

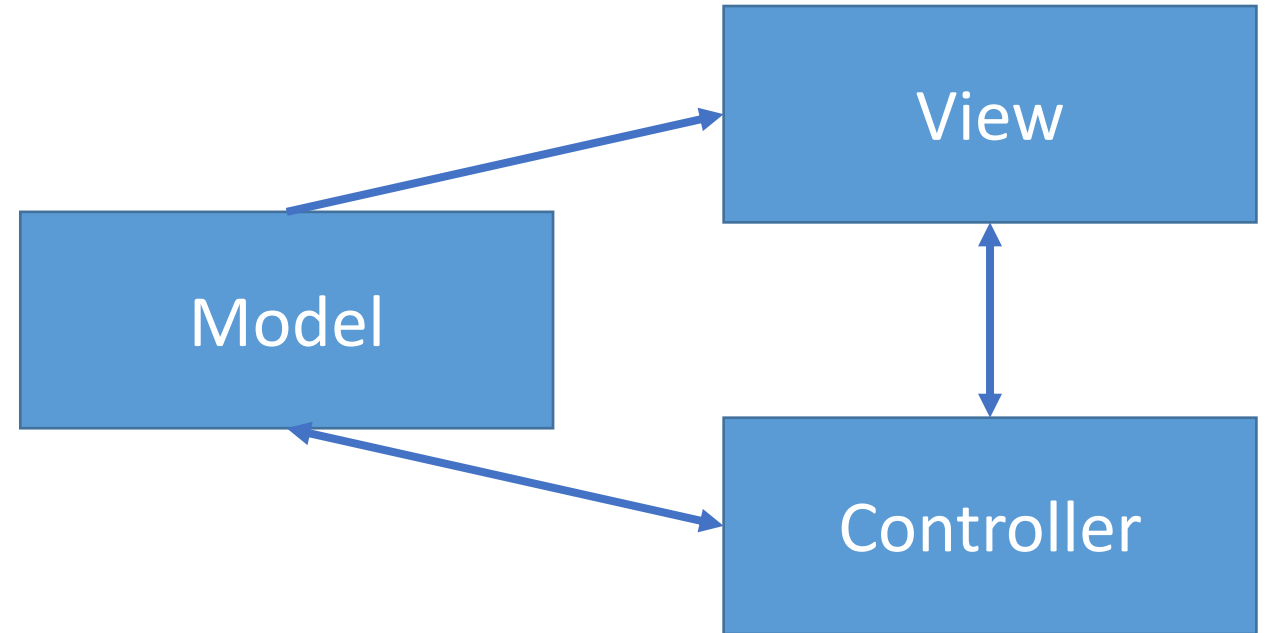
Model-View-Controller

Administrivia

- Homework is due today
- Tag on time, use the right capitalization, follow the exact instructions we provide
- If the CI isn't running as expected, validate locally/on attu
- Make sure your validation passes
- Commit and Push regularly and often. Every time you do something even slightly significant
- Check GitLab to ensure that your code was submitted correctly

