

<https://clx.asso.fr/spip/?Les-tags-LINK-en-HTML>



HTML 4

Les tags LINK en HTML

- L'Association - Libre et Libertés Numériques -



Date de mise en ligne : vendredi 20 décembre 2002

Copyright © Club LinuX Nord-Pas de Calais - Tous droits réservés

La norme [HTML 4.0](#) (ainsi que HTML 4.01 et suivantes) facilite la gestion de documents volumineux en plusieurs pages.

Ainsi, vous pouvez définir les autres pages d'un document en cours de visualisation. Il faut considérer une page HTML donnée comme faisant partie d'une hiérarchie correspondant à un document complet. De la même façon que lorsque vous lisez un livre, vous savez en permanence quel est le chapitre lu, quelle section, etc.

LaTeX permet de faire ce genre de choses depuis longtemps, et les outils type *docbook* ou *sgml-tools* implémentent cet enrichissement de la norme HTML. Ainsi, chaque section pourra être traitée comme une page HTML. Les liens vers le chapitre dont elles font partie, les autres chapitres, la table des matières ou l'index seront mentionnées dans l'en-tête de la page HTML que vous visualisez, et le fureteur se chargera de gérer la hiérarchie.

Hiérarchie d'un document

Commençons par définir un document, et la hiérarchie de ses chapitres, sections et sous-sections. En français, on appelle cela le plan.

Notre document d'exemple sera un rapport de stage, son sujet traitera du cryptage par GPG (sujet traité par ailleurs sur le site de CLX, voir [FAQ clés GnuPG](#)).

Le rapport est organisé ainsi :

- Table des matières
- 1 Introduction
 - 1.1 Remerciements
 - 1.2 But du stage
- 2 Principe de base des clés asymétriques
 - 2.1 l'algorithme RSA
 - 2.2 L'algorithme DES
 - 2.3 L'algorithme DSA
- 3 Implémentation de GPG
 - 3.1 Gestion des clés
 - 3.1.a Création d'une clé GPG
 - 3.1.b Révocation d'une clé
- Conclusion
- Index

Le document est divisé en chapitre, eux-mêmes découpés en sections, sous sections, et enfin paragraphes. Ici, nous traiterons sur des pages séparées les chapitres et les sections, c'est-à-dire les niveaux 1 et 2, considérant que les niveaux inférieurs sont plus lisibles sur une seule page.

Ainsi, les sous sections *Création d'une clé GPG* et *Révocation d'une clé* seront sur la page *Gestion des clés*.

Si vous réalisez un tel rapport en LaTeX puis l'exportez en PDF, vous visualiserez en permanence à gauche la hiérarchie de votre document, et vous pourrez directement à un endroit donné d'un document. Si vous l'exportez en HTML, il y a fort à parier qu'avec des outils de conversion récents, vous aurez une série pages HTML correspondant à ce que nous avons évoqué, au sein desquelles vous pourrez trouver des balises ajoutées dans l'en-tête des pages.

Implantation des balises

Elles sont de la forme et doivent figurer dans l'en-tête de la page (la partie encadrée par le

L'idée étant que pour chaque page indépendante, on puisse indiquer la hiérarchie au dessus mais aussi au dessous au sein de laquelle elle s'intègre. Pour cela, vous disposez d'un certain nombre de valeurs prédéfinies, à mettre dans le paramètre REL de la balise LINK :

- top : la page situé tout en haut. Ici, ce serait la première page, qui pourrait ressembler à une table des matières ;
- up : la page située juste au dessus dans l'arborescence ;
- next et previous : les page situées au même niveau, suivantes et précédentes ;
- last et first : les premières et dernières pages du niveau.

Voilà pour la navigation. Par exemple, pour la page "L'algorithmme RSA", voici ce que cela donnerait :

Mais on peut aussi positionner d'autres balises, permettant un accès direct aux documents :

- **chapter**, pour le niveau 1 ;
- **section**, pour le niveau 2 ;
- **subsection**, pour le niveau 3 ;
- **appendices**, pour les annexes ;
- **glossary**, pour le glossaire ;
- **index**, pour l'index.

Au niveau de l'interface, ces balises de navigation directe ne sont malheureusement pas toujours gérées, mais si elles le sont, elles apparaîtront dans un menu popup très agréable à utiliser.

Formats multiples

Un autre intérêt des balises LINK est de permettre de visualiser des documents dans plusieurs langues, ou dans plusieurs formats. Pour cela, il suffit de renseigner les attributs "media" (pour le format) et "lang" (ou hreflang) pour la langue. Typiquement, ils s'insèrent dans une balise "alternative". Par exemple :

Notez qu'il est tout à fait possible de gérer conjointement plusieurs formats d'affichage par le biais d'une même page HTML en utilisant les styles CSS :

```
p { color: rgb(230, 100, 180) } H1 { color: blue} H1 { text-align: center }
```

Ainsi, l'exemple suivant ajoute un effet sonore aux ancres (tag "A") pour être utilisé par un synthétiseur vocal :

```
A { cue-before: uri(bell.aiff); cue-after: uri(dong.wav)}
```

Les media définis par la norme HTML 4.0 sont :

- **screen**, pour les écrans d'ordinateurs classiques ;
- **aural**, destiné à la synthèse vocale ;
- **tty**, pour les affichage à taille fixe, utilisables entre autres par lynx ou mosaic ;
- **print**, pour une sortie imprimante ;
- **projection**, pour les vidéo-projecteurs ;
- **tv**, pour une diffusion vers un téléviseur (basse résolution entre autres) ;
- **handed**, pour les PDA type PSION, PALM ou VR3 ;
- **braille**, pour les afficheur destinés aux aveugles [1] ;
- **all**, pour tout le monde. Cela implique un format compatible avec tous.

