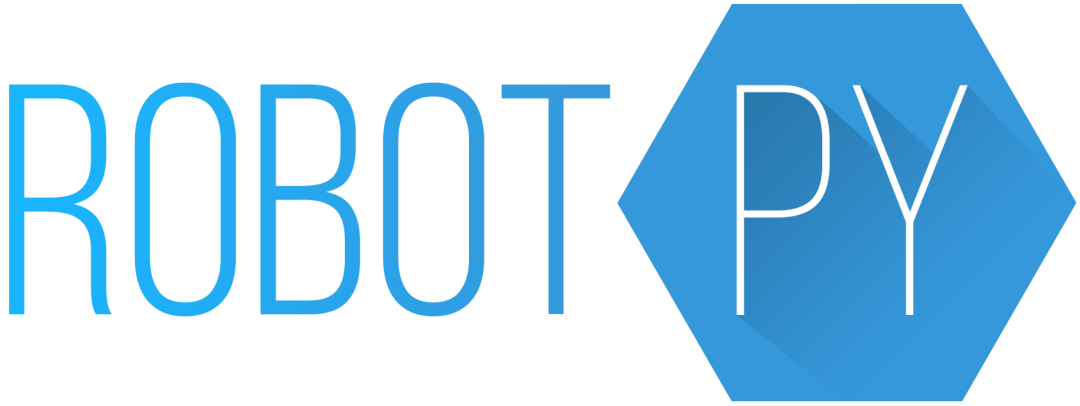


Driver Station Dashboards Using HTML + Javascript

Dustin Spicuzza
September 10, 2016

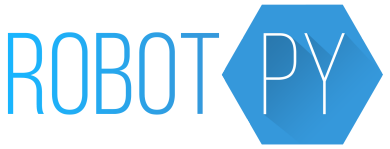
NE FIRST University Day

Agenda

ROBOT PY

- History + Alternatives
- Why HTML + Javascript
- How it works
- Getting started
- Real dashboard code
- Demos

Intended Audience



- Some familiarity with a browser
- Some programming experience
  - If you don’t… that’s ok too

Who Am I?

ROBOTPY

- Software engineer at BBN Technologies
- Mentoring FRC since 2009
- Co-maintainer of RobotPy since 2010
- My team's UI-related awards:
  - 2012 Boston Regional; Innovation in Control
  - 2013 Boston Regional; Innovation in Control
  - 2014 Virginia Regional; Industrial Design
  - 2015 Greater DC Regional; Innovation in Control
  - 2016 Chesapeake Champs; Innovation in Control

Who Am I?

ROBOTPY

- Software engineer at BBN Technologies
- Mentoring FRC since 2009

Dashboards

ROBOTPY

- Back in the day…
- Lots of options now
  - LabVIEW Dashboard
  - SmartDashboard/SFX
  - Custom (Java, C#, et al)

Dashboard Applications

ROBOTPY

- Robot Control by secondary operator
  - Click a button, make something happen
- Set up autonomous mode
  - Makes it easy to switch between different mode settings on the fly

Dashboard Applications

ROBOTPY

- Tuning the robot
  - Changing positions dynamically
  - Better than constant code redeploy!
- Providing visual feedback for sensors
  - Good for debugging hardware issues

Dashboards

MY JOURNEY

My Journey: 2009/2010

ROBOTPY

WebDMA (C++)

My Journey: 2009/2010

ROBOTPY

WebDMA (C++)

- Not actually a dashboard
- Ran an in-process webserver on the cRIO
  - UI was HTML/JS
- Great for tuning

