

Programmation Web

HTML



TP 2: HTML





Spécifications

- L'objectif de ce TP est de vous familiariser avec les principales balises HTML vues en cours.
- Vous allez mettre en pratique ces notions à travers la création d'une page Web complète, dont le contenu portera sur :
« **Histoire du Web — Brève Chronologie et perspectives** »
- Le fichier **TP2_HTML.pdf** est votre guide de réalisation : lisez-le attentivement avant de commencer.
- Le fichier **Texte.pdf**, comporte le contenu textuel à insérer dans votre page Web, ainsi que des annotations précisant les balises HTML à utiliser ainsi que certaines valeurs d'attributs attendues.
- Le dossier **ressources** contient les fichiers multimédias à mettre sur la page.
- Ce TP commencera par un bref guide d'initialisation de votre espace Github.
- L'énoncé du TP est déposé sur Github. Vous devez donc créer un compte, et demander à ce qu'on vous ajoute au classroom (Ces étapes sont détaillées dans les slides suivants). Le lien vers l'Assignment vous est également fourni dans les slides suivants.
- Une fois que vous avez accès au repo de votre Assignment, cloner-le en local, et commencer à travailler.
- A la fin de votre travail, vous devrez soumettre votre réalisation en faisant un commit/push à votre repo (tout est détaillé dans les slides).
- **Livrables attendus:** 8 fichiers HTML: chaque fichier correspond à une étape (ex. step1.html...jusqu'à...step1.html).



Github

À vous

Création d'un compte Github

1

Aller à la page <https://github.com/login>

2

Choisir: Créer un compte, ou bien (se connecter avec son compte Google)

3

Vous pouvez vous connecter directement si vous avez déjà un compte

Sign in to GitHub

Username or email address

Password [Forgot password?](#)

Sign in

or

Continue with Google

Continue with Apple

New to GitHub? [Create an account](#)

[Sign in with a passkey](#)

Github

À vous **Remplir Formulaire**

1 Maintenant, il faudra indiquer à l'enseignant l'identifiant de votre compte GitHub, en remplissant le formulaire suivant:

2 [Cliquer ici pour vous rendre sur le form](#)


Comptes GitHub

* Indique une question obligatoire

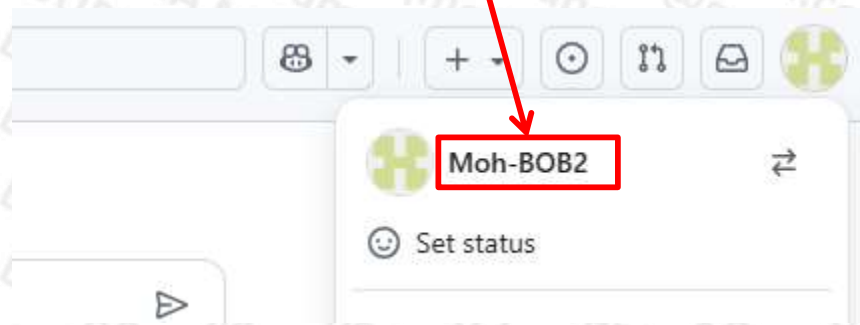
E-mail *

☐ Enregistrer mo.boubenia@gmail.com comme adresse e-mail à joindre à ma réponse

Identifiant GitHub *

Votre réponse 

3 Votre ID github se trouve ici (en cliquant en haut à droite sur l'interface de github):



4 Coller votre ID dans le formulaire



Github

À vous Accepter l'Assignement du TP

1

Pour avoir accès à l'espace TP2-HTML sur Github, rendez vous à la page suivante: https://classroom.github.com/a/1v_jY9vD

2

La première fois, GitHub classroom vous demandera une vérification pour pouvoir rejoindre le classroom. Cliquer simplement sur votre nom dans la liste):

Accept the assignment — TP2-HTML

Once you accept this assignment, you will be granted access to the `html-tp-Moh-BOB2` repository in the `usthb-boubenia` organization on GitHub.