MVC

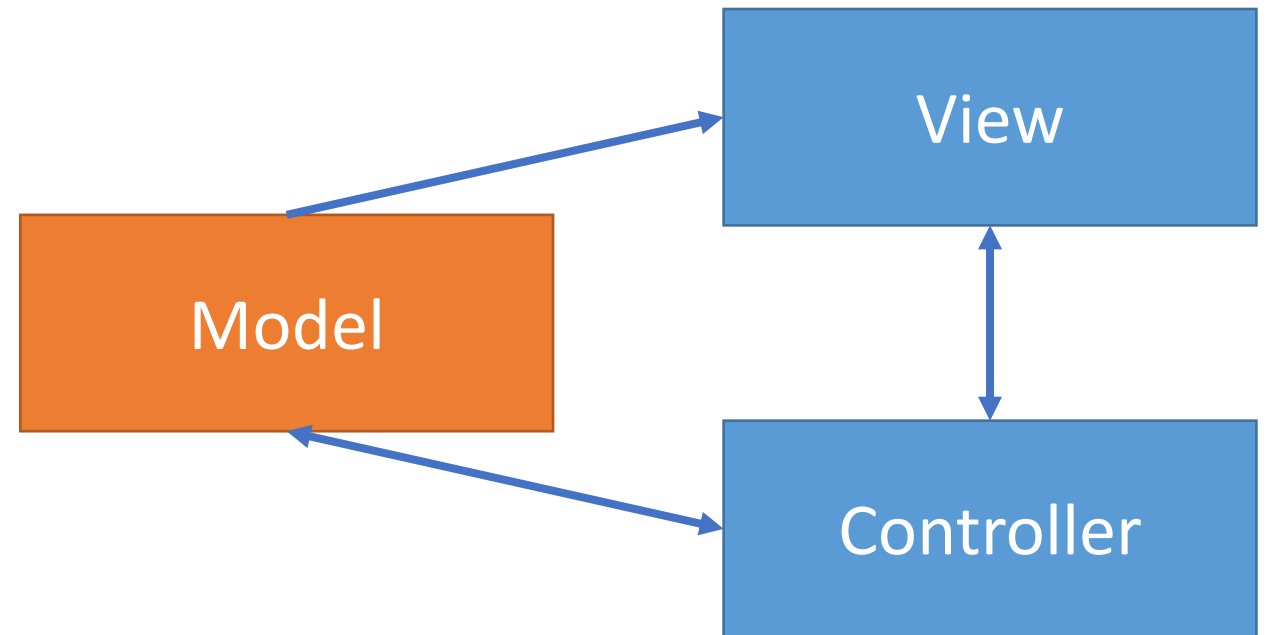
- MVC is an architectural pattern used in UI development
- The application is divided into three interconnected parts



Model

Retrieves and stores data by interacting with the data source

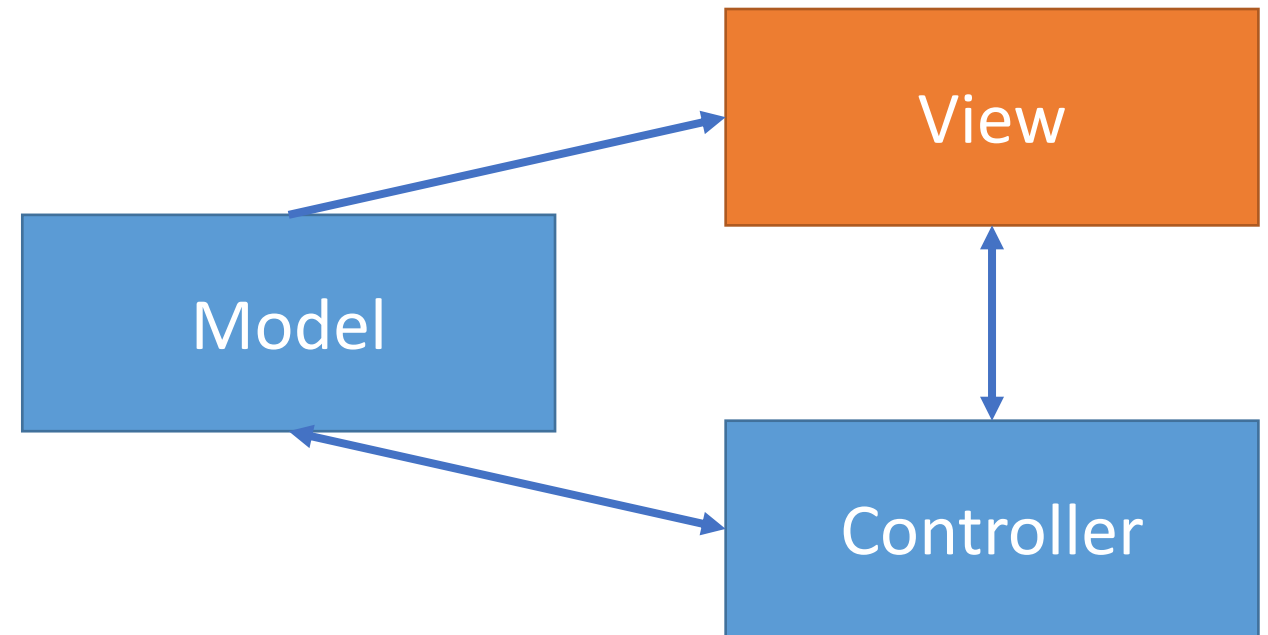
- Which table is this data stored in?
- Which SQL query should I use to get this data?