My Journey: 2012

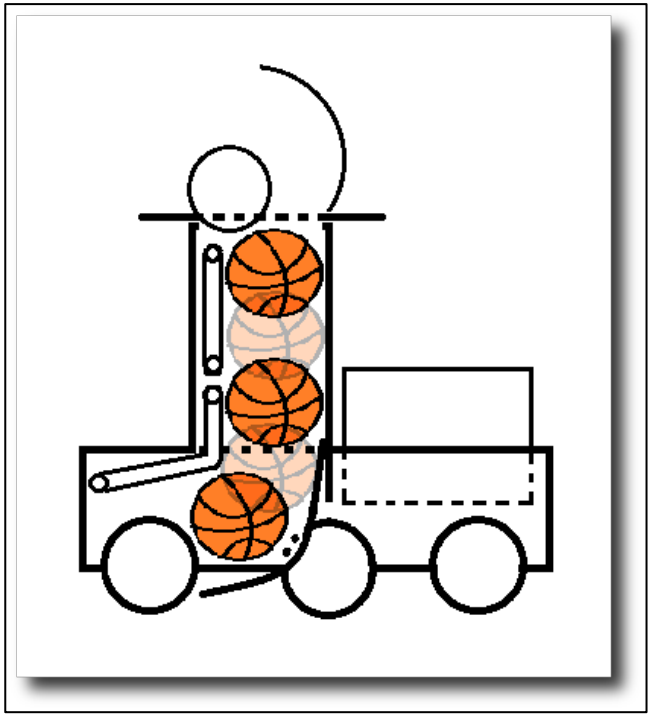
ROBOTPY

Custom SmartDashboard widget (Java)

- Show ball when sensor is on

But…

- Little/poor documentation
- Java swing is complicated
- Layout is terrible
- Hard to setup



My Journey: 2013/2014

ROBOTPY

PyGTK + touchscreen laptop

My Journey: 2013/2014

ROBOTPY

PyGTK + touchscreen laptop

- Very shiny
- Integrated with image processing
- Communications via pynetworktables

But…

- Python is great, GTK is not
- Difficult to customize interface
- Very difficult to teach