4

Accepter l'assignement

Accept this assignment

Join the classroom:

To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email).

3

si vous recevez une erreur, il suffit simplement d'actualiser la page

Identifiers

Moh-BOB2

5

Si tout se passe bien, vous verrez cet écran.

Cliquer ici pour consulter le repo du TP



You're ready to go!

You accepted the assignment, TP2-HTML

Your assignment repository has been created:

<https://github.com/usthb-boubenia/html-tp-Moh-BOB2>

We've configured the repository associated with this assignment.



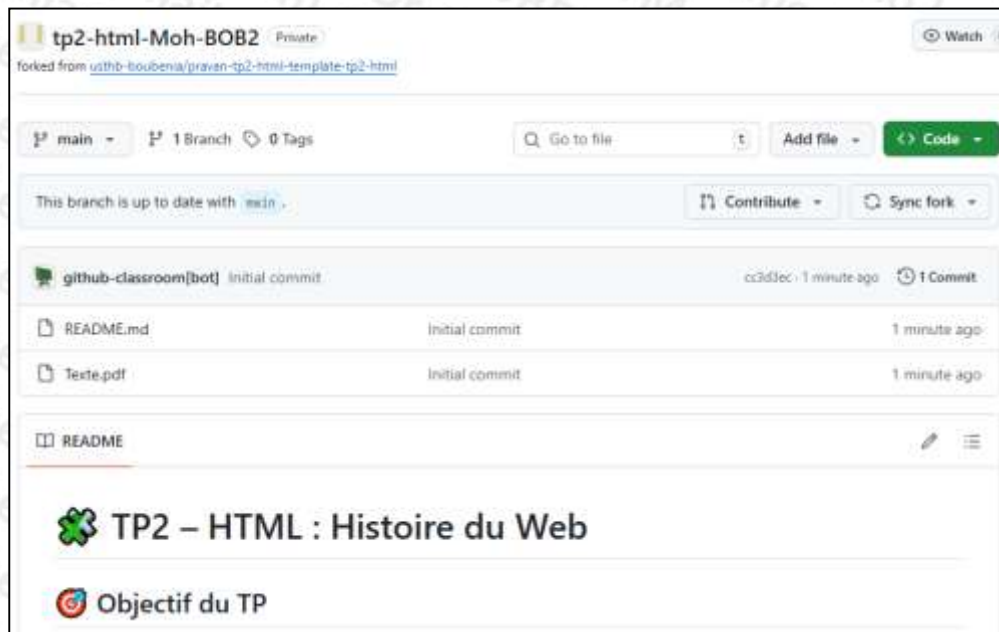
Github

À vous

Votre Repo individuel du TP

1

Votre repo individuel s'affichera comme suit





Github

À vous

Téléchargement/Installation de GitBash

1 Aller à la page suivante pour télécharger GitBash:
<https://git-scm.com/downloads/win>

2

Download for Windows

Click here to download the latest (2.51.0(2)) x64 version of Git the most recent maintained build. It was released 12 days ago, on

Other Git for Windows downloads

Standalone Installer

Git for Windows/x64 Setup.

Git for Windows/ARM64 Setup.

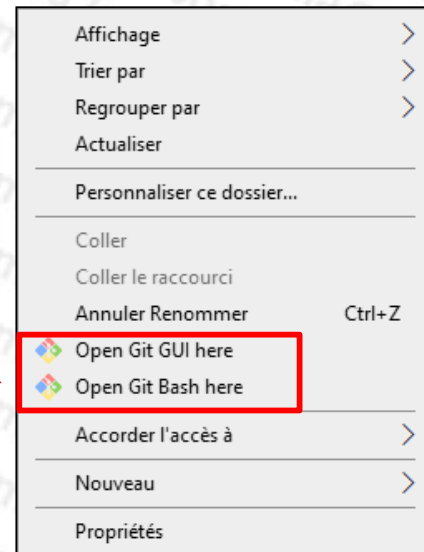
Portable ("thumbdrive edition")

Git for Windows/x64 Portable.

