



Manuel du développeur de DocGen

Auteur(s) : Florian Legendre



Légendes et Abréviations utilisées

Question : Ceci est une question de l'enseignant

Réponse : Ceci est une réponse de l'enseignant ou validée par l'enseignant

Réponse : *Ceci est une réponse du ou d'un des auteurs non validée par l'enseignant*

```
1 Ceci est du code source.  
2 Selon les langages, différents mots seront colorés selon  
3 si ce sont des mots clefs ou non (comme int, char, etc.).
```

Listing 1 – Exemple de code source

```
Ceci est un formatage automatique Latex d'un texte copié-collé  
directement depuis un terminal Bash ayant valeur de capture  
d'écran. La coloration correspond à une coloration quelconque  
d'un terminal Bash (les chemins étant habituellement coloré et  
le nom de l'utilisateur aussi comme crex@crex:~$ ...)
```

Listing 2 – Exemple d'une pseudo capture d'écran Bash

Table des matières

I	Cahier des Charges	3
II	Spécifications Fonctionnelles	4
1	Génération des index	5
1.1	Index hiérarchiques	5
1.1.1	Définition	5
1.1.2	Solution 1	6
III	Bibliographie et Glossaires	9
IV	Annexes	10

Première partie

Cahier des Charges

Deuxième partie

Spécifications Fonctionnelles

Chapitre 1

Génération des index

Index hiérarchiques

1.1.1 Définition

Un index hiérarchique ou arborescent est une forme d'indexation où les entrées sont contenues dans d'autres entrées, etc. reflétant ainsi la généralisation (on remonte dans l'arbre) ou la spécification (on descend dans l'arbre) des entrées les unes par rapport aux autres.

Ce type d'index est appelé *indexT* dans le code en référence à l'anglais *Tree Index*.

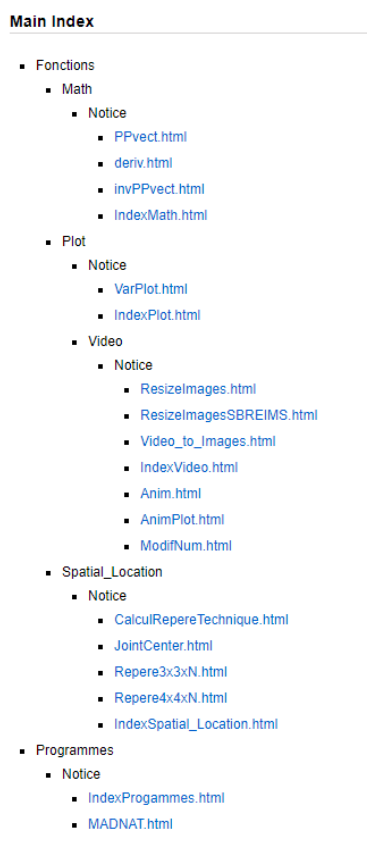


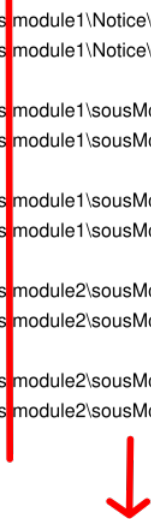
FIGURE 1.1 – Exemple d'indexation hiérarchique