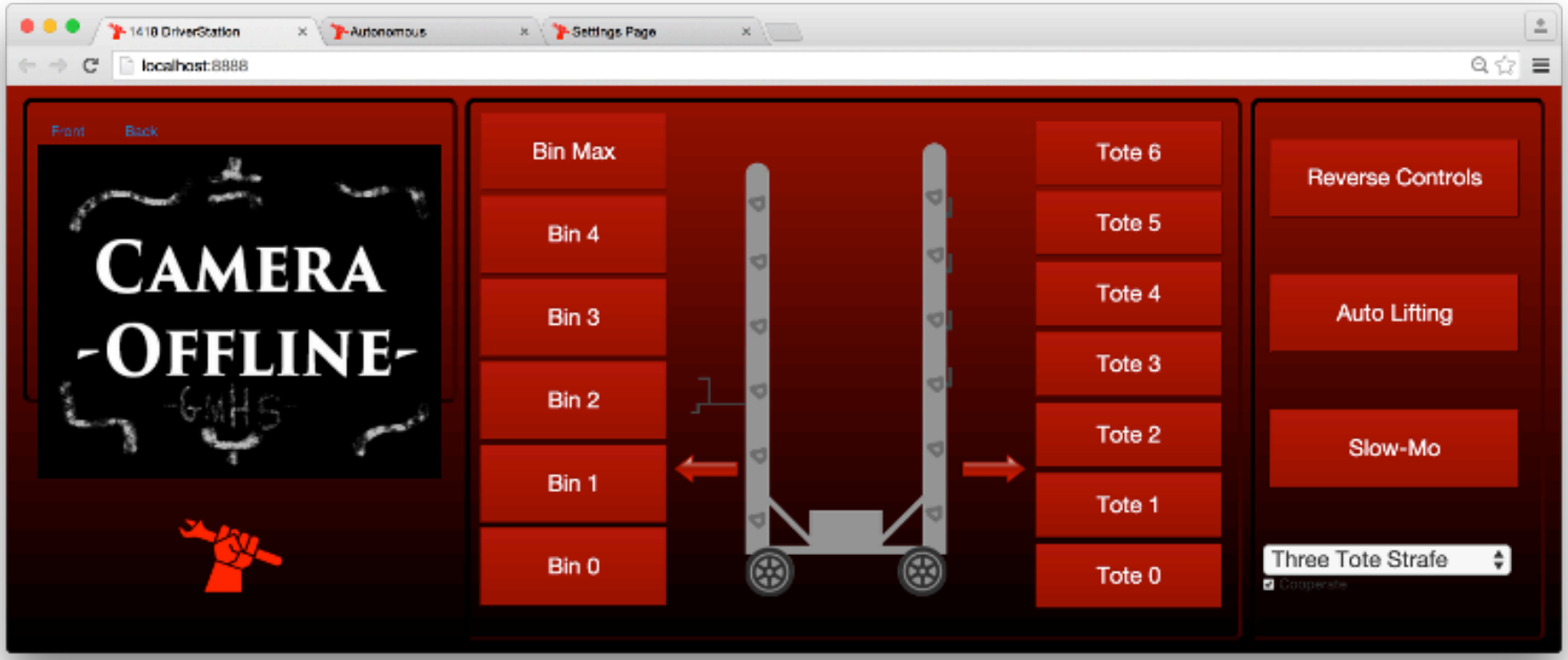
Introducing

PYNETWORKTABLES2JS

My Journey: 2015 - Present

ROBOTPY

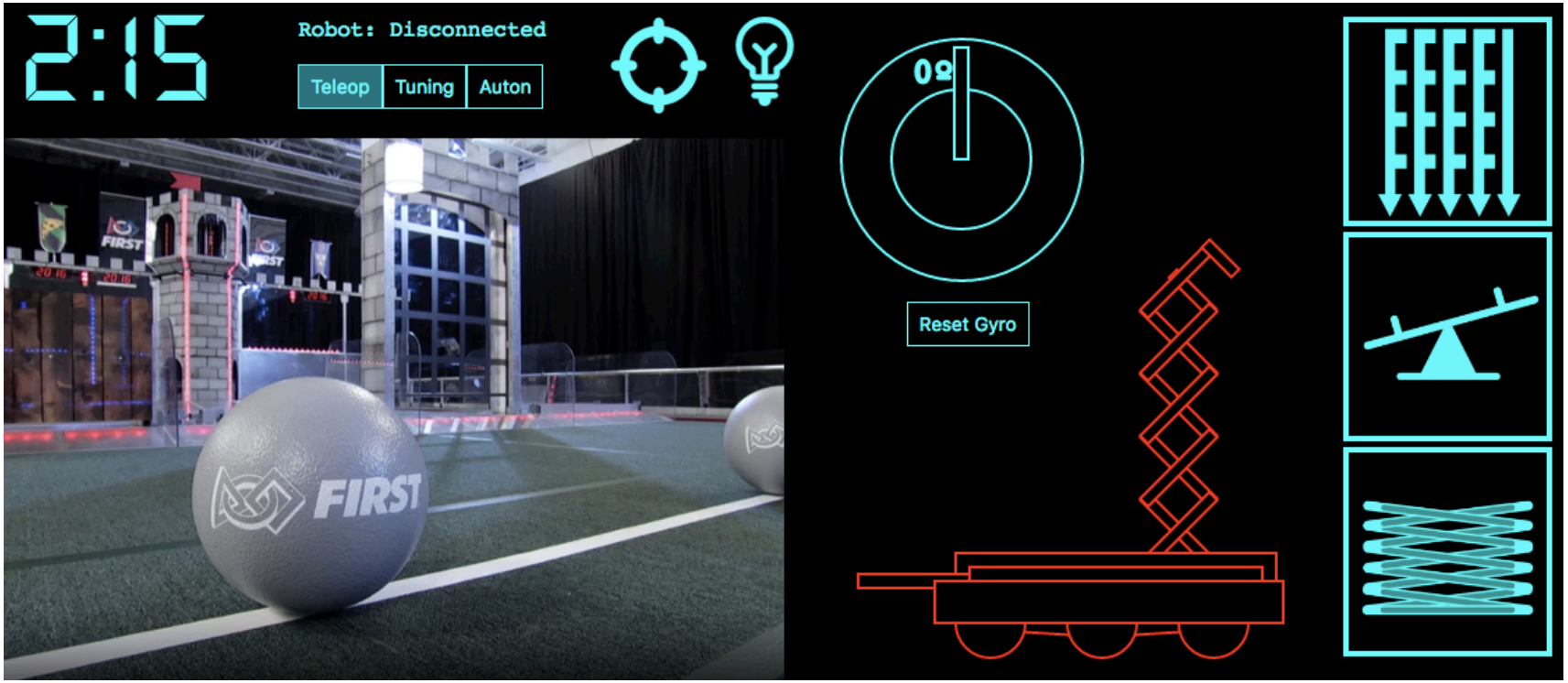
2015 UI (Team 1418)



