

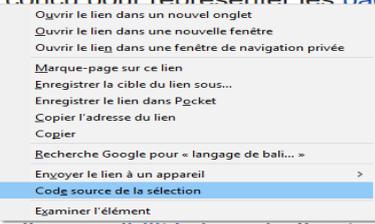
Qu'est-ce que le Web ? Qu'est-ce qui le caractérise le plus ?

Nous avons vu qu'il s'agissait de la notion de documents **hypertextes** accessibles via des liens (ou **hyperliens**). Nous allons voir maintenant comment on parvient à créer ces liens. Et nous allons devoir utiliser les **URLs** ... Cela nous permettra également de montrer pourquoi il faut être vigilant lorsqu'on reçoit des liens par email.

A – Gérer les liens dans un document HTML

01 – Trouver la page wikipédia présentant le html. Sélectionner avec la souris uniquement le texte **langage de balisage** .

L'**HyperText Markup Language**, généralement abrégé **HTML**, est le **langage de balisage** conçu pour représenter les pages web. C'est un langage personnel. HTML permet également de structurer sémantiquement et logiquement le contenu des pages, d'inclure des images, des formulaires informatiques. Il permet de rendre les pages **interopérables** avec des équipements conformes aux exigences de l'**accessibilité du web**. Il est



02 – En utilisant un clic droit et en sélectionnant **Code Source de la sélection**, vous devriez obtenir ceci :

```
<a href="/wiki/Langage_de_balisage" title="Langage de balisage">langage de balisage</a>
```

On voit clairement que le code ne comporte pas uniquement le texte sélectionné (**langage de balisage**) mais également d'autres indications. Le navigateur fournit un visuel en fonction de ce code source qu'il interprète : c'est un bout de ce code source que vous avez sous les yeux.

On voit qu'on parvient à séparer

- le texte (**langage de balisage**) à afficher de
- l'utilité qu'on lui donne (un lien vers /wiki/Langage_de_balisage)

Le HTML est un langage descriptif : c'est le langage qu'on doit utiliser pour faire comprendre au navigateur ce qu'on veut précisément qu'il fasse. On sépare les éléments à afficher du sens qu'on veut leur donner en utilisant des **balises** (tag en anglais). Les **balises** sont en rouge dans ce document.

On entoure le texte :

- d'une **balise ouvrante** (**<a**) puis
- de la **balise fermante** correspondante qui comporte un / en plus (**</a**) .

On utilise les chevrons < et > pour indiquer au navigateur qu'il a affaire à une balise.

```
<a>langage de balisage</a>
```

03 – Quel est le nom de la balise qui sert visiblement à créer le lien ?

La **balise d'ouverture** peut contenir des éléments descriptifs supplémentaires.

Dans le cas des balises **a**, on peut fournir la destination du lien à l'aide de

- l'attribut **href** suivi de
- = suivi de
- l'URL de la destination entre deux guillemets

```
<a href="/wiki/Langage_de_balisage">langage de balisage</a>
```

L'URL est visiblement `/wiki/Langage_de_balisage`

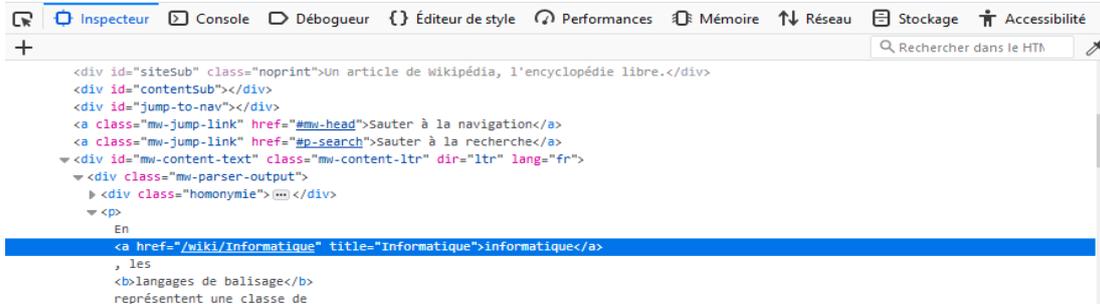
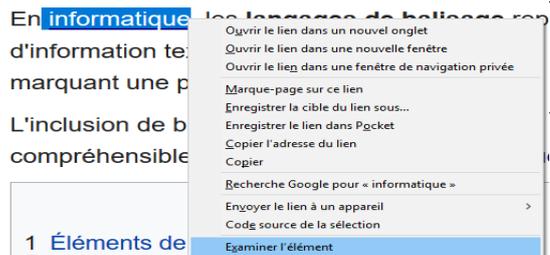
Cette URL est clairement incomplète : il n'y a pas de nom de site !

