

# In-class Exercises: XPath and XQuery

Given the example XML file **quiz.xml**, find the following:

1. Get the solution to every question (regardless of the question type).

**Solution:**

```
fn:doc("quiz.xml")/quiz/questions/*/@solution
```

**or**

```
fn:doc("quiz.xml")/quiz/questions//@solution
```

2. Get the elements of the true-false questions only.

**Solution:**

```
fn:doc("quiz.xml")/quiz/questions/tf-question
```

3. Get the text of the multiple-choice (mc) question which has the QID 'Q888'.

**Solution:**

```
fn:doc("quiz.xml")/quiz/questions/mc-question[@qid="Q888"]
```

–see what happens when you remove @

–This returns the entire mc-question element. Now get the *text* of the question:

```
doc("quiz.xml")/quiz/questions/mc-question[@qid="Q888"]/question/text()
```

4. For the student with the ID s555555555, list the *second* response answer recorded for them.

**Solution:**

```
fn:doc("quiz.xml")/quiz/class-responses/student[@sid="s555555555"]/response[2]/@answer
```

5. Now return that answer to question 4, enclosed by a <secondAns></secondAns> tag.

**Solution:**

```
<secondAns>fn:doc("quiz.xml")/quiz/class-responses/student[@sid="s555555555"]/response[2]/@answer </secondAns>
```

6. Write an XQuery FLWOR expression that returns the text for each multiple-choice question in quiz.xml, along with the *number* of options available for that question. Hint: you can use **count(option)**.

**Solution:**

```
for $q in doc('quiz.xml')/quiz/questions/mc-question
```

```
return ($q/question/text(), $q/count(option))
```

And as we noticed when running the demo in class, using count() this way also works:

```
return ($q/question/text(), count($q/option))
```