View

Gets data from the Model and presents it in a User-Friendly manner

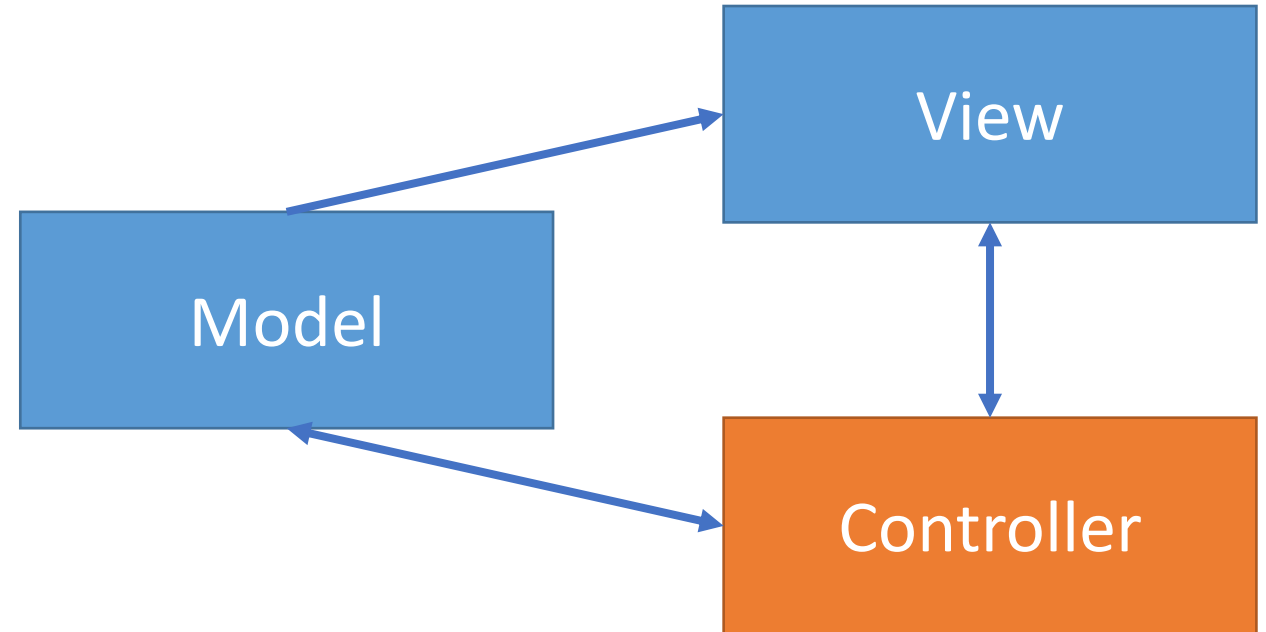
- Looks like the Model is giving me different data, let me update to show the new data
- Looks like the controller wants me to change the current display, let me change it



Controller

Listens for changes to the data or state in the UI, notifying the model or view accordingly

