

Web Apps

What's (instinctively) next?
The future of Web APIs

Json

- for XML
- uses JS syntax, you can create JS objects for powerful functionality
- Works with PHP and MySQL server-side

http://www.w3schools.com/json/tryit.asp?filename=tryjson_server_sql_style

Angular JS

- Works within HTML elements by creating new attributes
- powerful tool for templates:
 - `<script type="text/ng-template" id="templateId.html"> </script>`

Inspect Pasha's website (your fellow classmate):

<http://deepblue.cs.camosun.bc.ca/~cst401/comp140/lab7/#/>

<https://docs.angularjs.org>

Ajax

- Asynchronous JavaScript and XML
- talk to the server without a refresh
- not a programming language

```
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```

http://www.w3schools.com/ajax/ajax_xmlhttprequest_send.asp

Polymer - Google

- library with api web-components
- less work for better results
- custom elements in HTML simplifying the manipulation of the DOM!

Check it out!

<https://www.polymer-project.org>

Ruby on the Rails - Basecamp

- powerful programming language built off of Ruby to build web applications.

<http://railsapps.github.io/rubyonrails-nitrous-io.html>

- “Maz” (mid-90s) created a scripting language more powerful than Perl and more OO than Python.
- Rails is a web application development framework – but more simply a gem
- Intuitive, flexible and extensive and INTERPRETIVE

http://guides.rubyonrails.org/getting_started.html

And there's so much more...