Forms and Validation

- Topics:
- Sending data to the server
  - so far all we have sent to a server is clicks
  - forms
- HTML 5 validation
- Server side

<form> element

- Forms allow us to submit data to the server
- There are two ways of handling information submitted with a form: GET and POST.

```
<form action="submit_response" method="get">
```

- GET is the default and returns a query string that can be accessed by server-side code:
  - URL?name=value&name=value ...

```
http://example.com/submit_response?q=A&val=3
```

GET

- GET is best for small amounts of insecure data. If you have large chunks of data or sensitive data, use POST.

```
<form action=“submit_response" method="post">
```

- On submit, the data gets sent to the URL set in the action attribute for processing.

URL encoding

- When using GET, certain special characters will be encoded for the URL query.
  - Such as: space, comma, forward slash, equals sign, ampersand ...

```
“Xena's cool!?” -> “Xenia%27s+cool%3F
%21”
```

  - Automatically encoded by the browser. But beware when taking in the data on the server-side ...

- On submit, the data is sent in HTTP requests to the server:
  - GET asks a server for a page or data. Parameters are sent in the URL query.
  - POST submits the data to a web server and retrieves the server's response. Parameters are embedded in the request's HTTP packet.

File Upload

- You can allow uploading through your form:

```
<form action="/file-upload"
      method="post"
      enctype="multipart/form-data">
  <input type="file" name="avatar">
  <input type="submit">
</form>
```

- Then you have to process this request on the server-side...

Form Validation

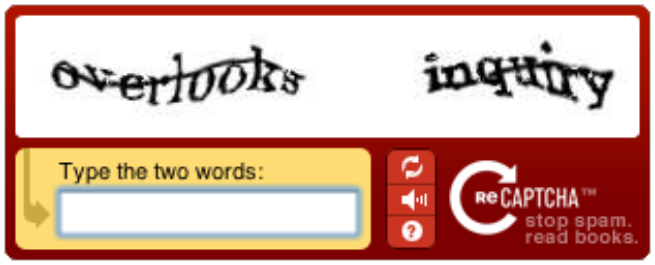
- HTML 5 validation
- Client-side
- Server-side

Which should you use? Which is best?

- Form validation means ensuring that the necessary values are submitted and are in the expected format. This includes:
    - Preventing blank (no input or space characters) values when the field is required. (HTML 5)
    - Ensuring the type of values submitted ,e.g. integer, phone number, postal code ... (HTML 5 + JS)
        - Ensuring the format and range of values.
    - Ensuring confirmation fields fit together, e.g. Comparing same email address typed twice.

- Validation involves solid knowledge and careful consideration of the data values being accepted for submission.
    - What's a valid credit card format? Phone number?
- You can use regular expressions to check the format of data values, but you need to already know the right format.

- Provide appropriate and useful feedback on the front-end to end-users.
    - Can validate on-submit or on-the-fly.
    - Can provide static hints beside fields.
    - Can provide tooltips (on-hover tips).
    - Can provide dynamic tips (shown as user interacts with a field).

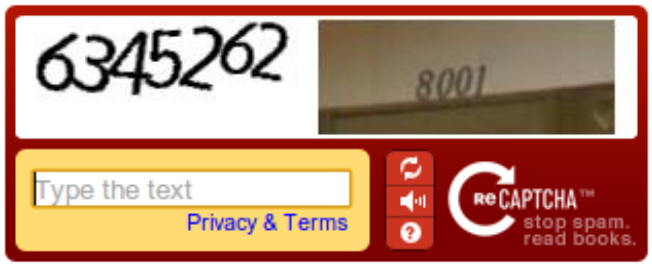


CAPTCHA

- Completely Automated Public Turing test to tell Computers and Humans Apart
- Luis von Ahn (CMU)
- Problem easy for humans to solve, but hard (impossible?) for computers to solve.
- Let to spam-sponsored CAPTCHA farms

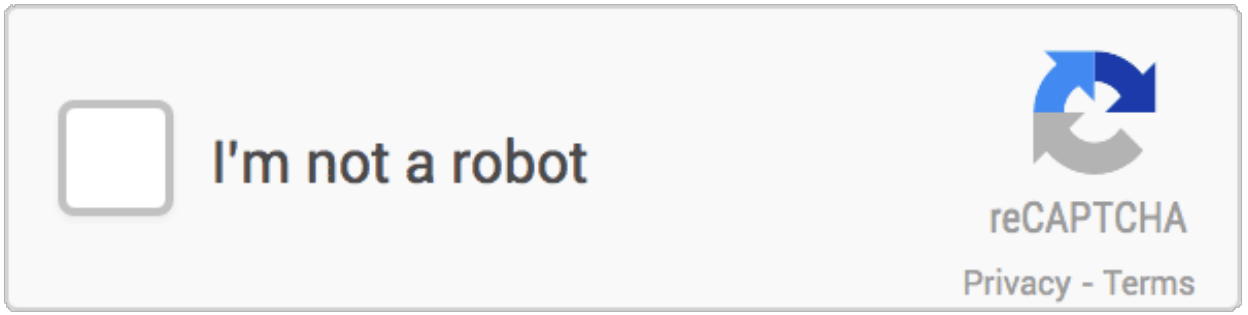
reCAPTCHA

- ~2011
- Put all that free labor to good use
- Translate real words in images and archival texts



NoCAPTCHA

- 2014 - Google found that AI could crack most CAPTCHA and reCAPTCHA images with high accuracy
- led to NoCAPTCHA: monitor interaction prior to asking to click a box “I am not a robot”



https://security.googleblog.com/2014/12/are-you-robot-introducing-no-captcha.html