

# INF600A: Laboratoire #5

## Analyse (simplifiée) de fichiers XML

### Solution

---

#### 1. Fichiers motifs.rb et balise-xml.rb

```
# Motif pour association entre un (1) attribut et sa valeur.
ATTRIBUT_ET_VALEUR =
  %r{
    (#{NOM_ATTRIBUT})
    \s*=\s*
    "(#{VALEUR_ATTRIBUT})"
  }x

def self.les_attributs( chaine_attributs )
  attributs = {}
  while match = Motifs::ATTRIBUT_ET_VALEUR.match( chaine_attributs )
    chaine_attributs = match.post_match

    nom_attribut = match[1].to_sym
    valeur_attribut = match[2]

    attributs[nom_attribut] = valeur_attribut
  end

  attributs
end

# Autre solution, plus simple, avec scan, et purement fonctionnelle
def self.les_attributs( chaine_attributs )
  chaine_attributs.scan( Motifs::ATTRIBUT_ET_VALEUR )
    .map { |nom, valeur| [nom.to_sym, valeur] }
    .to_h
end
```

---

## 2. Fichier fichier-xml.rb

```
def each
  @balises ||= les_balises_de_la_source( @lignes )

  @balises.each { |b| yield(b) }
end
```

---

### 3. Fichier enumerables\_test.rb

```
let(:lignes) { [ '<B1 a1="v1" a2="v2">',
                 '<B1 a1="v1" a3="v3">',
                 '<B1 a3="v3" a1="v1">',
                 '',
                 '</B3>',
                 '<B3/>',
                 '',
                 '<B2 a3="v3">',
                 '<B2 a3="v3"/>',
                 '<B2 a1="v1"/>',
                 ]
}

it "identifie les diverses balises ouvrantes" do
  attendu = ["B1", "B2"]

  FichierXML.balises(lignes)
    .select(&:ouvrante?)
    .map(&:balise)
    .uniq
    .must_equal attendu
end

it "identifie tous les attributs, independamment de la balise" do
  attendu = [:a1, :a2, :a3]

  FichierXML.balises(lignes)
    .flat_map(&:attributs)
    .uniq
    .must_equal attendu
end

it "compte le nombre d'occurrences de chacun des attributs" do
  attendu = {:a1 => 4, :a2 => 1, :a3 => 4}

  FichierXML.balises(lignes)
    .flat_map(&:attributs)
    .group_by { |x| x }
    .map { |a, occs| [a, occs.size] }
    .to_h
    .must_equal attendu
end
```