

TC 7.0 xml

XML is another markup language, but where html is all about hypertext or linking pages together, xml is called "extensible markup language" meaning it creates a container or format for data to be presented.

You have already been using this whenever you create a word or excel document, that's why they are saved as .docx or .xlsx

The "x" is xml, meaning the document is a container, your words are the data.

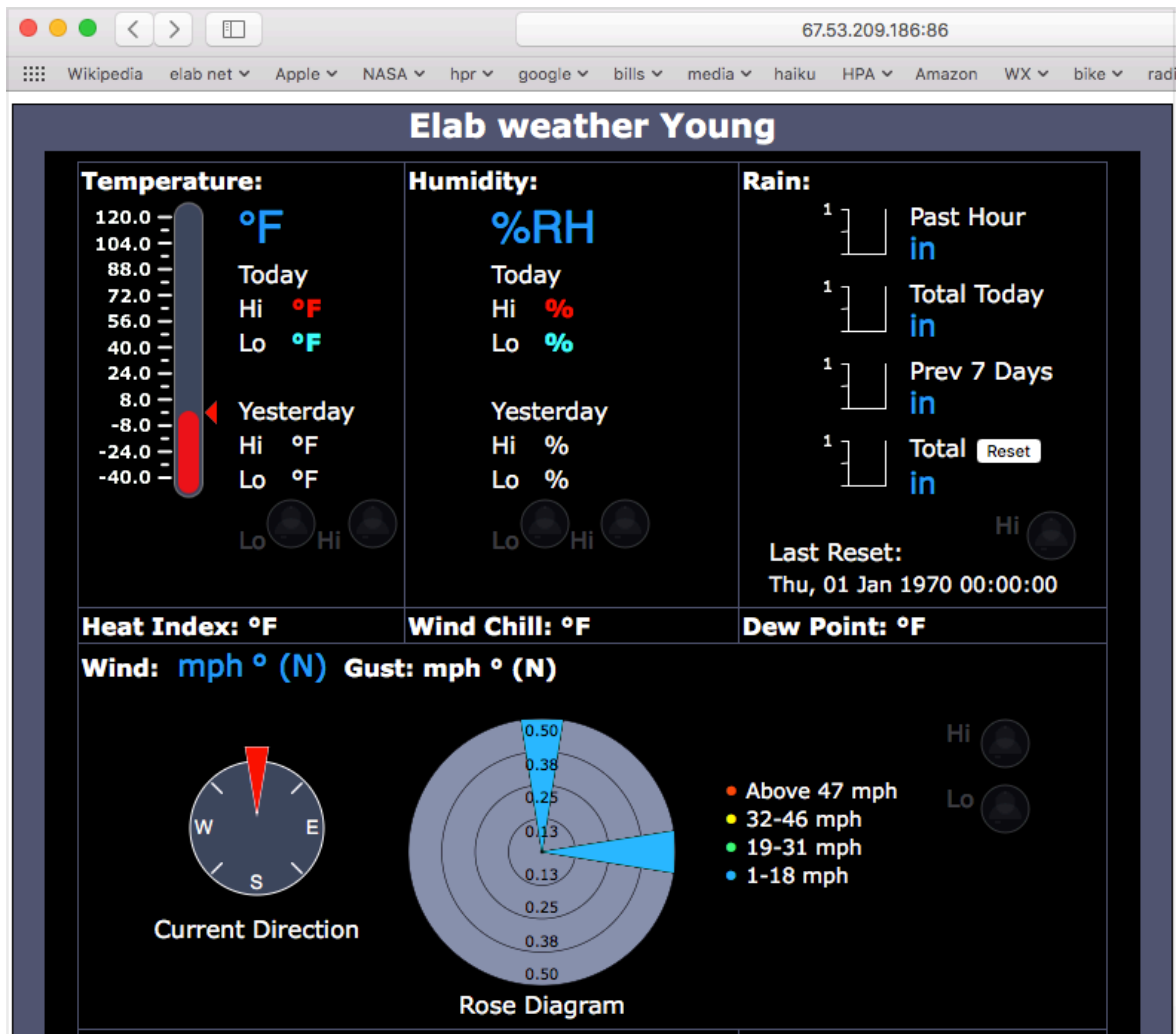
Here's an example:

Go to <http://67.53.209.186:86> on your browser

This is one of our weather stations here at the energy lab, using port forwarding (remember that?) to port inbound port 86 to the actual port 80 on the machine at 10.14.62.4, which is inside our network.

