tout en étant extensible.

h. Support multi-plateforme

- Disponible sur **Windows**, **Mac**, et **Linux**.

Si vous travaillez sur HTML/CSS/JavaScript, VS Code est l'un des meilleurs choix pour booster votre productivité ! 😊

13. Significations des termes en HTML

- **Texte** : Le contenu visible par l'utilisateur dans une page web. En HTML, le texte est souvent encadré par des balises pour définir son rôle ou style (par exemple, titre, paragraphe).
- **Balise** : Un composant utilisé pour définir le type d'information ou sa structure (ex. : `<p>` pour un paragraphe).
- **Référence** : Un lien ou un chemin vers une ressource externe ou interne, comme une image, une feuille CSS ou un autre document HTML.

14. C'est quoi une balise ? C'est quoi un élément ?

- **Balise** : Une instruction qui indique au navigateur comment traiter une partie du contenu. Elle est délimitée par `<` et `>`. Exemple : `<h1>`.
- **Élément** : La combinaison d'une balise ouvrante, d'une balise fermante, et de son contenu. Exemple :

```
<h1>Mon Titre</h1>
```

Ici, `<h1>` est la balise, et tout `(<h1>Mon Titre</h1>)` constitue l'élément.

15. Comment insérer un saut de ligne en HTML ?

Utilisez la balise `
` :

```
Voici un texte avec un<br>saut de ligne.
```

16. Comment insérer une image ?

Utilisez la balise `` avec les attributs `src` et `alt` :

```

```

- `src` : Spécifie l'emplacement de l'image.
- `alt` : Fournit une description alternative pour les lecteurs d'écran ou si l'image ne se charge pas.

17. Comment appelle-t-on les éléments sans balise fermante ?

Ces éléments sont appelés **éléments auto-fermants** ou **éléments vides**. Exemples : `
`, ``, `<hr>`.

18. Comment insérer un paragraphe en HTML ?

Utilisez la balise `<p>` :

```
<p>Ceci est un paragraphe.</p>
```

19. Différence entre `<p>...</p>` et `
`

- `<p>` : Utilisé pour structurer un paragraphe entier. Il crée automatiquement un espace avant et après le texte.

```
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
```

- `
` : Utilisé pour insérer un saut de ligne **à l'intérieur d'un paragraphe** ou d'un autre élément.

```
Ligne 1<br>Ligne 2
```

20. Signification des balises : `mark`, `sub`, et `sup`

- `<mark>` : Met en surbrillance un texte (comme un stabilo).

```
<p>Ce mot est <mark>important</mark>.</p>
```

- `<sub>` : Affiche un texte en indice (plus bas que le texte normal).

```
H<sub>2</sub>O
```

- `<sup>` : Affiche un texte en exposant (plus haut que le texte normal).

```
2<sup>nd</sup>
```

21. Comment ajouter un titre principal à un document HTML ?

Utilisez la balise `<h1>` pour le titre principal :

```
<h1>Bienvenue sur mon site</h1>
```

22. Combien y a-t-il de niveaux de titres en HTML ?

Il y a **6 niveaux de titres** en HTML :

- `<h1>` (le plus important)
- `<h2>`
- `<h3>`
- `<h4>`
- `<h5>`
- `<h6>` (le moins important)

Exemple :

```
<h1>Titre principal</h1>
<h2>Sous-titre</h2>
<h3>Sous-sous-titre</h3>
```

23. Différence entre éléments de bloc et éléments en ligne

a. Éléments de bloc

- **Comportement** : Occupent toute la largeur disponible (nouvelle ligne automatiquement).
- **Rôle** : Structurent le contenu.
- **Exemples** : `<div>`, `<p>`, `<h1>` à `<h6>`, ``, ``, `<table>`.

b. Éléments en ligne

- **Comportement** : Ne prennent que la largeur nécessaire et s'alignent avec d'autres éléments.
- **Rôle** : Agissent principalement à l'intérieur des éléments de bloc.
- **Exemples** : ``, `<a>`, ``, ``, ``.