Git for Windows/ARM64 Portable.

4

Si l'installation se déroule bien, vous devriez voir apparaître le menu Git lorsque vous faites cliquer droit sur n'importe quel répertoire



3

Une fois téléchargé, procéder à l'installation, en faisant suivre sans modifier les paramètres par défaut (accepter seulement qu'il le rajoute dans le menu contextuel de Windows)



Github

À vous

Création d'un clé SSH

1 Créer un dossier n'importe où sur votre machine, puis entrer dans le répertoire et faites cliquer doit/Open Git Bash Here

3 Commencer par créer une clé ssh:

```
ssh-keygen -t ed25519 -C "user@gmail.com" -f ~/.ssh/id_ssh_github1
```

Votre email

4 Appuyez simplement sur entrer dans la question qu'on vous posera

5 Si tout se passe bien, vous verrez apparaître votre clé privé/public dans le répertoire C:\Windows\user\<votre_nom>\.ssh

2

Une console Git va s'ouvrir.

```
MINGW64:/c/Users/BOB/Desktop/Pravan_Cours/TP2-HTML/repo-git/templ...
BOB@DESKTOP-NR7V098 MINGW64 ~/Desktop/Pravan_Cours/TP2-HTML/repo-g
2-html (main)
$ |
```

id_ssh_github1
id_ssh_github1.pub



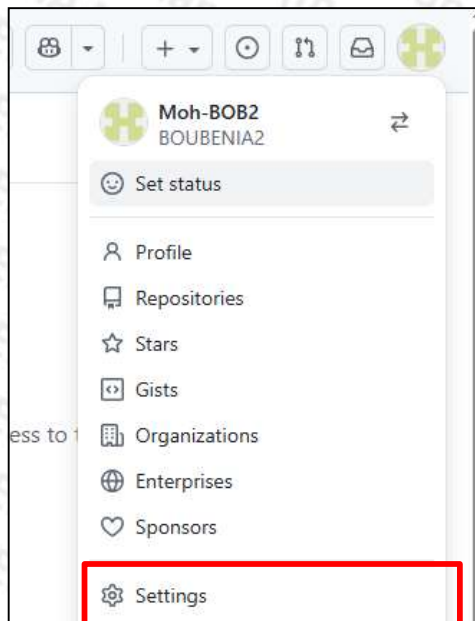
Github

À vous

SSH Github

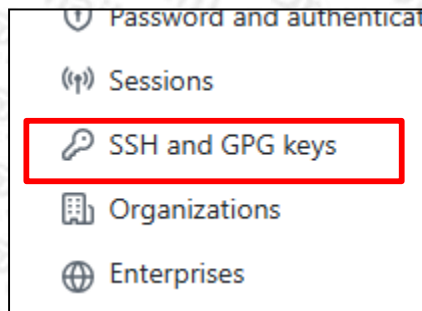
1

Il faudra maintenant indiquer à Github votre clé SSH. Pour cela, Aller sur Settings



2

Sur le menu de gauche, cliquer sur SSH, puis New SSH Key



New SSH key

3

Donner un nom à votre clé, puis coller votre clé public. Puis cliquer sur Add SSH Key

4

Le contenu de votre clé public se trouve dans le fichier: id_ssh_github1.pub

Title

MySSHkey

Key type

Authentication Key

Key

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AA



Github

À vous

Création d'une configuration SSH

1

Aller dans votre répertoire:
C:\Windows\user\<votre_nom>\.ssh
Créer le fichier config :

config
id_ssh_github1
id_ssh_github1.pub

2

Mettez dedans la configuration suivante et sauvegarder

```
# === Compte perso (student) ===
Host github.com-student
  HostName github.com
  User git
  IdentityFile ~/.ssh/id_ssh_github1
  IdentitiesOnly yes
```

3

Pour tester la clé, revenez à la console de Gitbash, et taper:

```
ssh -T github.com-student
```

Taper oui à la question que github vous posera

```
$ ssh -T github.com-student
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256: [redacted]
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi Moh-B0B2! You've successfully authenticated, but GitHub does not provide shell
access.
```

Si tout se passe bien, vous recevrez un message du genre : Hi...