Autres types de liens

Vous le verrez plus loin, Mozilla gère également les liens non normalisés par le W3C. Cela dit, sans sortir du cadre strict de la norme, un certain nombre de liens sont utilisables, pour préciser diverses choses :

- **help**, pour donner une page précisant plus d'informations, de l'aide, des liens *etc.* ;
- **authors**, les auteurs de la page, une balise par auteur ;
- **copyright**, qui désigne une page donnant toutes les informations de copyright *ie.* la licence, les droits d'auteurs, *etc.* ;
- **bookmarks**, qui permet de donner des liens directs sur des documents annexes en relation avec la page courante.

Et d'autres balises non normalisées en HTML 4.0 mais gérées par les navigateurs :

- **search**, désignant une page permettant de faire une recherche sur le document (le site) que vous visualisez ;
- **shortcut icon**, qui permet d'indiquer un icône à afficher correspondant à la page que vous visualisez [2], vous le verrez dans l'exemple SPIP plus loin.

Implémentation courante et SPIP

Ces balises sont bien sûr très bien adaptées aux gros documents, comme un rapport, un livre, ou ce genre de choses trop volumineux pour être vu en une seule page, mais vous allez voir, on peut aisément en détourner l'usage pour l'adapter à n'importe quel site Web.

Ainsi, sur le site du CLX, si votre navigateur gère les balises de liens, vous pouvez naviguer dans tout le site sans passer par le liens hypertexte.

Pour utiliser ces balises dans un site Web complet, on peut imaginer la totalité du site comme une sorte de gros document. Les rubriques et sous rubriques pourraient alors être vues comme des chapitres et des sections, les pages contenues dans une rubriques seraient alors des sous sections. Il est également possible de sortir du cadre strict de la norme du W3C, et de créer de nouvelles balises, comme "rubrique", "sous-rubrique", "article", "brève" [3]

... qui seront correctement traitées par Mozilla, en sachant que la page risque alors de ne plus être validée par le validateur [4] du [W3C](#).

Ces balises rendent tout de même possible l'utilisation des barres de navigation de tous les navigateurs qui le gèrent [5], et peuvent être intégrées aisément au sein d'un squelette [SPIP](#) [6] de la sorte :

```
<BOUCLE_ART(ARTICLES){id_article}> CLX - #TITRE <BOUCLE_RUBall(RUBRIQUES){id_secteur=#ID_SECTEUR}>
</BOUCLE_RUBall> <BOUCLE_RUBal(RUBRIQUES){id_rubrique}> </BOUCLE_RUBal>
<BOUCLE_RUBarts(ARTICLES){id_rubrique}> </BOUCLE_RUBarts> <BOUCLE_RUBrub(RUBRIQUES){id_parent}>
</BOUCLE_RUBrub> <BOUCLE_RUBbrv(BREVES){id_rubrique}> </BOUCLE_RUBbrv> <BOUCLE_AUTal(AUTEURS){id_article}>
</BOUCLE_AUTal> <BOUCLE_ARTnext(ARTICLES){id_rubrique}{par date}{inverse}{age_relatif>0}{0,1}>
</BOUCLE_ARTnext> <BOUCLE_ARTprev(ARTICLES){id_rubrique}{par date}{age_relatif<0}{0,1}> </BOUCLE_ARTprev>
<BOUCLE_ARTfirst(ARTICLES){id_rubrique}{par date}{0,1}> </BOUCLE_ARTfirst>
<BOUCLE_ARTlast(ARTICLES){id_rubrique}{par date}{inverse}{0,1}> </BOUCLE_ARTlast> </BOUCLE_ART>
```

Navigateurs utilisant ces balises

On ne peut que regretter une fois de plus que les navigateurs les plus répandus ne soient pas compatibles avec les standards définis par le [WWW consortium](#). Cependant, d'autres navigateurs gèrent correctement ces balises depuis leur apparition (Lynx et Mosaic), et d'autres relativement moins répandus les gèrent depuis peu. Voici la liste (non exhaustive) des navigateurs supportant les liens LINK :

- [Mozilla](#) (sauf le 1.0) ;
- iCab ;
- Opera 5.0 (Macintosh uniquement) et Opera 7 bêta (toutes plateformes) ;
- Lynx, bien sûr ;
- NCSA mosaic ;
- CAB (sur Atari) ;
- Emacs-w3 ;
- AOLpress (eh oui !).

Netographie

- L'excellente page de Mathias Gutfeld [wenn Sie sprechen deutsch](#) et sa page de liens en Anglais (ouf) ! : <http://www.subotnik.net/html/link.h...>
- [101 trucs que Mozilla sait faire et qu'IE ne sait pas](#)
- [Le HTML 4.01](#) par le W3C
- La page du [Web Design Group](#)

Voilà, bon courage, et à vos claviers !

[1] non voyants si vous préférez une pointe de démagogie.

[2] IE le gère bien cette fois-ci.

[3] Attention toutefois à ne pas mettre d'accent.

[4] oops...

[5] les IE* n'en font une fois de plus pas partie.

[6] Consultez également les [squelettes](#) mis en place par le CLX.