# Un algorithme pour vérifier les balises HTML

La plupart des balises HTML fonctionnent par paire : une balise ouvrante comme <h1>, et la balise fermante correspondante avec le slash devant le nom : </h1>.
Les règles pour que le code soit correct sont les suivantes :

- Toutes les balises ouvrantes doivent avoir une balise fermante correspondante, sauf les balises autofermantes (meta, link, img, br).
- Pour chaque paire de balises, elles peuvent être :
  - l'une après l'autre <X> </X> <Y> </Y>
  - l'une à l'intérieur de l'autre <X> <Y> </Y> </X>
  - mais **PAS** en chevauchement ~~<X>~~ ~~<Y>~~ ~~</X>~~ ~~</Y>~~

Voici une méthode pour vérifier que toutes les balises d'un fichier sont correctes :
On prend toutes les balises dans l'ordre, sauf les autofermantes qu'on ignore, et à chaque fois :

- Si la balise est ouvrante, on rajoute son nom sur un papier en haut d'une pile.
- Si la balise est fermante, on vérifie qu'en haut de la pile on a bien une balise de même nom, et on l'enlève de la pile. Sinon, c'est une erreur.

À la fin, si la pile est vide et qu'on n'a pas rencontré d'erreur, alors le code est correct.

**Exercice 1 :** Munissez vous de petits bouts de papier et d'un stylo, et vérifiez les codes suivants avec cette méthode. Indiquez à chaque fois si le code est correct, sinon soulignez l'endroit où une erreur est détectée.
1) <html> <head> </head> <body> <h1> </h1>     </body> </html>

2)  Le texte entre les balises n'est pas important pour vérifier leur validité. 

3) <ol> <li> 1 </li> <li> <ol> <li> 2.1 </li> 2.2 </li> </ol> </ol>

4) <html> <head> <meta charset="UTF-8"> <link href="style.css"> </head> <body>  

**Exercice 2 :** Si vous avez votre site web sur les ordinateurs de la salle, ouvrez une des pages html dans Geany, et vérifiez (et corrigez en même temps) l'imbrication des balises dans votre code.

Cette méthode est une suite **précise** d'instructions pour résoudre **tous** les problèmes du type : est-ce que les balises de ce code HTML sont correctement imbriquées ? On appelle ce genre de méthode un **algorithme**. On peut l'appliquer "à la main" mais c'est aussi possible de programmer un ordinateur pour qu'il le fasse automatiquement.

**Exercice 3 :** Voici un programme en python qui exécute cet algorithme. Il utilise des fonctionnalités de python qu'on n'a pas vues en classe, donc c'est normal de ne pas le comprendre en entier.

```
1  import re
2
3  autofermantes = {"meta","link","img","br","source"}
4
5  pile=[]
6  def empiler(x):
7      pile.append(x)
8  def depiler():
9      return pile.pop()
10 def est_vide():
11     return len(pile)==0
12
13 numero_ligne=1
14 #la boucle for suivante parcourt toutes les lignes du fichier "index.html"
15 for line in open("index.html").readlines():
16     #la boucle suivante parcourt toutes les balises html de la ligne actuelle
17     for t in re.findall('\<\w+.*?\>|\</\w+.*?\>', line):
18         balise = re.search('\</?(\w+)',t).group(1)
19         #si la balise ne commence pas par "</", c'est une balise ouvrante
20         if t[:2]!="</":
21             if balise not in autofermantes:
22                 empiler(balise)
23         #sinon c'est une balise fermante
24         else:
25             if est_vide():
26                 print("erreur l"+str(numero_ligne)+", balise fermante "
27                     +balise+" mais aucune balise ouverte")
28             else:
29                 dessus_pile = depiler()
30                 if balise != dessus_pile:
31                     print("erreur l"+str(numero_ligne)+", balise fermante "
32                         +balise+", dernière ouverte "+dessus_pile)
33     numero_ligne += 1
34
35 if not est_vide():
36     print("erreur, balises non fermées :",pile)
37 print("Verification terminée")
```

1) À quoi servent à votre avis les lignes qui commencent par # ?

2) Combien de **fonctions** sont définies dans ce programme ? Donnez leur noms.

3) Tracer une flèche entre chaque **else** ("sinon") et le **if** ("si") qui lui correspond.

4) Quelles sont les lignes qui correspondent aux messages affichés si une erreur est trouvée ?

5) Que va afficher ce programme si aucune erreur n'est trouvée ?

6) Quel est le nom du fichier qui va être vérifié par ce programme ?

7) À quoi sert la variable numero_ligne dans ce programme ?

8) Vous pouvez télécharger ce code à l'adresse https://nodfs.xyz/verif.py , le placer dans votre dossier de site web, l'ouvrir avec Thonny et l'exécuter pour vérifier vos pages HTML.