My Journey: 2015 - Present

ROBOTPY

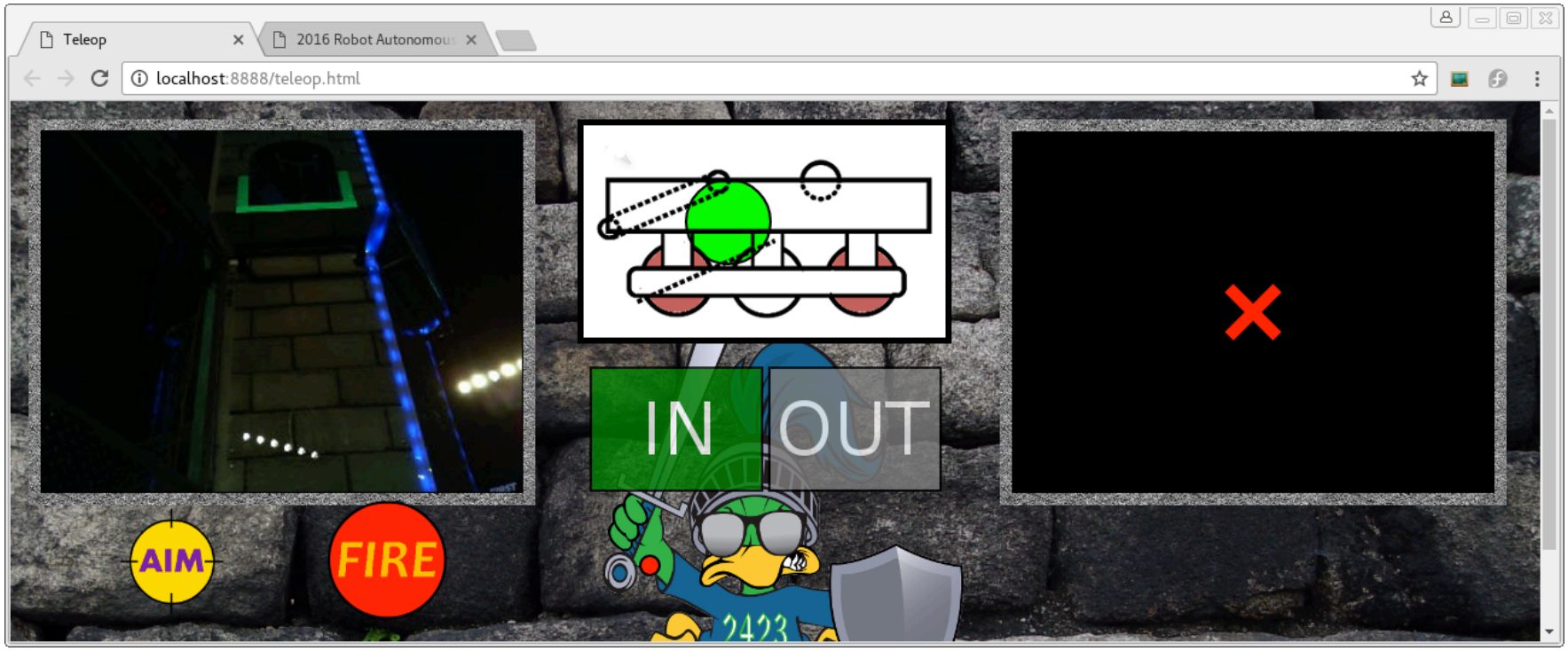
2016 UI (Team 1418)



