Data Mining
Introduction au web mining

Thierry Dhorne

2 mars 2015

Intérêt du web mining

- le web est constitué de
  - texte
  - tableaux numériques
  - images
  - vidéos
  - ....
- ➤ il contient donc des informations plus ou moins stratégiques
- qu'il est éventuellement judicieux d'analyser

Comment miner le web ?

- de manière statique et manuelle

➤ comme T. Dhorne demande aux étudiants de le faire : copié-collé

- de manière automatique

➤ c'est parfois plus rapide mais parfois moins (il faut programmer un peu)

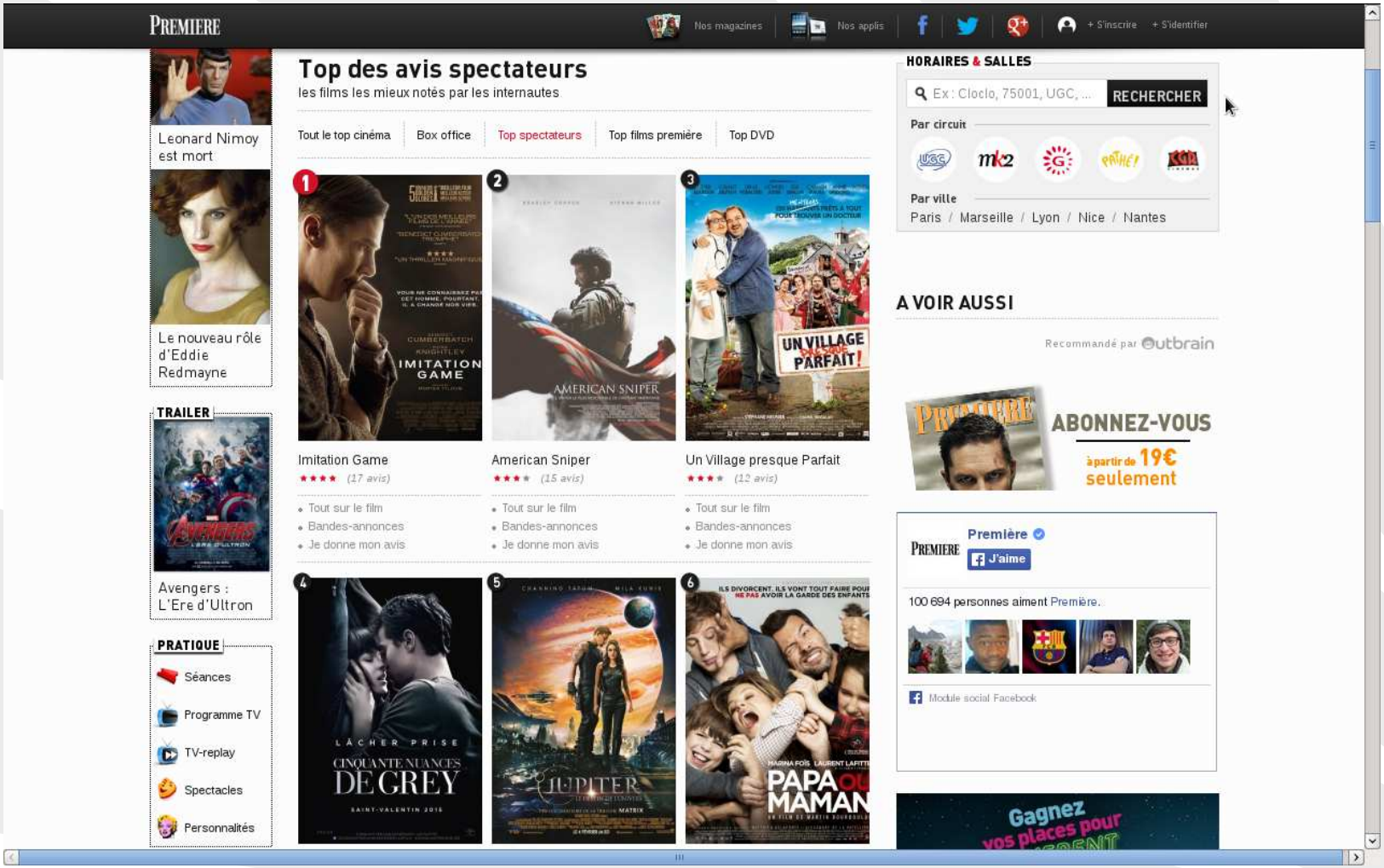
- de manière dynamique

➤ il faut programmer (une fois !) mais après ça marche tout seul !