La présence de `/` indique qu'il s'agit d'une adresse absolue : cela veut dire qu'on donne le chemin à suivre en partant de la source située à gauche.

Or, comme on n'indique pas de site, c'est qu'il s'agit du site qui a fourni la page : fr.wikipedia.org.

04 – Sur quelle page va-t-on aboutir en cliquant sur `langage de balisage` ? Vérifiez en l'utilisant.

05 – Sélectionner le texte d'un des liens. En utilisant un clic droit, sélectionner **Examiner l'Élément** ou **Inspecter**.



06 – Faire un double-clic sur le mot **Informatique** dans l'inspecteur. Modifier l'adresse fournie derrière href. Par exemple, href="/wiki/Chat".

Attention, nous ne venons pas de modifier la page de Wikipédia, ce serait de la dégradation !

La page HTML originelle est toujours fournie à l'identique.

Le code que vous venez de modifier est simplement le code **local** que votre navigateur est en train d'analyser et qui est stocké sur votre ordinateur après son envoi par le serveur de Wikipédia.

Le code que vous voyez apparaître est donc le code html stocké côté client.

07 – Tester le nouveau lien. Peut-on faire une confiance aveugle au lien qu'on trouve sur Internet ? Pourquoi les emails des banques ou autres précisent souvent que les emails officiels ne contiennent jamais de lien ?

Alors comment peut-on vérifier la validité d'un lien ? Le survoler est déjà un bon réflexe : en bas à gauche de la plupart des navigateurs la destination du lien apparaît. Le lien pourrait encore être changé à la volée une fois le clic effectué mais la plupart du temps, les arnaqueurs ne s'embêtent pas à être aussi sournois.



Maintenant que vous savez comment fonctionnent les liens, nous allons créer un ensemble de pages et y créer des liens. Cela vous permettra de créer votre propre banque de liens.

B – Structure d'une page HTML

Une page vide à l'affichage pourrait contenir ceci :

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

  <body>
  </body>
</html>
```

08 – Quelles sont les balises contenues entre la balise `<html>` d'ouverture et la balise de fermeture `</html>` ?

La première ligne `<!DOCTYPE html>` permet de signaler que le document est réalisé en HTML. Il s'agit d'une balise particulière : elle est dite orpheline car elle ne comporte pas de balise de fermeture.

La balise `<html>` contient l'intégralité du document.

La balise `<head>` va contenir des données qu'on veut transmettre au navigateur mais qui ne seront pas affichées sur la page : la langue utilisée, le titre à mettre sur l'onglet ...

La balise `<body>` va contenir ce qu'on désire voir s'afficher sur la page.

Remarque : la notion de balises contenues intégralement dans d'autres balises est extrêmement importante. Pour visualiser facilement ce qui est contenu dans quoi, on utilise le système de tabulation. Ainsi, on voit bien que `<head>` est contenue dans `<html>` car `<head>` est tabulée vers la droite par rapport à `<html>`.

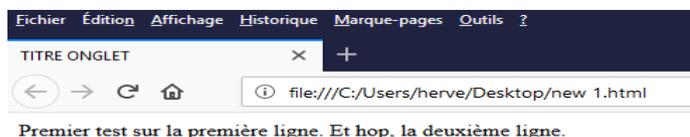
On remarque que la balise de fermeture `</head>` est au même niveau que sa balise d'ouverture.

09 – Utiliser le lien suivant : <https://trinket.io/html/1d129d8199>

ou via une iframe : <https://formation.infoforall.fr/vidé.html>

10 – Copier-coller le code HTML ci-dessous. Lancer le code si l'autorun n'est pas activé.