My Journey: 2015 - Present

ROBOTPY

2016 UI (Team 2423)



My Journey: 2015 - Present

ROBOTPY

Interactive camera threshold tuning

Why pynetworktables2js

TL;DR: It's simpler

pynetworktables2js lowers the barrier of entry for teams that want an additional way to tune/control their robot with a minimal amount of programming.

Why pynetworktables2js

ROBOTPY

TL;DR: It’s simpler

It’s as easy as making a webpage.

Why pynetworktables2js

ROBOTPY

- Many students already know how to create webpages
  - And if not, there's TONS of ways to learn on the web
- Outreach to nontraditional students
  - Introduce students to programming concepts
  - Artistic students
- Yet another way to learn real world skills

Why pynetworktables2js

ROBOTPY

- Very flexible
  - Anything you can do in a webpage
- Simple to get started
- Cross platform
  - Windows + OSX + Linux
- Actually has documentation
  - Unlike most other dashboard alternatives

Let’s get started

HOW IT WORKS

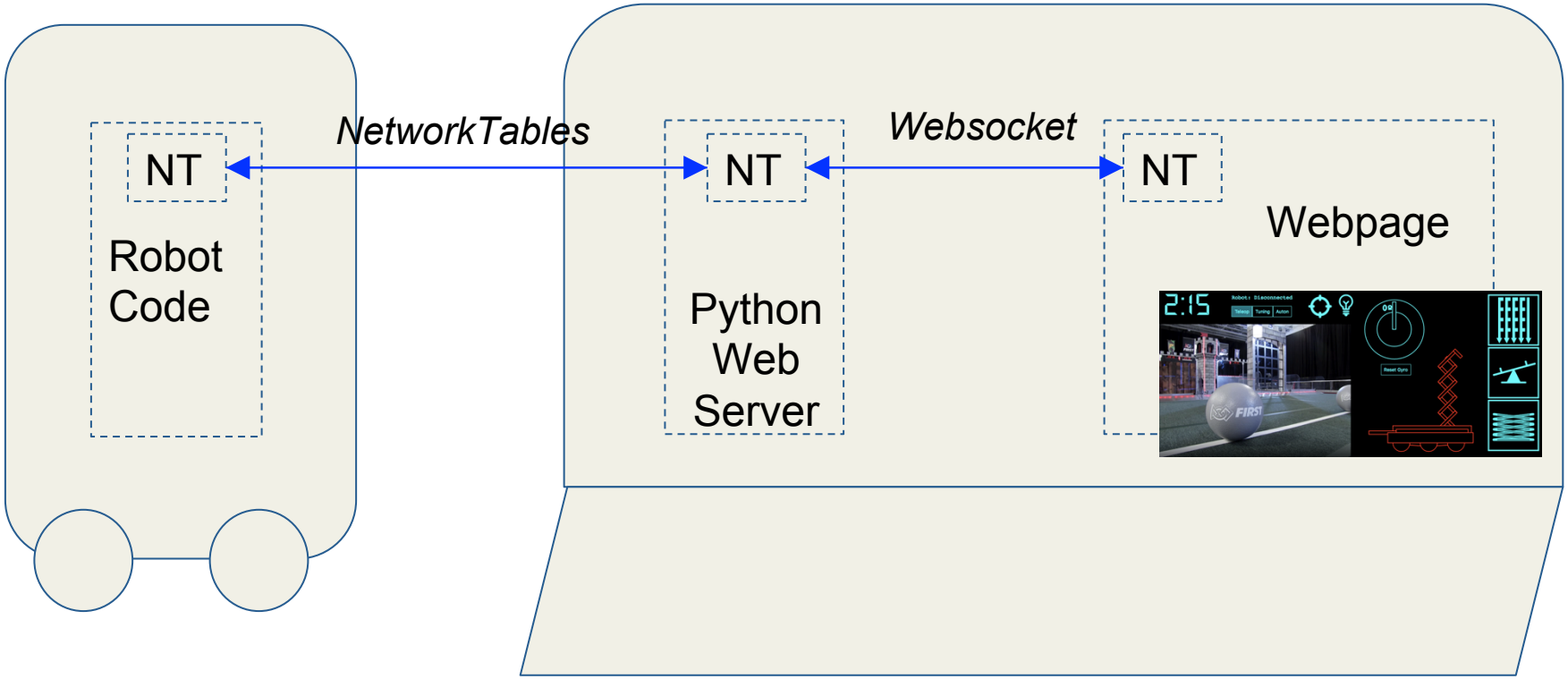
How it works

ROBOTPY

- Browser talks to Python server running on your laptop
- Python server uses NetworkTables to communicate with Robot
  - Results are passed back/forth to webpage

How it works

ROBOTPY



How it works

ROBOTPY

- Note that the webserver runs on the laptop
- You could run it on the robot
  - I don't recommend this because of bandwidth reasons
  - Also, it means you can't load the page until the robot connects to FMS

NetworkTables

ROBOTPY

- Easy to use distributed key-value protocol created by WPILib team
- Very little setup required*
- Latency is “good enough” for 99% of teams
  - pynetworktables is 50ms max

* mDNS issues in 2016, hopefully they’ll be fixed in 2017…

NetworkTables Basics

ROBOTPY

On the server…

```
table.putString(“key”, “something”)
```

On the client…

```
x = table.getString(‘key’)

// output is ‘something’
printf(“%s\n”, x);
