

# launch \_code

LC101 2.5



# Class Agenda

1. Announcements
2. Last Class Review
3. New Material
4. Studio (Flicklist 3)



# Announcement

The Mentor Center is closed during the day on Wednesday, Thursday and Friday

GeekGala is October 19th. Tickets can be bought through EventBrite

Hackathon this Saturday at the Mentor Center, more info on slack



# Review

Query parameters - ?q=dogs the piece of the URL that passes information to the server

HTML <form> - the form tag lets you build a form in the browser

Flask form - to work with forms in flask, we need to build an HTML form in python, and we need to create a handler for the form with flask



# Validation

Validation - the program checks the user input to make sure it's correct, a specific type, or in a certain range.

Validation allows us to:

- Ensure user-submitted data “makes sense”
- Prevent users from submitting data that might break our application
- Provide user-friendly feedback when data doesn't meet our criteria



# Client side validation

Verifying the data with HTML, or JavaScript directly from the user's browser

## Pros:

- Data can be validated in “real time”
- A request to the server is never made

## Cons:

- Client side validation can be bypassed by disabling JS, or by submitting requests in different ways.



# Server Side Validation

After the HTTP request is made the first thing the server does is validate the data. This can happen in the request handler, or within the model classes.

## Pros:

- Can not be bypassed

## Cons:

- Is not user-friendly, a request has to be submitted before validation, not immediate response.



# Where should we Validate?

**Both client side, and server side.** We want to verify our data in as many places as possible.

When you get bad data, it can break your application. We have made errors in this class that have shut down our local server because of bad data. We don't want our user to do this on accident, or intentionally. So we **need to check the user input multiple times.**





# Flask - Redirect

Redirect - when you send the user to a different path.  
Used very frequently on form submits. Instead of changing the HTML of the page they are on, redirect takes them to a completely different path

Documentation:

<http://flask.pocoo.org/docs/0.12/quickstart/#redirects-and-errors>

Example: `redirect_example.py` - general use

Example: `prep_work_redirect_example.py` - on form submit



# HTML Escaping

Definition, why it's necessary, how you do it

HTML Escaping is taking the user input and sanitizing the common HTML tags out of it. So it won't render as HTML, it will render as text.

Our example used the cgi library. You can find it's documentation here:

<https://docs.python.org/3/library/cgi.html>

Example: `prep_work_client-side_example.py`



# Studio

Flicklist 3

Walkthrough

TF Groups

