

Esercitazione XPATH

Paolo Papotti

<http://papotti.dia.uniroma3.it>



Recipes

```
<!ELEMENT recipes (recipe*)>
<!ELEMENT recipe (title, ingredient+, preparation,
                  comment?, nutrition)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT ingredient (ingredient*, preparation?)>
<!ATTLIST ingredient
  name CDATA #REQUIRED
  amount CDATA #IMPLIED
  unit CDATA #IMPLIED>
<!ELEMENT preparation (step+)>
<!ELEMENT step (#PCDATA)>
<!ELEMENT nutrition EMPTY>
<!ELEMENT comment (#PCDATA)>
<!ATTLIST nutrition
  calories CDATA #REQUIRED
  fat CDATA #REQUIRED
  carbohydrates CDATA #REQUIRED
  protein CDATA #REQUIRED
  alcohol CDATA #IMPLIED>
```

```
<recipes>
  <recipe>
    <title>Beef Parmesan</title>
    <ingredient name="beef" amount="1.5"
                unit="pound"/>
    <ingredient name="onion" amount="1"/>
    <ingredient name="green rings"
                amount="1"/>
    <preparation>
      <step>Boil pasta</step>
    </preparation>
    <comment>... </comment>
    <nutrition calories="1167" fat="23"
               carbohydrates="45" protein="32"/>
  </recipe>
  ...
```

XPath solutions - recipes

"The titles of all recipes that use olive oil."

- `//recipe[ingredient/@name="olive oil"]/title`
- `//recipe[.//ingredient/@name="olive oil"]/title`
- `//recipe[.//ingredient/@name="olive oil"]/title/text()`

XPath queries - recipes

1. "The titles of all recipes, returned as strings."
2. "The titles of all recipes with less than 500 calories."
3. "The titles of all recipes that do not use butter."
4. "The amount of sugar needed for Zuppa Inglese."
5. "The first two steps in preparing Zuppa Inglese."
6. " The titles of all recipes that requires five steps."

XPath - recipes

1. "The titles of all recipes, returned as strings."
- `fn:doc("recipes.xml")//title/text()`

XPath - recipes

1. "The titles of all recipes, returned as strings."
 - `fn:doc("recipes.xml")//title/text()`
2. " The titles of all recipes with less than 500 calories."
 - `//recipe[nutrition/@calories < 500]/title`
 - `//recipe[//@calories < 500]/title [no!]`

XPath - recipes

1. "The titles of all recipes, returned as strings."
 - `fn:doc("recipes.xml")//title/text()`
2. " The titles of all recipes with less than 500 calories."
 - `//recipe[nutrition/@calories < 500]/title`
 - `//recipe[//@calories < 500]/title [no!]`
3. " The titles of all recipes that do not use butter."
 - `//recipe[not(../ingredient/@name="butter")]/title`
 - `//recipe[not(./ingredient/@name="butter")]/title [no!]`

XPath - recipes

1. "The titles of all recipes, returned as strings."
 - `fn:doc("recipes.xml")//title/text()`
2. " The titles of all recipes with less than 500 calories."
 - `//recipe[nutrition/@calories < 500]/title`
 - `//recipe[//@calories < 500]/title [no!]`
3. " The titles of all recipes that do not use butter."
 - `//recipe[not(./ingredient/@name="butter")]/title`
 - `//recipe[not(./ingredient/@name="butter")]/title [no!]`
1. "The amount of sugar needed for Zuppa Inglese."
 - `number(//recipe[title="Zuppa Inglese"]//ingredient[@name="sugar"]/@amount)`

XPath - recipes

1. "The titles of all recipes, returned as strings."
 - `fn:doc("recipes.xml")//title/text()`
2. " The titles of all recipes with less than 500 calories."
 - `//recipe[nutrition/@calories < 500]/title`
 - `//recipe[//@calories < 500]/title [no!]`
3. " The titles of all recipes that do not use butter."
 - `//recipe[not(./ingredient/@name="butter")]/title`
 - `//recipe[not(./ingredient/@name="butter")]/title [no!]`
1. "The amount of sugar needed for Zuppa Inglese."
 - `number(//recipe[title="Zuppa Inglese"]//ingredient[@name="sugar"]/@amount)`
5. The first two steps in preparing Zuppa Inglese."
 - `//recipe[title="Zuppa Inglese"]/preparation/step[position()=1 or position()=2]`

XPath - recipes

1. "The titles of all recipes, returned as strings."
 - `fn:doc("recipes.xml")//title/text()`
2. " The titles of all recipes with less than 500 calories."
 - `//recipe[nutrition/@calories < 500]/title`
 - `//recipe[//@calories < 500]/title [no!]`
3. " The titles of all recipes that do not use butter."
 - `//recipe[not(./ingredient/@name="butter")]/title`
 - `//recipe[not(./ingredient/@name="butter")]/title [no!]`
1. "The amount of sugar needed for Zuppa Inglese."
 - `number(//recipe[title="Zuppa Inglese"]//ingredient[@name="sugar"]/@amount)`
5. The first two steps in preparing Zuppa Inglese."
 - `//recipe[title="Zuppa Inglese"]/preparation/step[position()=1 or position()=2]`
6. "The titles of all recipes that requires five steps."
 - `//recipe[./step[5]]/title ??`
 - `//recipe[preparation/step[5]]/title !!`

XPath queries - recipes

7. "The number of recipes in the document."
8. "The average number of calories per recipe."
9. "The recipes that have an ingredient in common with Zuppa Inglese."
10. "The titles of recipes have some compound ingredients."
11. "The titles of recipes that have no compound ingredients."
12. "The titles of recipes that have only compound ingredients."