Exemples d'applications

- suivre en temps réel les commentaires sur votre site
- disposer d'informations statistiques au jour le jour
- scruter le web pour trouver des informations nouvelles
- surveiller un concurrent (dynamique du site)

Exemple 1

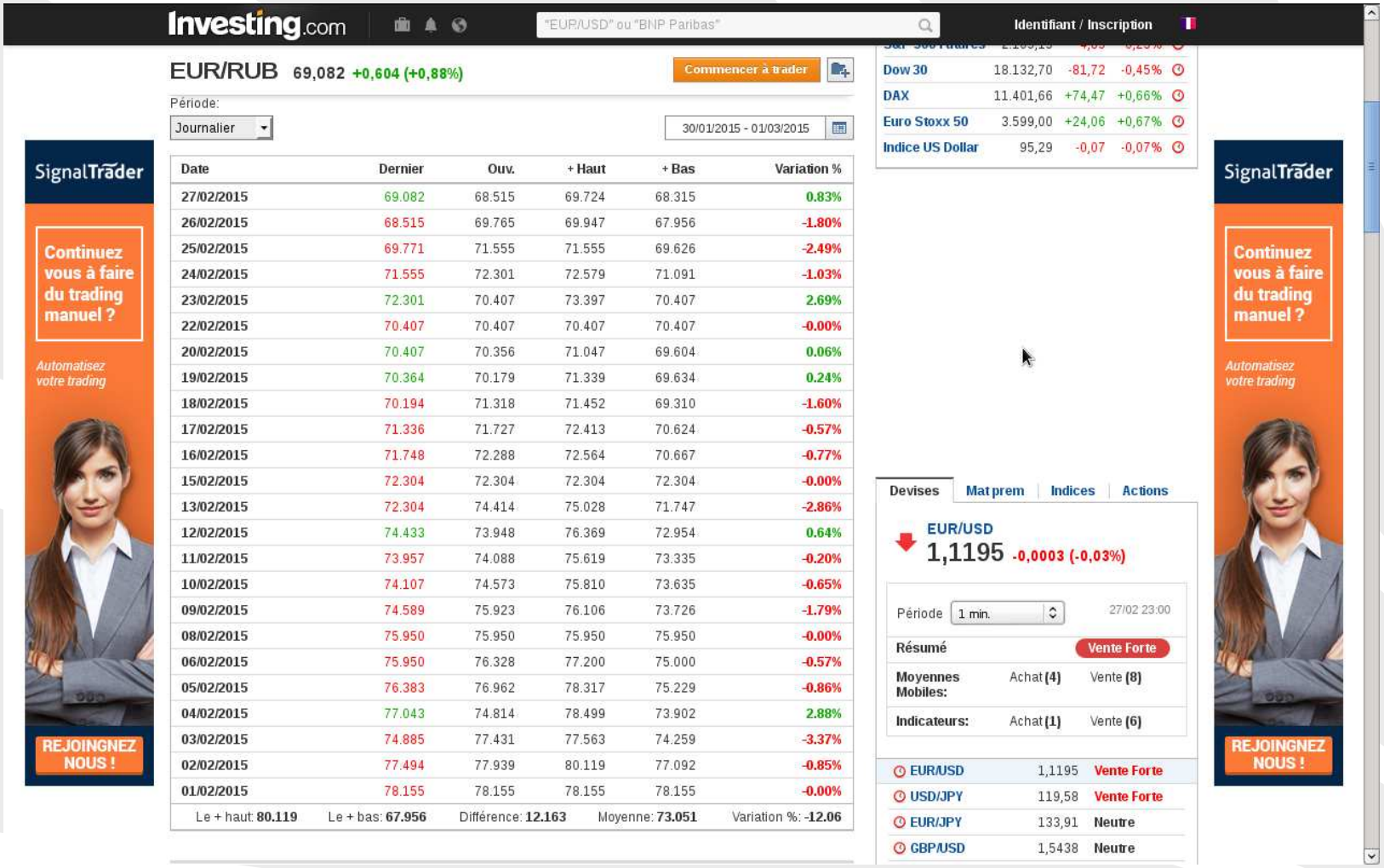


Exemple 2

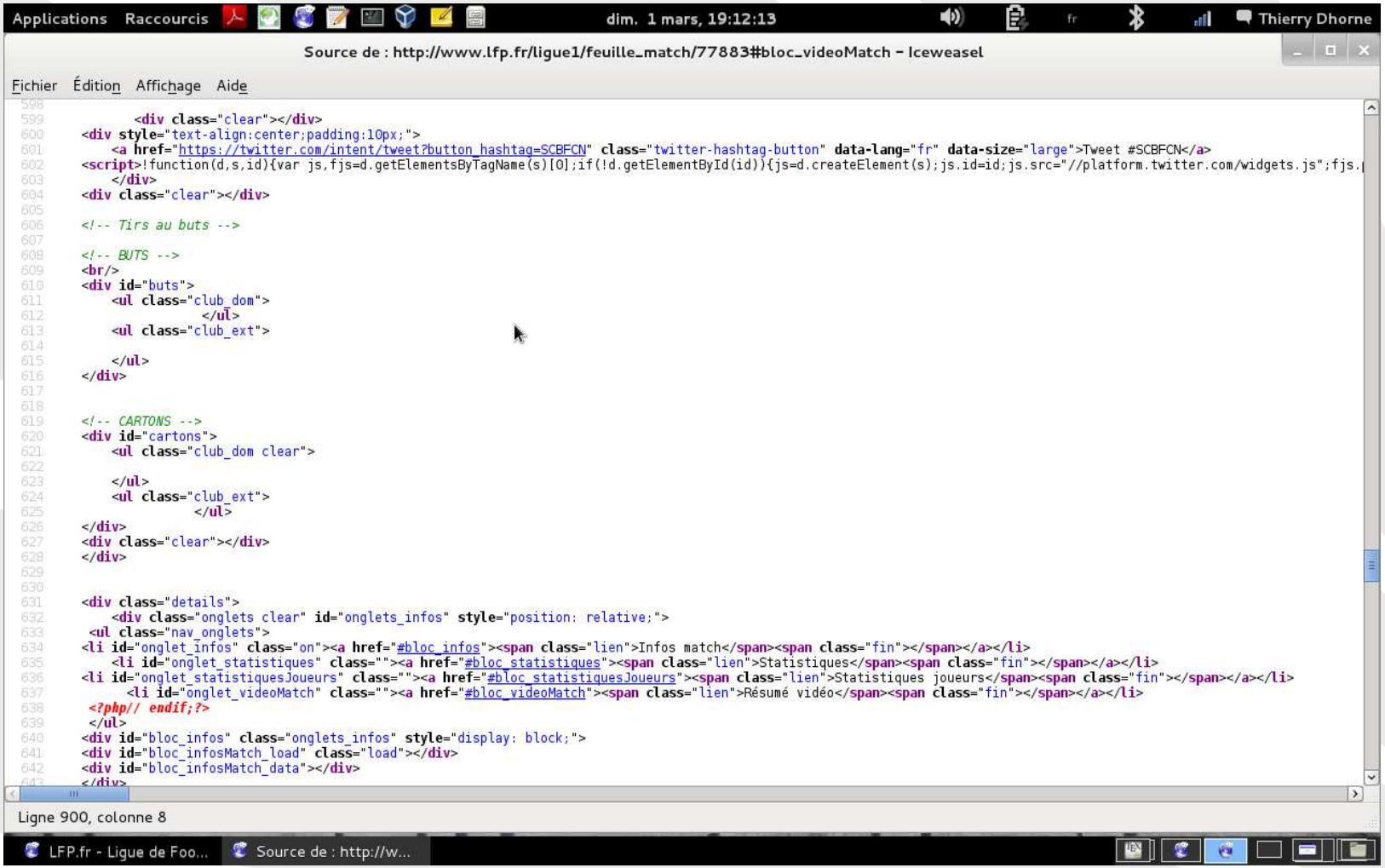
- suivre les sites des départements STID et voir ceux qui bougent le plus