Github

À vous

Cloner le repo du TP

1

Vous allez maintenant pouvoir cloner votre repo avec votre clé SSH.

2

Commencez d'abord par copier l'URL de votre repo.
(Si vous avez du mal à le retrouver, aller dans le menu->organisation->usthb-boubenie) et vous allez pouvoir le localiser.

3

Il faut récupérer le lien SSH comme suit:

4

Aller dans la console git bash et taper:

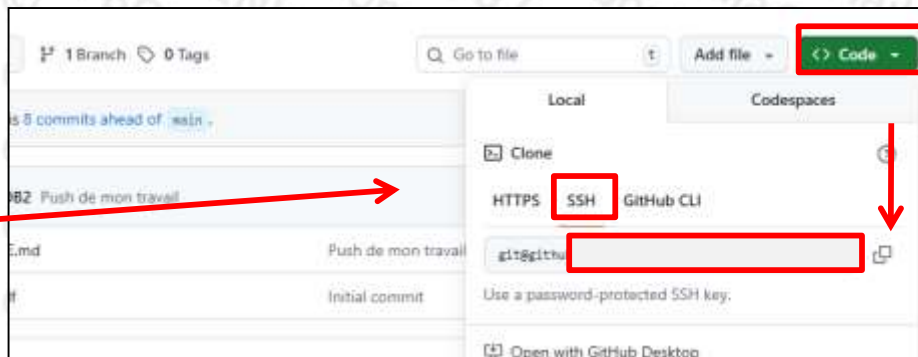
```
git clone git@github.com:student:usthb-boubenia/<le nom de votre repo>.git
```

```
git config user.email "votre-email"
git config user.name "id_GitHub"
```

Email et username doivent correspondre exactement à ceux liés à votre compte Github

```
MINGW64:/c/Users/BOB/Desktop/Pravan_Cours/TP2-HTML/repo-git/templ...

BOB@DESKTOP-NR7V098 MINGW64 ~/Desktop/Pravan_Cours/TP2-HTML/repo-g
2-html (main)
$ |
```





Github

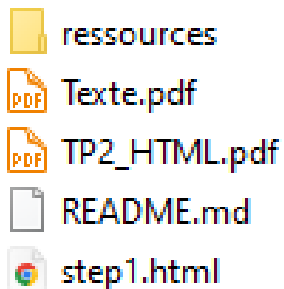
Tout est bon !

1

Votre espace est désormais prêt, vous pouvez commencer à travailler en consultant les slides suivants:

2

C'est ici que vous allez créer votre premier fichier



- ressources
- Texte.pdf
- TP2_HTML.pdf
- README.md
- step1.html





Etape 0: Test d'un push

À vous

HTML

1

Créer une page web step1.html, dont le contenu initial est comme suit:

2

Procéder à l'indentation du code saisi

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
</head>
<body>
</body>
</html>
```



Etape 0: Test d'un push

À vous Votre premier commit/push

1

Ouvrez Git Bash dans le dossier, et taper les commande suivante:

```
git add .
```

```
git commit -m "Push de mon travail"
```

```
git push origin main
```

```
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 386 bytes | 193.00 KiB, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:student:usthb-boubenia/tp2-html-Moh-BOB2
83d7e81..d43328c main -> main
```

2

Revenez à l'interface graphique de github, sur votre repo, vous devriez voir apparaitre vos changements

```
MINGW64:/c/Users/BOB/Desktop/Pravan_Cours/TP2-HTML/repo-git/templ...

