Welcome to:

# Data Validation in Web Applications

# Hi! I'm Scott Kirkland

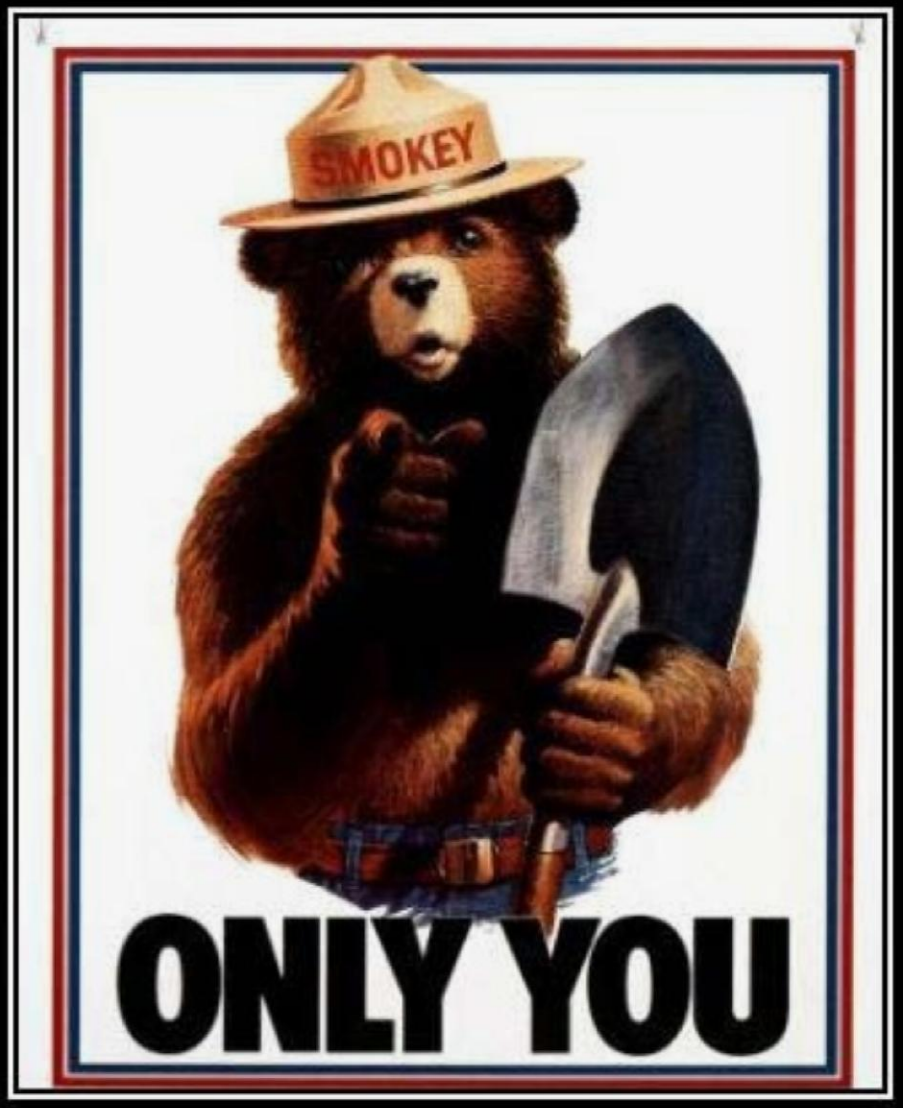
- Slinging code @ UCDavis for 8 years
- Currently Sr. Application Architect
  - for the College Of Agricultural and Environmental Sciences Dean's Office
- Open source coder
  - https://github.com/srkirkland
    - MvcHtml5, Data Annotations Extensions, ITSecuritySymposium
  - https://github.com/ucdavis
    - UCDArch, Web Applications
- Co-founded the local .NET User Group

Welcome to:

# Data Validation in Web Applications

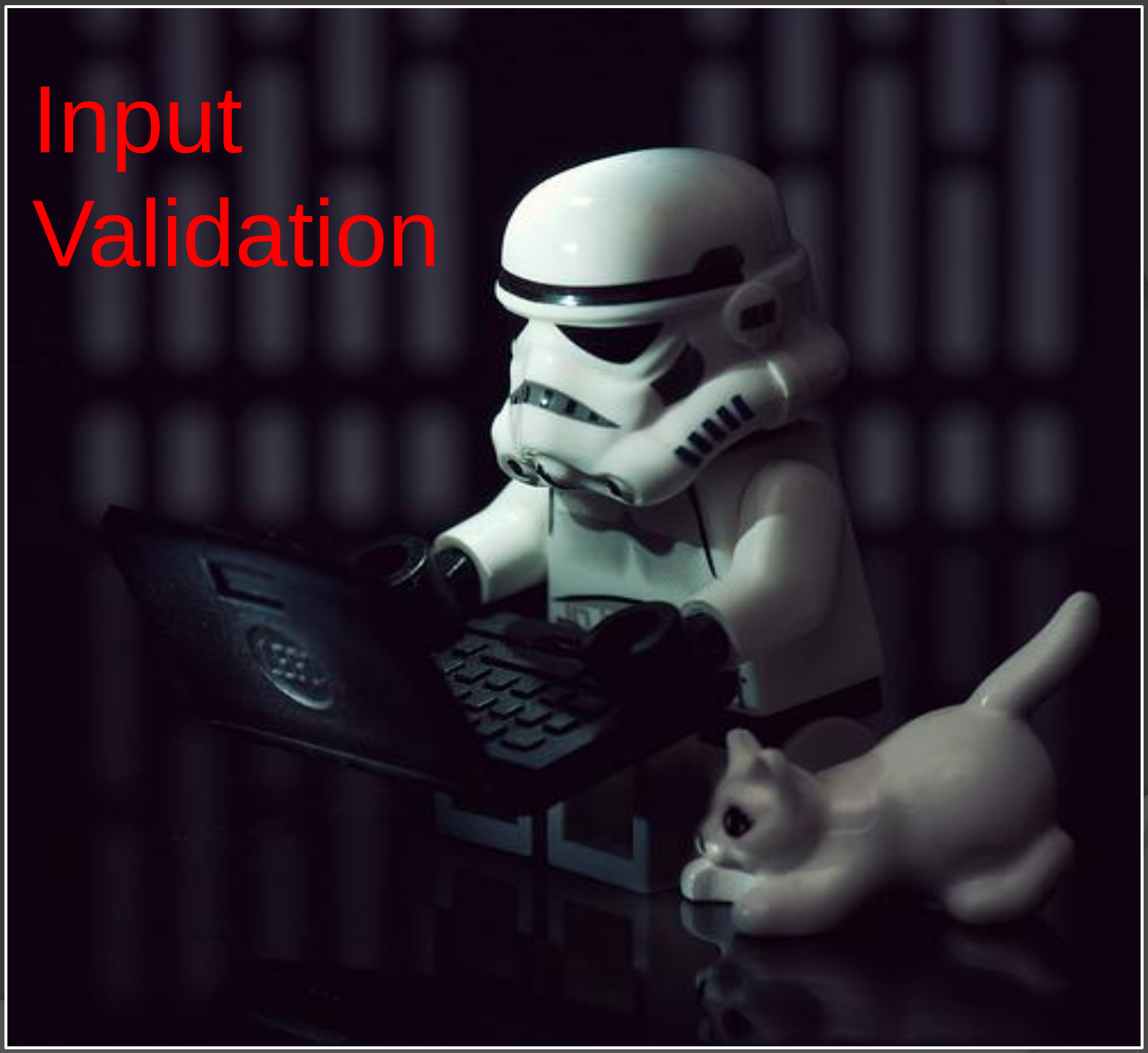
The most common web application security weakness is the failure to properly validate input from the client or environment.

- OWASP[1]




# Can Prevent Bad Data

# Input Validation



# Input Validation in Web Forms

Ensure user supplied data is

- Strongly typed
- Correct syntax
- Within length boundaries
- Contains only permitted characters
- Or that numbers are correctly signed and within range boundaries
- Is “business rule correct”

# Client Side Validation

Validate data on the client first

- Provides better feedback to the end user
- Makes your site feel more responsive

Always validate on server-side as well!

# JavaScript Validation

- “Current” solution, useful & widely supported (Probably about 95%)
- Any JavaScript errors and validation disappears
- Fairly difficult to implement, though libraries help[3]