➤ on scane les sites tous les jours et on regarde s'ils ont changé par rapport à la veille

Exemple 3



Vu du browser



XML - HTML

- XML : eXtensible Markup Language

➤ les langages de markup permettent l'annotation de structures distinctes du texte

➤ extensible signifie que l'on peut créer ses propres annotations

- en XML, les marqueurs de structures sont $<$ et $>$
- HTML : HyperText Markup Language est le langage i initial du web et reste le plus important aujourd'hui

➤ mais plus le plus efficace

- HTML (bien que plus ancien) peut être considéré comme un sous XML

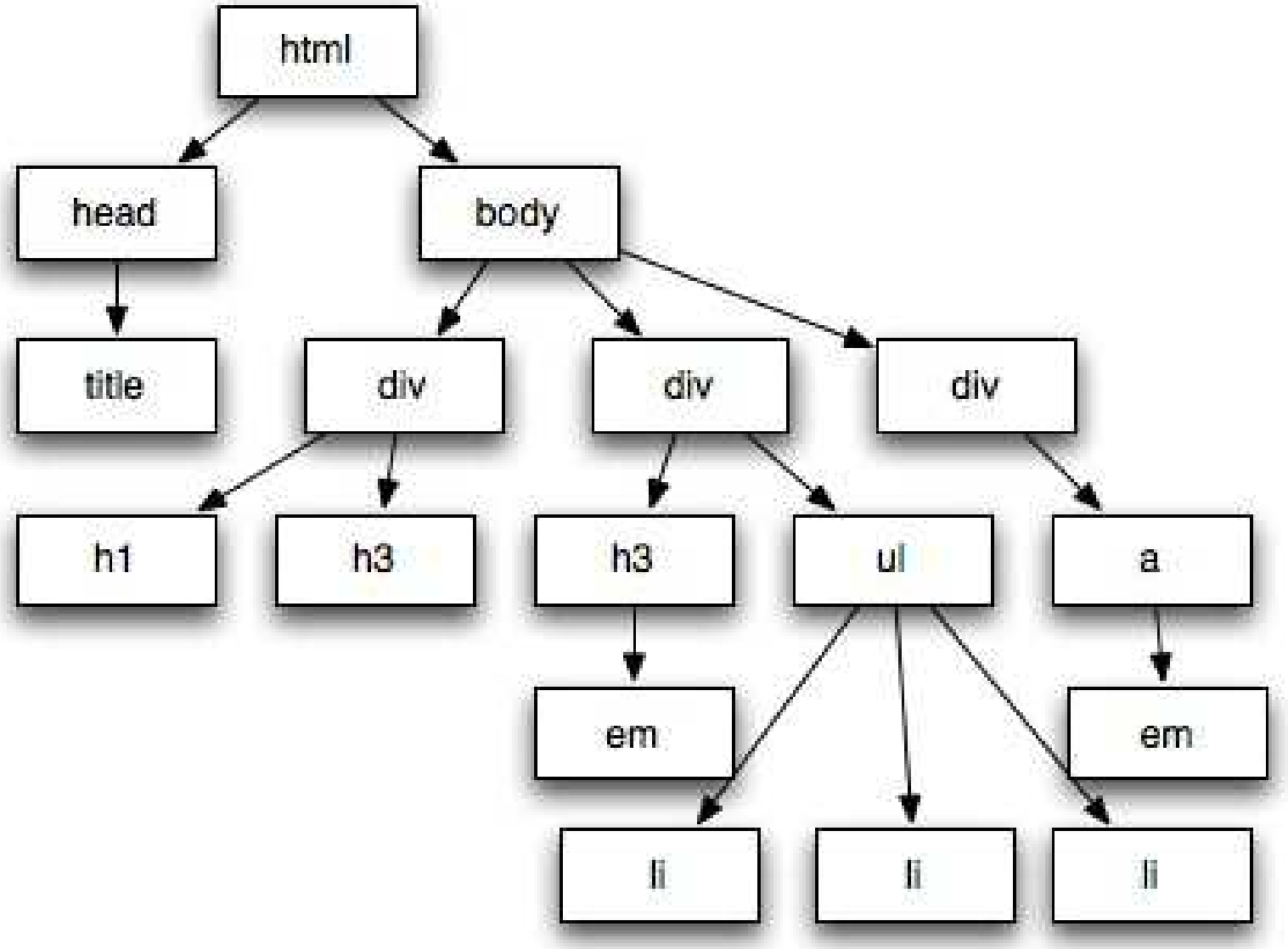
➤ moins rigoureux (et moins flexible)