BOB@DESKTOP-NR7V098 MINGW64 ~/Desktop/Pravan_Cours/TP2-HTML/repo-g
2-html (main)
$ |
```



Moh-BOB2 Push de mon travail

README.md

Push de mon travail

Texte.pdf

Initial commit

step1.html

Push de mon travail





Etape 1: Structure & Titre de la page

À vous

1

Donner le titre suivant à la page:

Histoire du Web — Brève Chronologie et perspectives

2

Le document doit avoir la structure suivante (sans mettre de bordure).
Respecter le nom des id.

Choisir les balises sémantiques appropriées:

Footer

Section

Header

Nav

Main

3

Séparateurs de thématique

4

Les valeurs Id1, id2, id3...etc sont détaillées à la fin du TP

Histoire du Web — Brève Chronologie et perspectives

Entête

Menu (id1=voir tableau)

Corps

Thématique 1 (id2= voir tableau)

Thématique 2 (id3= voir tableau)

Thématique 3 (id4= voir tableau)

Thématique 4 (id5= voir tableau)

Pied de page



Etape 1: Structure & Titre de la page

À vous

Clé	valeur
id1	menu
id2	precursors
id3	creation
Id4	semantic
id5	resume



Etape 2: Partie Entête

À vous

1

Copier le contenu de step1.html dans un nouveau fichier nommé step2.html

2

L'entête doit avoir la structure sémantique suivante:

Titre de
niveau 1

Paragraphe

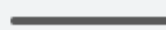
Contenu sonore
(music.mp3), de type
audio/mpeg

Histoire du Web — Chronologie détaillée et perspectives

Document pédagogique long, détaillé et référencé présentant l'histoire conceptuelle et technique du World Wide Web (تسبقة الويب العالمية), depuis les précurseurs intellectuels jusqu'aux projets de décentralisation et au Web sémantique. Il contient des contextes, des descriptions techniques, des acteurs clefs et des pistes pour l'avenir.



0:00 / 3:44



Voir le fichier PDF
pour les détails



Etape 3: Menu

À vous

1 Copier le contenu de step2.html dans un nouveau fichier nommé step3.html

2 Remplir la partie menu comme indiqué ci-contre

3 Liste ordonnée, dont chaque élément un signets (lien interne) renvoyant vers la thématique correspondante

1. [Précurseurs et idées fondatrices](#)
2. [Naissance du Web \(Berners-Lee\) et premières implémentations](#)
3. [Le Web sémantique, Linked Data et Solid](#)
4. [Résumé & dates clés](#)

Thématique 1

Thématique 2

Thématique 3

Thématique 4

Etape 4: Thématique 1

À vous

1 Copier le contenu de step3.html dans un nouveau fichier nommé step4.html

2 Titre de niveau 2

6 Titre de niveau 3

3 Un article (<article>)

8 Une figure qui contient une image, une vidéo et une légende. La légende contient un lien hypertexte. (voir les annotations sur le fichier PDF pour plus de détails)

4 Un article (<article>)

7 Titre de niveau 3

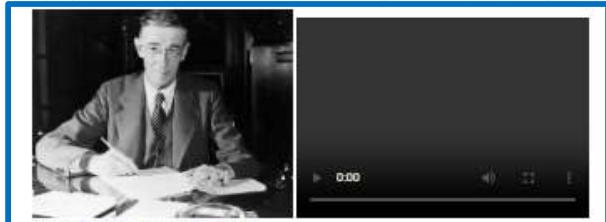
9 Une figure qui contient 2 images et une légende. La légende contient un lien hypertexte. (voir les annotations sur le fichier PDF pour plus de détails)

5 Un lien interne qui permet de remonter au menu

1. Précurseurs et idées fondatrices

Vannevar Bush (1945) — *As We May Think* et le Memex

En 1945, Vannevar Bush publie l'article *As We May Think* dans lequel il décrit le **Memex**, un dispositif imaginaire destiné à stocker, lier et retrouver rapidement des documents et des notes.