You could also have used the vpn to get into the 10.14.x.y network, then just gone to 10.14.62.4. or 10.14.62.5

You should see something like this:



If you then enter:

<http://67.53.209.186:86/state.xml>

you are now asking this device to spit out its data in xml format:

```
<datavalues>
  <windSpd>22.72</windSpd>
  <windDir>60.81</windDir>
  <rainTot>0.00</rainTot>
  <temp>67.1</temp>
  <humidity>58</humidity>
  <solarRad>1049.02</solarRad>
  <barPressure>--</barPressure>
  <aux1>--</aux1>
  <aux2>--</aux2>
  <rain1h>0.00</rain1h>
  <rainToday>0.00</rainToday>
  <rain7d>--</rain7d>
  <rainRst>1588598489</rainRst>
  <rainAlrm>0</rainAlrm>
  <tempH>67.6</tempH>
  <tempL>67.1</tempL>
  <tempHY>--</tempHY>
  <tempLY>--</tempLY>
  <heatIndex>--</heatIndex>
  <windChill>--</windChill>
  <dewPoint>51.8</dewPoint>
  <tempAlrm>1</tempAlrm>
  <humidityH>59</humidityH>
  <humidityL>57</humidityL>
  <humidityHY>--</humidityHY>
  <humidityLY>--</humidityLY>
  <humidityAlrm>2</humidityAlrm>
  <presN1>--</presN1>
  <presN3>--</presN3>
  <presN6>--</presN6>
  <presN9>--</presN9>
  <presN12>--</presN12>
  <presN15>--</presN15>
  <presN24>--</presN24>
  <presAlrm>2</presAlrm>
  <windGust>22.86</windGust>
  <windGustDir>50.82</windGustDir>
  <windAlrm>1</windAlrm>
</datavalues>
```

Why is this important?

It gives you containers, which can then be parsed or split out to do neat stuff with that data.

But wait:

What if I want to control something?

Easy peasy:

Go to <http://67.53.209.186:89>

You should see this:

elab-fans		
NW fan	fan OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
NE fans	fan OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
SE fan	Fan OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
SW fan	Fan OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
Relay 5	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
Relay 6	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
Relay 7	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
Relay 8	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
Relay 9	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
Relay 10	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>

Open a new window and type this:

<http://67.53.209.186:89/state.xml?relay1State=1>

Now watch the fan page...

Congratulations, you have remotely turned on the fans in the elab.

Now turn them off...

<http://67.53.209.186:89/state.xml?relay1State=0>

Why is this important?

Check this out:

<https://en.wikipedia.org/wiki/SCADA>

Look down about half way on the page:

- Industrial processes include manufacturing, Process control, power generation, fabrication, and refining, and may run in continuous, batch, repetitive, or discrete modes.
- Infrastructure processes may be public or private, and include water treatment and distribution, wastewater collection and treatment, oil and gas pipelines, electric power transmission and distribution, and wind farms.
- Facility processes, including buildings, airports, ships, and space stations. They monitor and control heating, ventilation, and air conditioning systems (HVAC), access, and energy consumption.

Here's where you come in:

SCADA is notoriously insecure, running in most cases on windows DOS, windows 95 or windows XP, none of which are secure.

xml to the rescue!

This is the language you will use to replace the insecure SCADA system.

Siemens in Germany is already doing this, developing among other things a smart energy grid:

<https://new.siemens.com/global/en/products/energy/energy-automation-and-smart-grid.html>

If you want to make buckets of money, working on something really cool, this could be it.

——end of module 7 xml——next: html links——