24. Différence entre une liste ordonnée et une liste non ordonnée

a. Liste ordonnée (``)

- **Caractéristique** : Les éléments sont numérotés ou classés (1, 2, 3... ou i, ii, iii...).
- **Usage** : Utilisée pour représenter un ordre ou une hiérarchie.

h4

```
<ol>
  <li>Première étape</li>
  <li>Deuxième étape</li>
</ol>
```

b. Liste non ordonnée (``)

- **Caractéristique** : Les éléments sont marqués avec des puces ou des symboles.
- **Usage** : Utilisée pour des listes sans importance d'ordre.

h4

```
<ul>
  <li>Élément A</li>
  <li>Élément B</li>
</ul>
```

25. Balises utilisées pour chaque type de liste

- Liste ordonnée : `` (Ordered List).
- Liste non ordonnée : `` (Unordered List).
- Les deux types utilisent `` (List Item) pour ajouter des éléments.

26. Comment ajouter des éléments dans une liste ?

Chaque élément de liste est défini par une balise `` :

h4

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

Pour une liste ordonnée :

h4

```
<ol>
  <li>Premier</li>
  <li>Deuxième</li>
</ol>
```

27. Abréviation d'une adresse Web

L'abréviation courante pour une adresse Web est **URL** (*Uniform Resource Locator*).

28. Signification des parties d'une URL

Exemple d'URL :

`https://www.example.com/page.html?query=123#section`

- **https://** : Protocole (HyperText Transfer Protocol Secure).
- **www** : Sous-domaine (optionnel).
- **example.com** : Nom de domaine principal.
- **/page.html** : Chemin vers une ressource spécifique sur le serveur.
- **?query=123** : Paramètres de requête (optionnels).
- **#section** : Ancre ou fragment pointant vers une partie précise de la page (optionnel).

29. Protocole utilisé pour les pages enregistrées localement

Le protocole utilisé pour accéder aux pages enregistrées localement est **file://**.

Exemple :

```
file:///C:/Users/VotreNom/Documents/page.html
```

Cela permet au navigateur d'afficher les fichiers directement à partir du système de fichiers local.

30. Comment créer un lien hypertexte ?

Pour créer un lien hypertexte, utilisez la balise `<a>` avec l'attribut **href** :

```
<a href="https://www.example.com">Cliquez ici</a>
```

- **href** : Indique l'adresse de la ressource (page web, fichier, ou autre).
- **Texte entre les balises** : Ce sera le texte cliquable visible pour l'utilisateur.

31. Comment insérer une image ? Quelle est la signification de ses attributs ?

Utilisez la balise `` :

```

```

Signification des attributs :

- **src** (source) : Chemin ou URL de l'image.
- **alt** (texte alternatif) : Description affichée si l'image ne peut pas être chargée (important pour l'accessibilité et le SEO).
- **width et height** : Dimensions de l'image (en pixels ou en pourcentage).
- **Autres attributs possibles** :
 - **title** : Infobulle affichée lorsque l'on survole l'image.
 - **loading** : Contrôle le chargement différé (**lazy** pour retarder le chargement).

32. Différence entre une adresse relative et une adresse absolue

a. Adresse absolue

- Indique le chemin complet vers une ressource.
- Utilisée pour des ressources externes ou sur des serveurs distants.
- Exemple :

```
<a href="https://www.example.com/page.html">Lien absolu</a>
```

b. Adresse relative

- Indique un chemin par rapport à l'emplacement actuel du fichier HTML.
- Utilisée pour des fichiers locaux ou dans la même structure de site.
- Exemple :

```
<a href="page.html">Lien relatif</a>
```

Différence clé :

- **Adresse absolue** : Toujours accessible quel que soit l'emplacement du fichier HTML.
- **Adresse relative** : Dépend de l'emplacement du fichier HTML.

33. Structure minimale d'un document HTML

Voici la structure minimale et l'explication de ses parties :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre du document</title>
  </head>
  <body>
    <p>Contenu visible de la page.</p>
  </body>
</html>
```

Explications :

1. `<!DOCTYPE html>` :

- Déclare que le document utilise HTML5.
- Aide les navigateurs à interpréter correctement le code.

2. `<html>` :

- Balise racine englobant tout le document HTML.

3. `<head>` :

- Contient les informations **méta** sur le document (non visibles par l'utilisateur).
- Exemples : titre, styles, encodage, scripts.

4. `<title>` :

- Définit le titre affiché dans l'onglet du navigateur.

5. `<body>` :

- Contient tout le contenu visible par l'utilisateur (texte, images, liens, etc.).
-