```
<!DOCTYPE html>
<html>
  <head>
    <title>TITRE ONGLET</title>
  </head>
  <body>
    Premier test sur la première ligne.
    Et hop, la deuxième ligne.
  </body>
</html>
```



11 – Est-ce que le navigateur a bien géré le passage à la ligne ?

Alors, comment passer à la ligne ?

Première méthode : on force le passage à la ligne avec une balise orpheline nommée br. Comme la balise est seule, on l'écrit avec un / à la fin : `
`

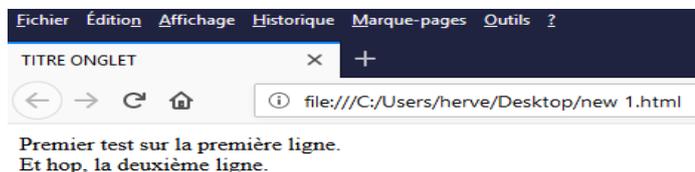
12 – Modifier le code qui utilise une balise orpheline `
`.

```
<!DOCTYPE html>
<html>
  <head>
    <title>TITRE ONGLET</title>
  </head>

  <body>
    Premier test sur la première ligne.<br />
    Et hop, la deuxième ligne.
  </body>
</html>
```

13 – Actualiser la page (pensez bien à sauvegarder la page avant!)

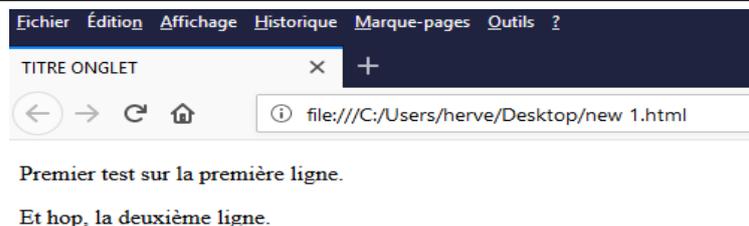
Et voici le résultat :



Deuxième méthode : on peut faire mieux. Le HTML possède une balise qui exprime le fait qu'on désire créer un paragraphe : un texte séparé des autres textes par un espace au dessus et un espace en dessous.

```
<body>
  <p>Premier test sur la première ligne.</p>
  <p>Et hop, la deuxième ligne.</p>
</body>
```

14 – Modifier le code et visualiser le résultat.



Il est temps de créer notre première page de liens. Voici un exemple de page qui fournit des liens vers des sites parlant de HTML. Vous le trouverez dans le deuxième onglet.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML</title>
  </head>

  <body>
    <h1>Trois sites pour apprendre le HTML</h1>

    <p>Numéro 1 : <a
href="https://openclassrooms.com/fr/courses/1603881-apprenez-a-
creer-votre-site-web-avec-html5-et-css3">OpenClassRooms</a></p>

    <p>Numéro 2 : <a
href="https://www.w3schools.com/html/default.asp">w3Schools</a></p>
>

    <p>Numéro 3 : <a
href="https://j-willette.developpez.com/tutoriels/html/les-bases-du-
html/?page=page_1">developpez.com</a></p>

  </body>
</html>
```

Remarque importante : Il est possible que certains liens semblent ne pas vouloir s'afficher en utilisant trinket. Si vous désirez néanmoins vérifier sous trinket que votre lien fonctionne bien, il faut simplement survoler le lien et faire un clic-droit et ouvrir le lien dans un nouvel onglet.

15 – Choisir un thème et réaliser une page html menant vers trois pages intéressantes vis à vis du thème. Vous enregistrerez la page sous le nom `liens_perso.html`

Il serait néanmoins intéressant de relier la page de lien HTML avec votre page de liens. Vous avez maintenant deux fichiers HTML au moins :

- `liens_html.html` et
- `liens_perso.html`.

Et ils sont enregistrés dans le même répertoire ou dans le même dossier trinket.

