newtFire {dhlds}

Authored by: Nicole L. Lottig (nll29 at pitt.edu), Brooke A. Stewart (bas160 at pitt.edu), and Rebecca Parker (rjp43 at pitt.edu | Twitter: @bcpkr396)
Edited and maintained by: Elisa E. Beshero-Bondar (ebb8 at pitt.edu)

Last modified: Sunday, 06-Aug-2017 18:16:14 EDT. Powered by firebellies.

Schematron Exercise 1

Preliminaries

Before beginning this assignment, please thoroughly read our introduction to Schematron. This tutorial will be useful to you during this assignment and the Schematron Exercise 2. To begin this assignment, you will need to open a new Schematron document in <oXygen/> under **File → New → New Document → (scroll to Schematron in the alphabetized list) → Schematron**. Once opened, you will keep the default xml line at the top, but you will delete everything from `<sch:schema>` down. You will then replace this with:

```
<schema xmlns:sch="http://purl.oclc.org/dsdl/schematron" queryBinding="xslt2"
    xmlns:sqf="http://www.schematron-quickfix.com/validator/process"
    xmlns="http://purl.oclc.org/dsdl/schematron">


        </schema>
```

You will be writing your Schematron **inside** the `<schema>` root element.

Analysis of the task

Background:

For this assignment, we are looking at votes for what place the Pitt-Greensburg DH Class will go for Spring Break. The options include: New York City, Mexico, London, and Rome. Each place gets between 0% to 100% of the votes. Assume here that this is the final voting poll, and there are no other options. This means that when you add the four percentages together, the result must be exactly 100%. Also assume that this is recording the already calculated percentage of the votes, not the raw count of the votes. All of these percentages are to be integer values.

Here is a Relax NG schema for the results of the Spring Break votes:

```
start = results
results = element results {place+}
place = element place {name, xsd:int}
name = attribute name {"NYC" | "Mexico" | "London" | "Rome"}
```

Here is a sample XML document that is valid against the above schema:

```
<results>
    <place name="NYC">34</place>
    <place name="Mexico">24</place>
    <place name="Rome">30</place>
    <place name="London">12</place>
</results>
```

Our Relax NG schema is a little sloppy and doesn't constrain the XML as thoroughly as it could have been better written (as we will discuss below). It lets us set a rule that the content of the element `<place>` must be a number (or xsd:int for integer), but the rule isn't really good enough as we will see from the from the following example:

```
<results>
<place name="NYC">27</place>
<place name="Mexico">39</place>
<place name="Rome">12</place>
<place name="London">15</place>
</results>
```

Do you see the problem? The four percentage values only total 93%! No matter how good our coding is, it is not possible to keep this type of error from happening by using Relax NG alone. That is why we use Schematron.

Task:

First, re-create the Relax NG schema file and the XML document by copying and pasting the **blue sample code above** into files with the appropriate file extensions. Associate your newly created Schematron and the Relax NG schema with your XML. As you write the following rules, "munge" (aka mess up) the XML to verify your rules are firing by entering correct and incorrect values into the XML.

1. Write a Schematron rule that verifies the four percentages always equal 100%.
2. Write a Schematron rule that fires an error when any location's voting percentage sits outside of the 0 to 100 range. There should be no negative integers and no integers greater than 100. (Hint: the Relax NG schema states that these values must be integers, so you will not have to worry about making sure of that; however, the computer parser will not recognize the values in each `<place>` as integers and instead will try to process them as strings of text. Use the `number()` function so the computer parses the values as numbers.)
3. Write a Schematron rule that tests there are only ever four place elements in our list of locations to visit for Spring Break.
4. Write a Schematron rule that tests if any of the `@name` values are repeated. It should not be possible for there to be any places that appear more than once in the XML. (Hint: Think about using the `count()` function for this. How many different values for `@name` should there be? How would you make sure each value is not repeated?

Optional Task:

Write a Schematron rule that tests whether the places are listed in order from greatest to least number of votes. (Hint: You will need to check the **numerical** value of each place with their **sibling** place's **numerical** value. Depending on your rule context you may need to clarify the position of the immediate sibling using the [1] position notation.)

Submission:

Upload your completed Schematron schema and your re-created XML document (**with your associated Schematron line**) on Courseweb. Please follow our standard filenaming conventions for homework assignments uploaded to Courseweb.