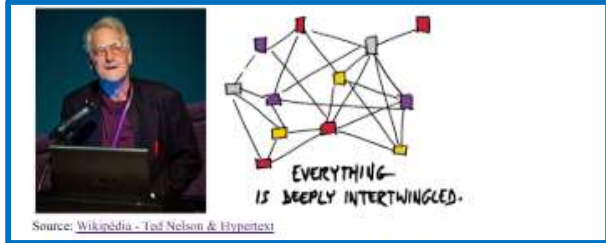
Source: [Wikipédia - Vannevar & Memex](#)

Même si le Memex était davantage un appareil mécanique et optique que numérique, l'idée de lier l'information a profondément influencé la recherche en informatique. Les notions de stockage associatif et de navigation non-linéaire y sont des exemples.

Importances : le **Memex** a servi de référence conceptuelle pour les idées d'hypertexte et de recherche.

Ted Nelson — l'inventeur du terme "hypertexte" et le projet Xanadu

Dans les années 1960, Ted Nelson formalise le terme **hypertexte** et propose le projet ambitieux **Xanadu**, qui vise à créer un système mondial de documents interconnectés, avec des liens bidirectionnels.



Source: [Wikipédia - Ted Nelson & Hypertext](#)

Le projet Xanadu n'a jamais atteint une implémentation mondiale, mais il définit des idées fortes : *liens permanents, traçabilité des citations, granularité fine des références*. Ces concepts re

[Revenir au menu](#)

Voir le fichier PDF pour les détails

Etape 5: Thématique 2

À vous

1 Copier le contenu de step4.html dans un nouveau fichier nommé step5.html

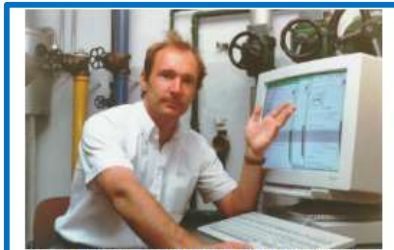
2 Titre

2. Naissance du Web — Tim Berners-Lee et premières im

7 Titre

1989–1991 — Tim Berners-Lee au CERN : HTML, HTTP, URL

Tim Berners-Lee propose, au CERN, un système pour partager des documents entre chercheurs en utilisant l'hypertexte et Internet



8 Une figure qui contient une image et une légende. (voir les annotations sur le fichier PDF pour plus de détails)

3 Un article

9 Une liste non ordonnée

Source: [History — WorldWideWeb NeXT Application](#)

HTML (HyperText Markup Language) — format pour structurer documents et liens.
HTTP (HyperText Transfer Protocol) — protocole d'échange client/serveur.
URI/URL — mécanisme d'adressage des ressources sur le réseau.

Exemple d'un code HTML: <p> Hello world </p>

En 1991 la première page Web est mise en ligne. Berners-Lee publie des logiciels de serveur et de client

8 Titre

4 Un article

1993–1994 — Premiers navigateurs graphiques : Mosaic et Netscape

Le navigateur Mosaic (1993) popularise l'affichage graphique d'images et de textes mêlés ; il rend l'expérience Web utilisable par

Conséquences :

l'accessibilité et la convivialité des navigateurs déclenchent une croissance exponentielle du

9 Titre

5 Un article

1994–1996 — Standardisation et premiers moteurs de recherche

À mesure que le Web se répand, la nécessité de standards devient impérative. Le W3C est fondé (1994) par Berners-Lee pour prom

Les premières problématiques émergent : indexation, performance, sécurité, et modélisation des données sur le Web.

[Retour au menu](#)

6 Un lien pour remonter

Voir le fichier PDF pour les détails

1 Copier le contenu de step5.html dans un nouveau fichier nommé step6.html

Cette fois, à vous de déduire les balises à mettre

3. Le Web sémantique, Linked Data et Solid

Le Web sémantique : objectifs et composants

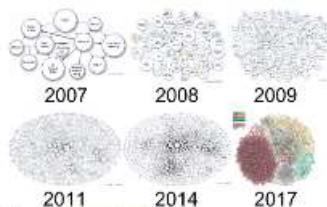
Le Web sémantique vise à ajouter une couche de métadonnées et de signification aux contenus Web afin que les machines puissent en faire usage.