XPath solutions - recipes

7. "The number of recipes in the document."
 - `count(//recipe)`

XPath solutions - recipes

7. "The number of recipes in the document."

- `count(//recipe)`

8. "The average number of calories per recipe."

- `sum(//recipe/nutrition/@calories) div count(//recipe)`

XPath solutions - recipes

7. "The number of recipes in the document."
 - `count(//recipe)`

8. "The average number of calories per recipe."
 - `sum(//recipe/nutrition/@calories) div count(//recipe)`

9. "The recipes that have an ingredient in common with Zuppa Inglese."
 - `//recipe[.//ingredient/@name=//recipe[title="Zuppa Inglese"]//ingredient/@name]`

XPath solutions - recipes

7. "The number of recipes in the document."
 - `count(//recipe)`

8. "The average number of calories per recipe."
 - `sum(//recipe/nutrition/@calories) div count(//recipe)`

9. "The recipes that have an ingredient in common with Zuppa Inglese."
 - `//recipe[.//ingredient/@name=//recipe[title="Zuppa Inglese"]//ingredient/@name]`

10. "The titles of recipes have some compound ingredients."
 - `//recipe[ingredient/ingredient]/title`

XPath solutions - recipes

7. "The number of recipes in the document."
 - `count(//recipe)`

8. "The average number of calories per recipe."
 - `sum(//recipe/nutrition/@calories) div count(//recipe)`

9. "The recipes that have an ingredient in common with Zuppa Inglese."
 - `//recipe[.//ingredient/@name=//recipe[title="Zuppa Inglese"]//ingredient/@name]`

10. "The titles of recipes have some compound ingredients."
 - `//recipe[ingredient/ingredient]/title`

11. "The titles of recipes that have no compound ingredients"
 - `//recipe[not(ingredient/ingredient)]/title`

XPath solutions - recipes

7. "The number of recipes in the document."
 - `count(//recipe)`

8. "The average number of calories per recipe."
 - `sum(//recipe/nutrition/@calories) div count(//recipe)`

9. "The recipes that have an ingredient in common with Zuppa Inglese."
 - `//recipe[.//ingredient/@name=//recipe[title="Zuppa Inglese"]//ingredient/@name]`

10. "The titles of recipes have some compound ingredients."
 - `//recipe[ingredient/ingredient]/title`

11. "The titles of recipes that have no compound ingredients"
 - `//recipe[not(ingredient/ingredient)]/title`

12. "The titles of recipes that have only compound ingredients"
 - `//recipe[not(ingredient[not(ingredient)])]/title`

XPath - libreria

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE libreria SYSTEM "libreria.dtd">
<?xml-stylesheet href="libreria_trasf.xsl" type="text/xsl"?>
<libreria>
  <libri>
    <libro>
      <titolo>XML Programming</titolo>
      <prezzo valuta="EUR">25</prezzo>
      <autore>Mario Rossi</autore>
      <autore>Paolo Bianchi</autore>
    </libro>
    <libro>
      <titolo>Java Programming</titolo>
      <prezzo valuta="USD">30</prezzo>
      <autore>Paolo Bianchi</autore>
    </libro>
  </libri>
</libreria>
```

XPath - libreria

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!ELEMENT libreria (libri)>
```

```
<!ELEMENT libri (libro*)>
```

```
<!ELEMENT libro (titolo,prezzo,autore+)>
```

```
<!ELEMENT titolo (#PCDATA)>
```

```
<!ELEMENT prezzo (#PCDATA)>
```

```
<!ELEMENT autore (#PCDATA)>
```

```
<!ATTLIST prezzo valuta CDATA #REQUIRED>
```

XPath - libreria

- Dato il documento XML libreria.xml, scrivere le espressioni XPath per le seguenti selezioni:
 1. L'insieme di tutti gli elementi **libro**
 2. Il **titolo** del *secondo* libro dell'archivio
 3. I libri di **Paolo Bianchi**
 4. I coautori di Mario Rossi
 5. Il numero di libri scritti da Mario Rossi
 6. L'autore di più di un libro

Soluzioni XPath - libreria

- Soluzioni

1. `//libro`

1. `//libreria/libri/libro[2]/titolo/text()`

1. `//libro[.//autore = "Paolo Bianchi"]/titolo`

1. `//libro[.//autore = "Paolo Bianchi"]/autore[not(. = "Paolo Bianchi")]`

1. `count(//libro[.//autore = "Paolo Bianchi"])`

1. *Si può fare (come?) ma è meglio usare XQuery..*

Credits

- Questo materiale è basato su dispense ed esercizi di
 - Ombretta Gaggi @ Università di Padova
 - Alessio Pace, Valter Crescenzi, Paolo Merialdo @ Università Roma Tre