HTML

- HTML est simplement un ensemble de marques

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "h
<html>
<head>
<title>The document title</title>
</head>
<body>
<h1>Main heading</h1>
<p>A paragraph.</p>
<a href = "www.stid-vannes.fr">Statistics Web
</body>
</html>
```

HTML - Représentation



HTML - Structure

- les boîtes s'appellent nœuds
- les lignes s'appellent arêtes

➤ une arête va de a à b si :

```
<a>
      <b>
      </b>
</a>
```

★ il n'y a qu'un unique chemin de la racine à un nœud quelconque

HTML - Structure

- à chaque nœud peuvent être associés les élements suivants :

– un nom (requis mais non nécessairement unique)
– un nombre (quelconque) d'attributs
– un texte (optionnel)

➤ exemple

```
<a href = "www.stid-vannes.fr">
Statistiques du site
</a>
```

➤ avec

– nom du nœud : a
– attribut : href de valeur "www.stid-vannes.fr"
– texte : Statistiques du site

Xpath

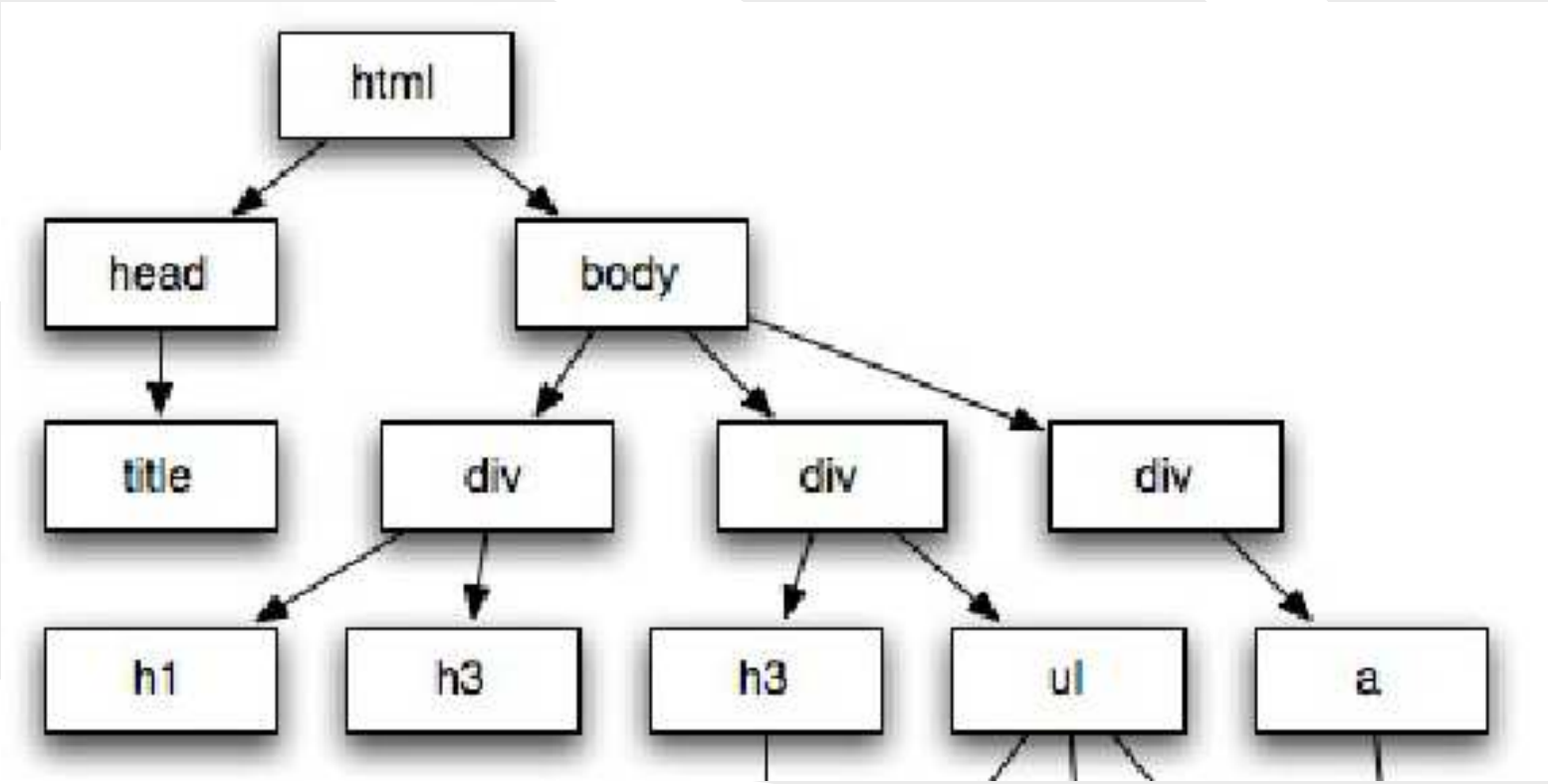
- XPath est un langage de requête qui permet de retrouver des nœuds specifiés dans un arbre XML/HTML

➤ quelques symboles utilisables pour représenter un chemin sur l'arbre :

– / trouve la racine
– // sélectionne un nœud quelconque sur l'arbre
– . sélectionne le nœud courant