# Input Validation w/ HTML5

# HTML5 Input Validation

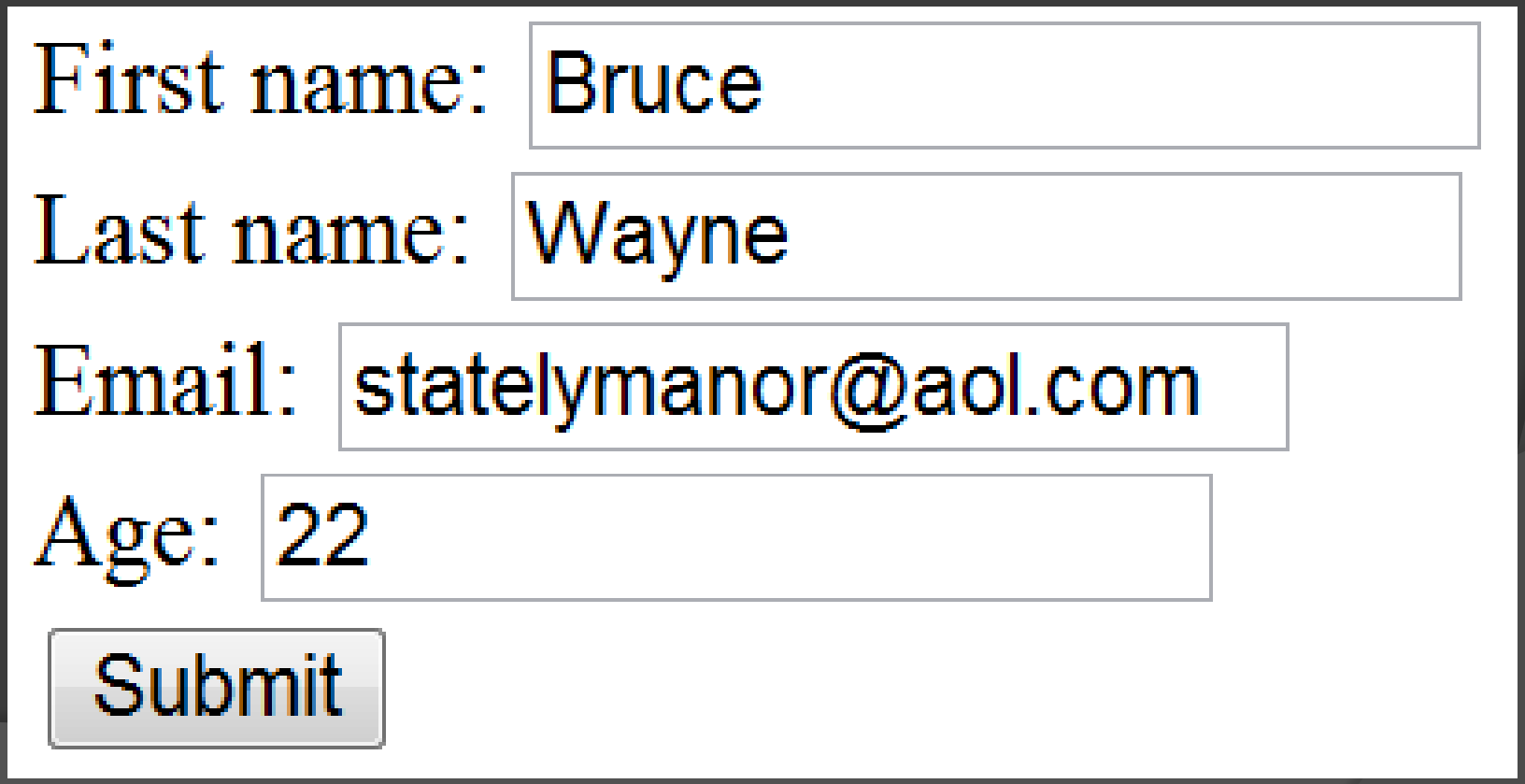
Two major form validation innovations

1. New Input Types
2. Constraint Validation

# HTML5 Input Types

New input types were added to augment