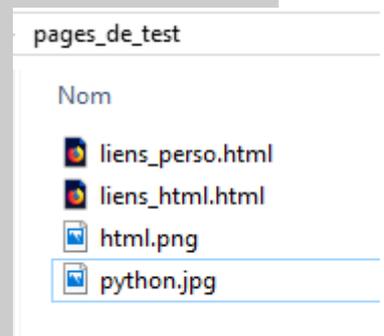
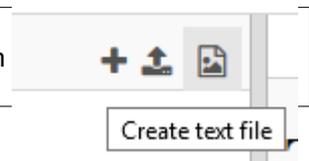
Comment créer un lien d'une page sur l'autre ? En effet, les pages ne sont pas accessibles sur Internet mais uniquement sur votre poste ou votre dossier trinket.

Et bien, il suffit de ne pas fournir l'adresse absolue mais l'**adresse relative** : on donne l'adresse par rapport au répertoire dans lequel se trouve la page actuelle.

Si vous mettez les deux pages dans le même répertoire, elles pourront ainsi se trouver très facilement : il suffit de donner le nom de la page.

Ici les deux pages sont stockées dans un répertoire nommé `pages_de_test` et on y trouve également deux images.

Comment codifier le fait qu'on fournit une **adresse relative** ? C'est simple : il suffit de ne pas placer `/` avant de fournir l'adresse.



Pour aller sur la page des liens perso depuis la page des liens HTML :

```
<p>Liens perso : <a href="liens_perso.html">Perso</a></p>
```

Pour aller sur la page des liens html depuis la page des liens perso :

```
<p>Liens html : <a href="liens_html.html">HTML</a></p>
```

16 – Rajouter les liens internes permettant de passer d'une page locale à l'autre.

C – Rajouter une image dans un document HTML

En 1993, après des années de pages HTML remplies uniquement de texte, une balise est apparue lorsque les connexions furent assez rapides et les écrans capables d'afficher des images : la balise orpheline ``.

On doit fournir à cette balise un attribut source nommée `src` qui contient l'adresse de votre image, absolue ou relative.

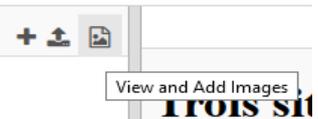
Pour afficher une image en HTML, le plus simple est d'enregistrer l'image dans le même répertoire que le fichier HTML lui-même. On pourra alors très facilement la localiser avec l'adressage relatif.

Exemple avec les images qui sont dans votre répertoire trinket juste au dessus :

```


```

17 – Utiliser les lignes fournies ci-dessus pour afficher ces images sur l'une de vos pages. Si vous voulez vérifier la disponibilité des images dans votre dossier trinket, il faut cliquer sur la troisième icône.



18 – Rechercher quelques images sur Internet et surtout leurs URLs. Rajouter ces images sur votre page. Il faudra donc utiliser leurs adresses absolues.

Comme vous pouvez le voir, la balise `img` n'est pas une balise block comme la balise `p` : elle ne force pas le passage à la ligne.

19 – Entourer vos images d'une balise `p` : `<p> </p>`

20 – Tentez de créer un lien sur l'image : insérez votre image entre `<a>` et ``

Images locales : ce n'est pas très correct d'afficher une image sur votre page en utilisant les ressources du serveur sur lequel elle est stockée. Lorsque vous créez réellement votre site, il faut donc télécharger vos images dans votre propre répertoire. Avec trinket, vous ne pouvez le faire qu'en ayant votre propre compte. Il suffit de le créer !

ETAPE FINALE – les pages que vous venez de créer ne sont pas atteignables autrement que via l'interface de trinket. Pour les récupérer en local, il faut :

- les télécharger avec le bouton SHARE et dézipper le dossier obtenu.
- cliquer sur l'un de vos fichiers HTML maintenant présent sur votre disque dur : votre navigateur devrait l'ouvrir et l'afficher.

Maintenant, vous pouvez télécharger chez vous un éditeur de code comme Notepad++ (<https://notepad-plus-plus.org/fr/>) pour travailler directement le code en local.



Merci à Hervé Owsinski pour son travail (très peu modifié par JNB)