- **RDF** (Resource Description Framework) : modèle de triplets pour représenter les données.
- **OWL** : ontologies pour définir des vocabulaires et relations complexes.
- **SPARQL** : langage de requête pour interroger des graphes RDF.
- **Microformats** et **JSON-LD** : moyens d'encoder des métadonnées dans les pages web.

Le Web sémantique ambitionne d'améliorer l'interopérabilité des données et d'alimenter des agents intelligents capables de combiner ces données.

Linked Data

Tim Berners-Lee a proposé des principes pour publier des données liées (Linked Data) : utiliser des URIs, fournir des descriptions, lier les données.



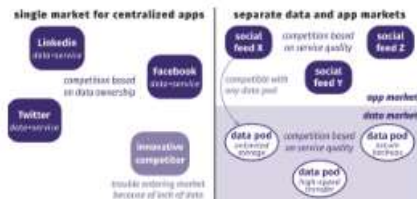
Source: [The Linked Data Wiki](#)

Solid (Social Linked Data)

Tim Berners-Lee:

«The Web as I envisaged it, we have not seen it yet.»

Solid est une initiative de Tim Berners-Lee visant à redonner le contrôle des données personnelles aux utilisateurs. L'idée : chaque utilisateur possède ses données.



Source: [Solid: Linked Data for personal data management](#)

Avantages : autonomie des utilisateurs, séparation des données et des services, interopérabilité des données via RDF/JSON-LD, Limitation de la collecte de données.

Ruben Verborgh:

Three paradigm shifts we need to prepare for:
- End users become data controllers.
- Apps become views.
- Interfaces become queries.

Cas d'usage du Web sémantique

Exemples : moteurs de recherche enrichis, assistants vocaux, intégration de données scientifiques, catalogues de bibliothèques, et agents intelligents.

[Revenir au menu](#)



Voir le fichier PDF pour les détails



Etape 7: Thématique 4

À vous

- 1 Copier le contenu de step6.html dans un nouveau fichier nommé step7.html

2

...

4. Résumé & dates clés

Date Invention

1945 Memex

1960 Hypertexte

1989 Web

[Retour au menu](#)

3

Un tableau
court

Voir le fichier PDF
pour les détails



Etape 8: Pieds de page

À vous

1

Copier le contenu de step7.html dans un nouveau fichier nommé step8.html

Document généré le 2025-10-09 10:38 UTC. © 2025 — Support pédagogique.



Validation W3C

À vous

1

Copier votre code HTML et validez-le sur le validateur W3C: <https://validator.w3.org/nu/#textarea>

W3C

2

Coller votre code ici:

Checker Input

Show ☒ source ☐ outline ☐ image report

Check by ☐ css

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Histoire du Web – Brève Chronologie et perspe
</head>
<body>
  <header>
</body>
</html>
```

3

Check

Message Filtering

- Error** End tag for `body` seen, but there were unclosed elements.
 From line 9, column 1; to line 9, column 7

```
header> </body></html>
```
- Error** Unclosed element `header`.
 From line 8, column 3; to line 8, column 10

```
<body> <header> </>
```

4

Vérifiez s'il y a des erreurs et corrigez les



Envoyer votre travail

À vous

1

Lorsque vous pensez avoir tout fini, soumettez votre travail dans votre repo:

```
git add .
```

```
git commit -m "Travail Terminé"
```

```
git push origin main
```

HTML



TP 2: HTML

1. Création d'un document HTML valide.
2. Familiarisation avec les balises HTML
3. Familiarisation avec HTML5
4. Familiarisation avec Github
5. Validation W3C d'un document HTML