<input type="text" />



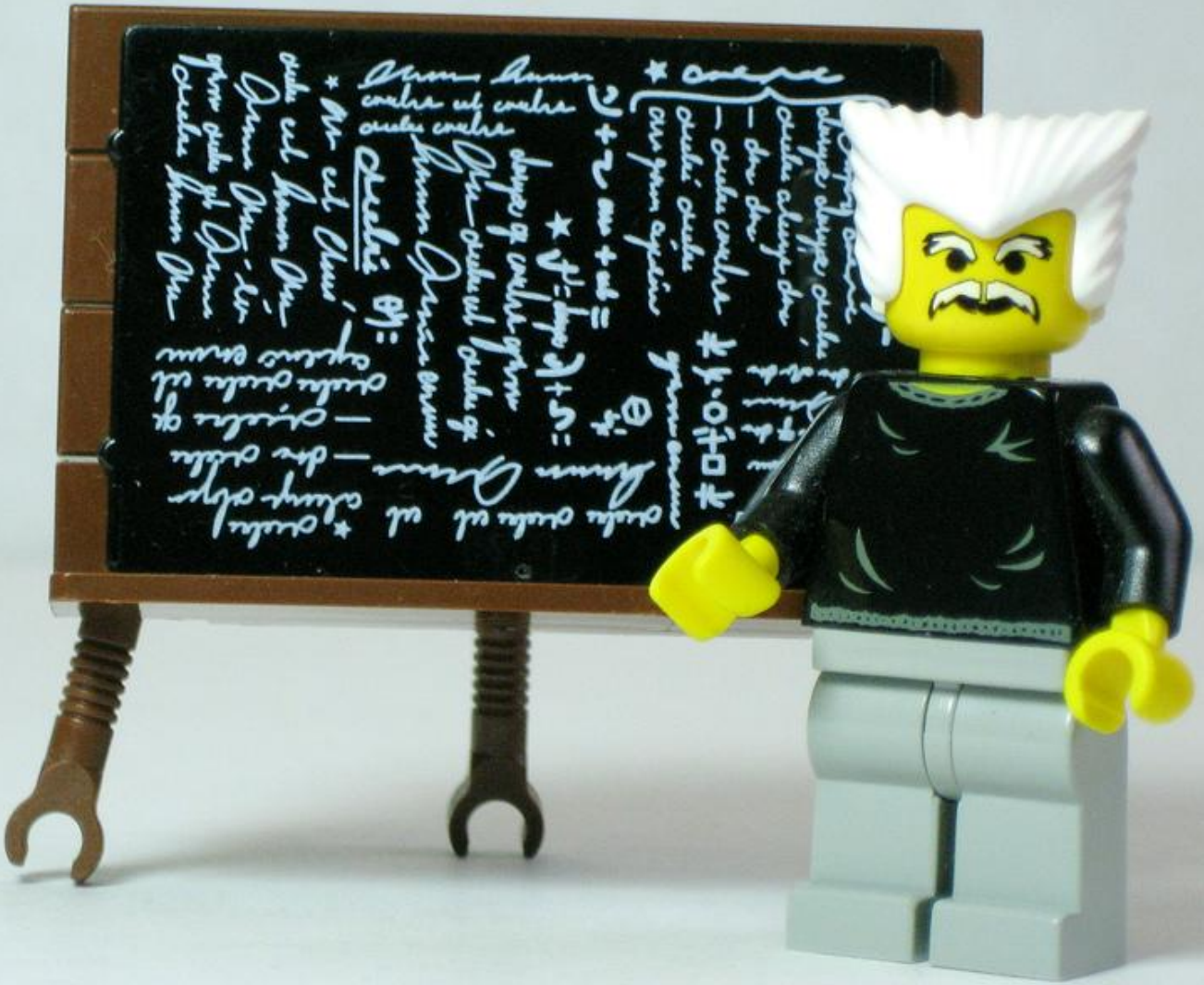
# HTML5 Input Types

- search
- tel
- url
- email
- number
- range
- color
- datetime
- date
- month
- week
- time
- datetime-local