1.1.2 Solution 1

Description de l'algorithme

Étape 1

On tronque de la liste des chemins récoltés la partie correspondant à la racine où est effectué l'index et on en profite pour mettre les chemins tronqués dans un tableau de chaînes de caractères.



```
E:\projects\MonProjet\codes\module1\Notice\script1.html
E:\projects\MonProjet\codes\module1\Notice\Manpage_***.html

E:\projects\MonProjet\codes\module1\sousModule1\Notice\script.html
E:\projects\MonProjet\codes\module1\sousModule1\Notice\Manpage_***.html

E:\projects\MonProjet\codes\module1\sousModule2\Notice\script.html
E:\projects\MonProjet\codes\module1\sousModule2\Notice\Manpage_***.html

E:\projects\MonProjet\codes\module2\sousModule1\Notice\script.html
E:\projects\MonProjet\codes\module2\sousModule1\Notice\Manpage_***.html

E:\projects\MonProjet\codes\module2\sousModule1\sousSousModule1\Notice\script.html
E:\projects\MonProjet\codes\module2\sousModule1\sousSousModule1\Notice\Manpage_***.html
```



```
module1\Notice\script1.html
module1\Notice\Manpage_***.html

module1\sousModule1\Notice\script.html
module1\sousModule1\Notice\Manpage_***.html

module1\sousModule2\Notice\script.html
module1\sousModule2\Notice\Manpage_***.html

module2\sousModule1\Notice\script.html
module2\sousModule1\Notice\Manpage_***.html

module2\sousModule1\sousSousModule1\Notice\script.html
module2\sousModule1\sousSousModule1\Notice\Manpage_***.html
```

FIGURE 1.2 – Suppression de la partie commune inutile

Étape 2

Si le paramètre *isExhaustive* est à *false* on supprime également de l'indexation les entrées qui ne sont pas des pages de manuel comme illustré ici :

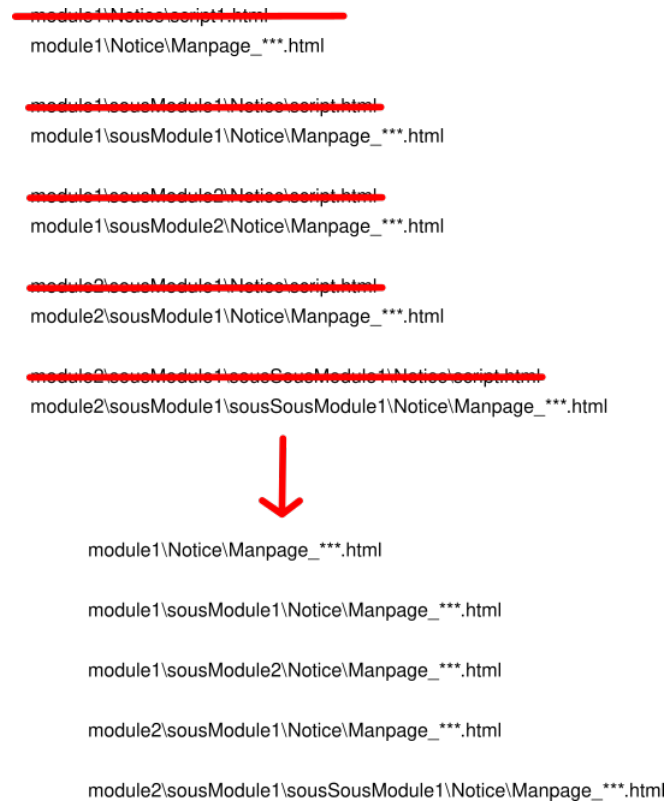


FIGURE 1.3 – Suppression des entrées qui ne sont pas des pages de manuel

Étape 3

Chaque chemin absolu est ensuite traité par la méthode *globalIndexArrMake(obj, subpathsList)* qui se charge d'en faire un tableau de noeuds¹. Les chemins absolus sont reconstitués en reconcaténant la partie tronquée à l'étape 1.

module1\Notice\Manpage_***.html

module1\sousModule1\Notice\Manpage_***.html

module1\sousModule2\Notice\Manpage_***.html

module2\sousModule1\Notice\Manpage_***.html

module2\sousModule1\sousSousModule1\Notice\Manpage_***.html



	globalIndexArr								
profondeur:	0	1	2	1	2	3	...	0	...
nom_noeud:	module1	Notice	Manpage_***.html	sousModule1	Notice	Manpage_***.html	...	module2	...
chemin_absolu:	E:/....	E:/....	E:/....	E:/....	E:/....	E:/....	...	E:/....	...

FIGURE 1.4 – Transformation des chaînes de caractères en tableaux de noeuds

Étape 4

1. Une structure de donnée serait probablement plus appropriée ici...

Troisième partie

Bibliographie et Glossaires

Quatrième partie

Annexes