```

pynetworktables2js Basics

ROBOTPY

- Very similar to other languages
  - getValue/putValue only
  - Doesn't need getNumber/getString/etc
  - Uses full path of keys, no tables

```
// getValue will return the correct type
var coolThing = NetworkTables.getValue("/table/key");

// putValue doesn't care about what type you pass
NetworkTables.putValue("/table/string_key", "my value");
NetworkTables.putValue("/table/bool_key", true);
NetworkTables.putValue("/table/number_key", 42.0);
```

Let’s get started

GETTING STARTED

Getting Started

ROBOT PY

- Install pynetworktables2js
  - See the documentation
- Create a directory with web content
- Run the server from inside the directory

```
On Windows:
py -m pynetworktables2js

On Linux/OSX:
python -m pynetworktables2js
```

- Point your browser at http://localhost:8888

Your first webpage

ROBOT PY

```
<!DOCTYPE html>
<html lang="en"><body>

Connected: <span id="connected">false</span>

<!-- Include the API -->
<script src="/networktables/networktables.js"></script>

<script type="text/javascript">

function onConnected(connected) {
  document.getElementById("connected").textContent = connected;
}

NetworkTables.addRobotConnectionListener(onConnected, true);

</script>
</body>
</html>
```

pynetworktables2js API

NetworkTables.addRobotConnectionListener(f)

- Calls a function to tell you that the page is connected to the robot
  - Driver wants to know if the dashboard isn't going to actually do anything!

pynetworktables2js API

ROBOT PY

NetworkTables.putValue(k,v)

- Sets the value in NetworkTables
  - Give it ~50ms to get to the Robot
- If not connected to robot, value may be discarded

pynetworktables2js API

ROBOTPY

NetworkTables.getValue(k, [default])

- Returns the current value of a key or null
- I rarely use getValue
  - What happens if the value changes?
  - How do you find out about the change?