# HTML5 Input Types

- Gives semantic meaning to your forms
- Enable behaviors based on input type

# DEMO: Html5 Input Types

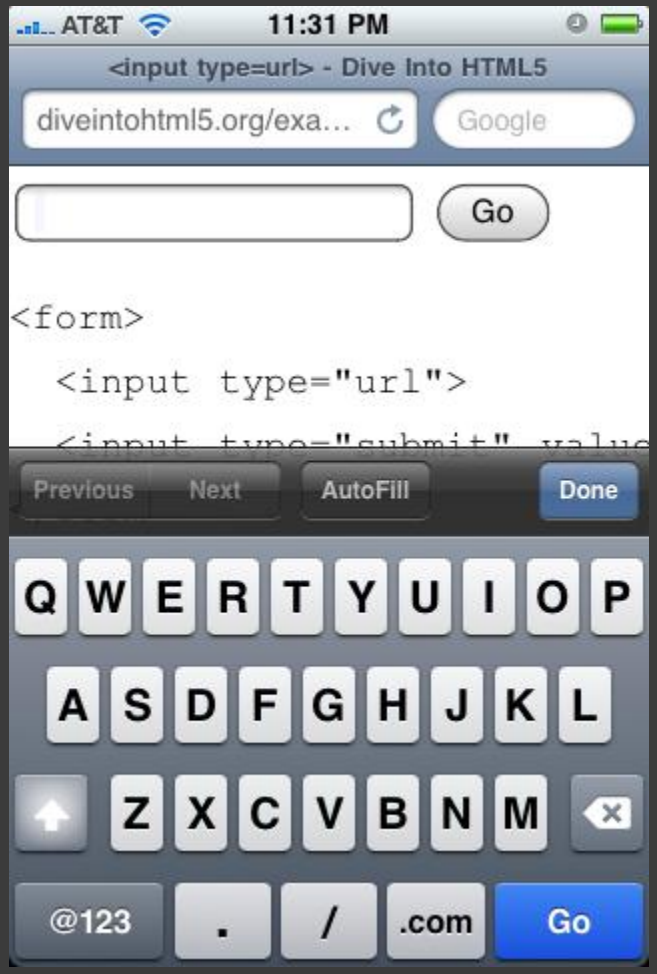


# HTML5 Input Types

*So, that was pretty cool*

- Simply changing input types can add basic validation
- Benefits go beyond validation
- Additive only – no drawbacks

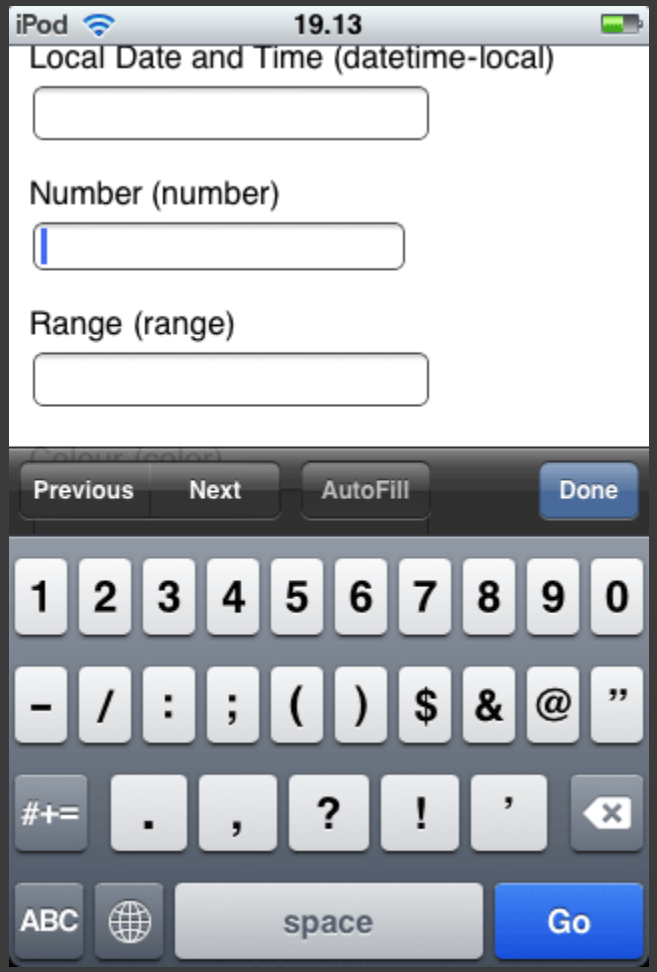
# HTML5 Input Types




**<input type="email" />**

**<input type="url" />**

# HTML5 Input Types




<input type="tel" />

<input type="number" />

# HTML5 Constraint Validation

- Required
- Pattern
- MaxLength
- Min/Max

# HTML5 Constraint Validation

Required

```
<input type="text" required />
```

MaxLength

```
<input type="text" maxlength="10" />
```

Pattern

```
<input type="text" pattern="[0-9]{5}" />
```

# DEMO: Html5 Constraints

# Of course, this only works in HTML5 capable browsers

- Older browsers will ignore these new attributes
- With JavaScript you can “Polyfill” for “regressive” enhancement