– .. sélectionne le parent du nœud courant
– @ sélectionne des attributs
– nodename recherche la position du nœud appelé

Navigation

- pour aller jusqu'à la balise (ancre) *a* on peut utiliser
  - "/../body/div/a"
  - "//a"



Navigation

- si la balise est :

```
<a href = "stid.vannes.fr">stats</a>
```

➤ on peut chercher avec

```
"//a[@href = 'stid.vannes.fr']"
```

➤ ou

```
"//a[text() = 'stats']"
```

R XML

- R dispose d'un package qui permet de travailler avec des arbres XML/HTML
- les deux fonctions les plus utiles (pour nous) sont

➤ htmlTreeParse

récupère une page HTML et crée une structure d'arbre interne à R

➤ on peut alors utiliser XPath pour parcourir l'arbre

```
>url <- "https://fr.wikipedia.org/wiki/Charolaise"
>doc <- htmlTreeParse(url,useInternalNodes = TRUE)
```

➤ getNodeSet recherche dans le document des nœuds spécifiés :

```
> x1 <- getNodeSet(doc, "//div[@class = 'bd']")
> x2 <- getNodeSet(doc, "//a")
```

Web mining

- le web est d'abord un monde automatique

➤ Google,

★ l'effort de recherche est considérable (big data)

- même si on cherche des choses simples (?) il faut automatiser

➤ d'où la nécessité de réfléchir

– aux objectifs
– aux enjeux
– aux techniques