

Données du Web - TD2 1

Basil Dalié - Yannis Naidja

Octobre 2019

Chapitre 1

XPath : films

1.1 Les titres des films

```
1 /descendant::TITRE/text()
```

1.2 Les titres des films parus en 1990

```
1 /descendant::ANNEE[text() =  
  ↪ '1990']/parent::FILM/child::TITRE/text()
```

1.3 Le resume d'Alien

```
1 /descendant::TITRE[text() =  
  ↪ 'Alien']/parent::FILM/child::RESUME/text()
```

1.4 Quel est le dernier film du document ?

```
1 /descendant::FILM[last()]
```

1.5 Quel est le dernier film du document paru en 1990 ?

```
1 /descendant::ANNEE[text() = '1990'][last()]/parent::FILM
```

1.6 Les titres des films qui ont un résumé

```
1 /descendant::RESUME[text() !=  
  ↪ '']/parent::FILM/child::TITRE/text()
```

1.7 Les titres des films dont l'élément résumé n'est pas présent.

```
1 /descendant::FILM[not(child::RESUME)]/child::TITRE/text()
```

1.8 Donnez les noms des acteurs qui ont joué dans Vertigo.

```
1 /descendant::ARTISTE[@id = /descendant::FILM[TITRE =  
  → 'Vertigo']/ROLES/ROLE/@idref]/ACTNOM/text()
```

1.9 Qui a mis en scène Vertigo ?

```
1 /descendant::ARTISTE[@id = /descendant::FILM[TITRE =  
  → 'Vertigo']/MES/@idref]/ACTNOM/text()
```

1.10 Donnez tout les films du directeur de Vertigo.

```
1 /descendant::MES[@idref = /descendant::FILM[TITRE =  
  → 'Vertigo']/MES/@idref]/preceding-sibling::TITRE/text()
```

1.11 Donnez les titres des films qui contiennent la lettre "V" (utiliser la fonctioncontains()).

```
1 //TITRE[contains(text(), 'V')]/text()
```

1.12 Les titres des films où l'acteur Bruce Willis a joué.

```
1 descendant::FILM/ROLES/ROLE[@idref =  
  → /descendant::ARTISTE[ACTNOM/text() = 'Willis' and  
  → ACTPNOM/text() = 'Bruce']/number(@id)]
```

1.13 Quel rôle joue Harvey Keitel dans Reservoir dogs ?

```
1 descendant::FILM[TITRE = 'Reservoir dogs']/ROLES/ROLE[@idref =  
  → /FILMS/ARTISTE[ACTNOM = 'Keitel' and ACTPNOM =  
  → 'Harvey']/@id]/text()
```

1.14 Qui a joué avec Harvey Keitel dans Reservoir dogs ?

```
1 /descendant::ARTISTE[@id = /descendant::FILM[TITRE/text() =  
  ↳ "Reservoir dogs"]/ROLES/ROLE/@idref and ACTNOM/text() !=  
  ↳ 'Keitel' and ACTPNOM/text() != 'Harvey']
```

1.15 Donnez les nœuds qui ont exactement trois descendants (utiliser la fonctioncount()).

```
1 /descendant::*[count(*)=3]
```

1.16 Donnez les nœuds dont le nom contient la chaîne "TI" (utiliser la fonctionname()).

```
1 /descendant::*[contains(name(), 'TI')]
```

1.17 Quel est le titre du film qui précède immédiatement Shining (dans l'ordre du document) ?

```
1 /descendant::FILM[TITRE =  
  ↳ 'Shining']/preceding-sibling::FILM[1]/TITRE/text()
```

Chapitre 2

XPath : films

2.1 Le nom complet de toutes les recettes

```
1 /recettes/recette/@nom
```

2.2 Les ingrédients de la recette dont le nom court est "Chiffonnade" ;

```
1 /recettes/recette[@nomCourt =  
  ↳ "Chiffonnade"]/materiel/ingredient/text()
```

2.3 Le nom complet des recettes utilisant du "persil" ;

```
1 /recettes/recette/materiel/ingredient[contains(text(),  
  ↳ "persil")]/parent::materiel/parent::recette/@nom
```

2.4 (Sans utiliser l'axechild) Le nom complet des recettes utilisant du "persil"

```
1 /descendant::ingredient[contains(text(),"persil")]/ancestor::recette/@nom
```

2.5 Le nom complet des recettes ayant plus de deux ingrédients, et contenant des oeuf

```
1 /recettes/recette[count(materiel/ingredient) > 2 and  
  ↳ materiel/ingredient[contains(text(),"oeuf")]]/@nom
```

2.6 (Sans utiliser la fonctioncount()) Le nom complet des recettes ayant plus de deux ingrédients, et contenant l'ingrédient "huile"

```
1 /recettes/recette/materiel/ingredient[last() > 2 and  
  → contains(text(),  
  → "huile")]/parent::materiel/parent::recette/@nom
```

2.7 La dernière recette du document

```
1 /recettes/recette[last()]/@nom
```

Chapitre 3

XPath : films

Exprimer en XPath les interrogations suivantes.

3.1 Le numéro des trains qui possèdent une voiture-bar

```
1 /gare/train/voiture/bar/parent::voiture/parent::train/@numero)
```

3.2 Le nom des usages ayant effectué au moins une réservation ;

```
1 /gare/usager[@id = /gare/train/voiture/resa/@id]/@nom
```

(Extra) Est-il possible d'exprimer en XPath les requêtes suivantes ?

3.3 Le numéro des trains dont au moins 2 places sont réservées :

```
1 /gare/train/voiture[count(resa) >= 2]/parent::train/@numero
```

3.4 Le nom des personnes ayant réservé exactement deux fois.

Impossible, car en XPath on ne peut pas compter le nombre d'occurrence d'un attribut

3.5 Les usagers n'ayant effectué aucune réservation

```
1 /gare/usager[not(@id = /gare/train/voiture/resa/@id)]
```