# One More Thing…

# Polyfill

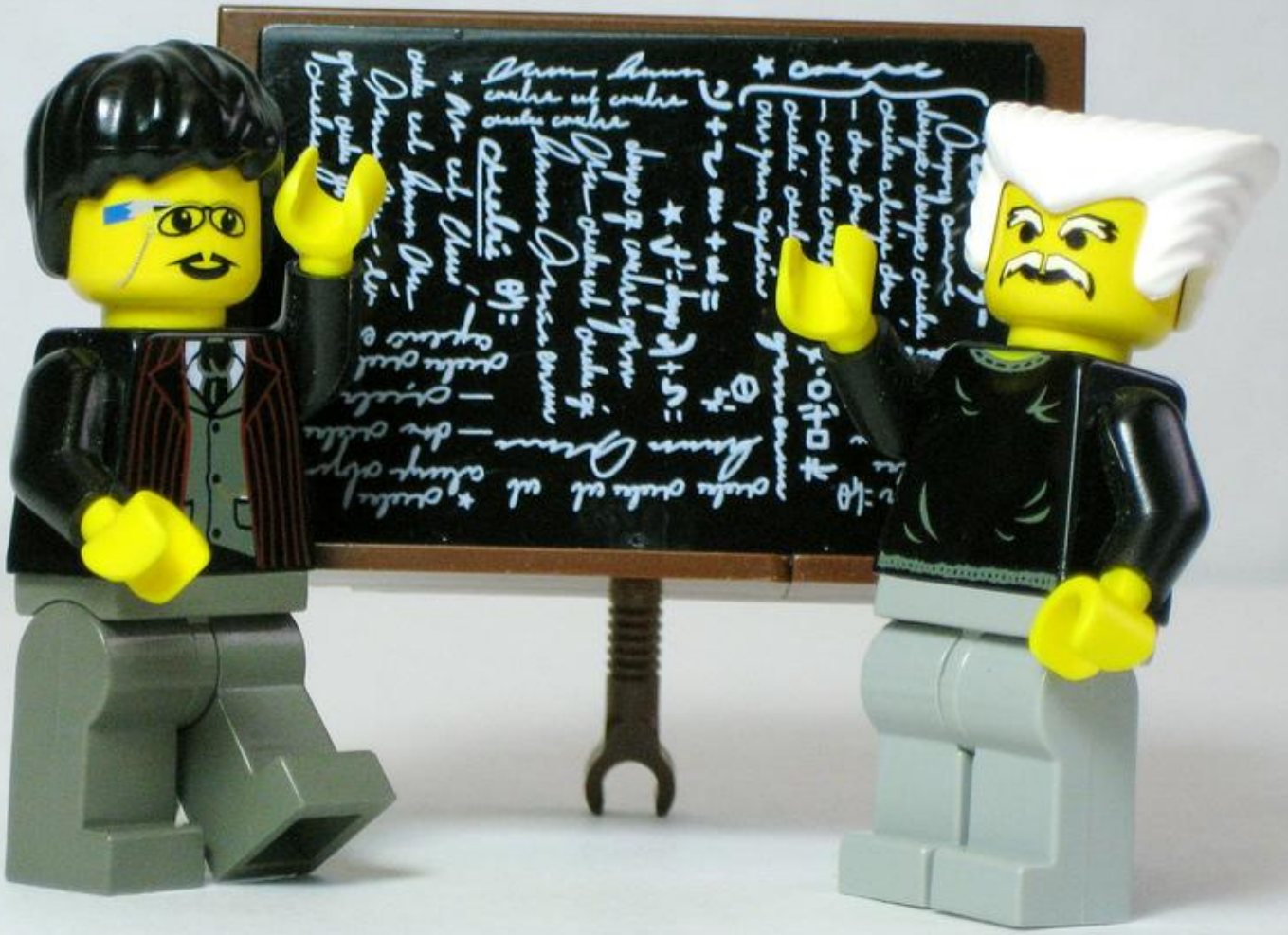
- A polyfill, or polyfiller, is a piece of code (or plugin) that provides the technology that you, the developer, expect the browser to provide *natively*

- Generally, you test the browser for a feature.  If it is not present natively, use JavaScript to add the feature

# Develop for tomorrow... today!

- Great library called Modernizr
  - http://www.modernizr.com/
- Helps with feature detection & media queries
- Allows older browsers to work with Html5 elements
- Much more

# DEMO: Polyfills



# HTML5 Data Validation: Pragmatic Advice

# Use the new input types

- They may do data validation for you
- Make your users happy (iOS & more)
- They will keep getting better
- Native experience

# Constraint Validation

- Useful for “first line of defense” or backup
- You should continue to use JavaScript for client validation

# Recap: Validating Web Forms

- Makes the experience better for your users
- Results in better, more reliable data
- First line of defense against a plethora of vulnerabilities

# Thanks for listening

- I'm Scott Kirkland
- Email: srkirkland@ucdavis.edu
- Web: http://weblogs.asp.net/srkirkland/
- GitHub:
  - Personal: https://github.com/srkirkland/
  - UCDavis: https://github.com/ucdavis/
- Slides and demo:
  - https://github.com/srkirkland/ITSecuritySymposium