- If not connected to robot, value isn't always meaningful

pynetworktables2js API

ROBOTPY

NetworkTables.addKeyListener(k, f, [imm])

- Set a function that will be called whenever the value for a particular key is changed.
  - It will usually call you when the robot connects, as the values will change
  - If you set the third parameter to true, it will call you immediately with the current value

Putting it all together

REAL DASHBOARD CODE

Putting it all together

ROBOTPY

- Let’s show how to display simple feedback of a digital input
- Here’s the steps
  - Create an SVG circle element
  - Add a listener for the value being sent over NetworkTables
  - Change color when the value is received

Putting it all together

ROBOTPY

1. Include the API in your page

```
<!DOCTYPE html>
<html>
<head>
  <script src="/networktables/networktables.js"></script>
</head>
```

Putting it all together

ROBOT PY

2. Create an SVG circle element

```
<svg width="100" height="100">
   <circle id="switch" cx="50" cy="50" r="40"
           stroke="black" stroke-width="4"
           fill="gray" />
</svg>
```

Putting it all together

ROBOTPY

3. Add a listener for the value being sent over NetworkTables

```
<script type="text/javascript">

function onSwitchChanged(key, value, isNew) {
  // TODO
}

NetworkTables.addKeyListener('/SmartDashboard/switch',
                             onSwitchChanged, true);

</script>
```

Putting it all together

ROBOT PY

4. Change color upon new value
   - ***Hint:*** Use jQuery to make it easier :)

```
function onSwitchChanged(key, value, isNew) {
  if (value) {
    $("#switch").css("fill", "red");
  } else {
    $("#switch").css("fill", "green");
  }
}
```

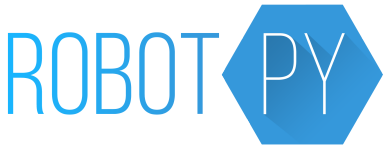
Putting it all together

DEMO

Putting it all together

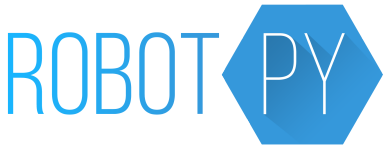
2015 UI DEMO

pynetworktables2js API



- There's lots more...
  - Read the documentation

pynetworktables2js API



- Useful helper functions for common UI tasks
  - Easy toggle button
  - HTTP Camera support
  - Robot connection indicator
  - SendableChooser related helpers

pynetworktables2js tips

ROBOTPY

- Tell your team to buy a touchscreen laptop
  - Very inexpensive these days (<$700)
- Create multiple pages, and show them using browser tabs
  - One page for teleop
  - One page for autonomous options
  - One page for tuning

Future work

ROBOTPY

- pynetworktables2js is a good start…
  - Need more widgets/helpers
  - Single executable deploy for 2017
  - Would love more contributors (hint, hint)
- Need a ‘TableViewer’ integrated into pynetworktables2js
  - There’s a PR on github, needs work

Future work

ROBOTPY

- NetworkTables 3.0 support for 2017
  - Enables persistent settings
  - Can talk to LabVIEW
  - Will probably happen…

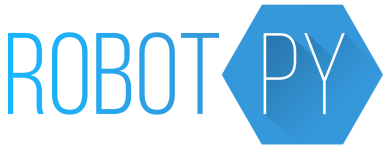
Want More?

ROBOTPY

- This presentation + code available online*
  - https://github.com/virtuald/frc-dashboard-workshop

* Sometime on Sunday

FRCDashboard Project



- Created by Erik Boesen, student on 1418
- Uses Electron framework to run HTML/JS app
  - Same thing that Atom text editor uses
- Goal is to provide easy customization and expansion using prebuilt modules
- Project is still very new

Resources

ROBOTPY

- pynetworktables2js
  - Code + Examples: https://github.com/robotpy/pynetworktables2js
  - Documentation: http://pynetworktables2js.readthedocs.io

Resources

ROBOTPY

- 1418's UI repos
  - https://github.com/frc1418/2015-ui
  - https://github.com/frc1418/2016-ui
- FRCDashboard
  - http://frcdashboard.github.io/

One more thing…

ROBOTPY

- Publicly editable repository of information related to FIRST Robotics
  - Technical topics
  - Non-technical
  - Team pages
- Add content to your team's page!

https://firstwiki.github.io

Questions

ROBOT PY

???