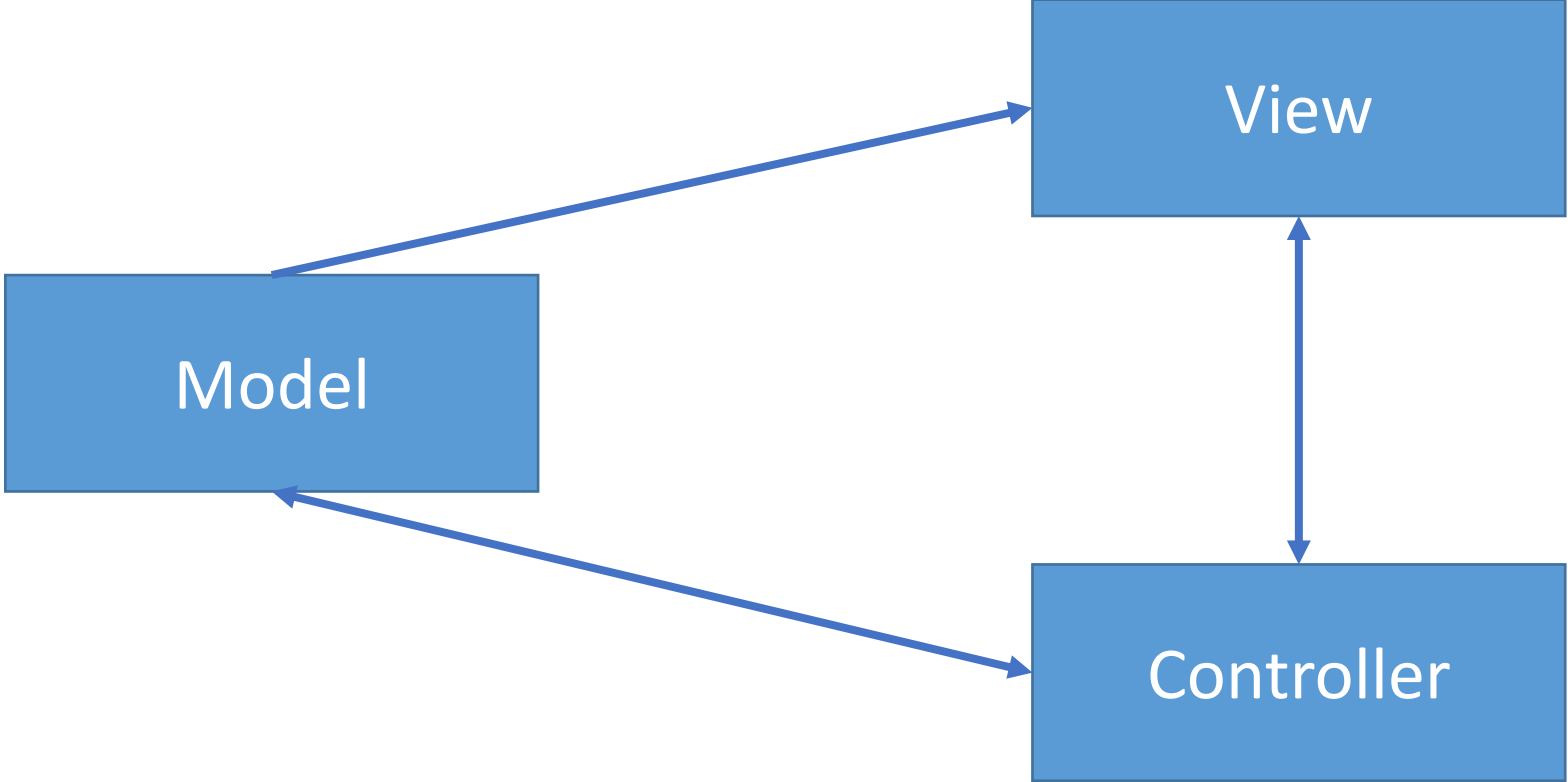
- Looks like there is now another item that matches this search. I better tell the view
- Looks like the user just clicked “Delete”. I should tell the model to delete the item



Benefits of MVC

- Organization of code
 - Maintainable, easy to find what you need
- Ease of development
 - Build and test components independently
- Flexibility and Decoupling
 - Swap out views for different presentations of the same data (ex: calendar daily, weekly, or monthly view)
 - Swap out models to change data storage without affecting user

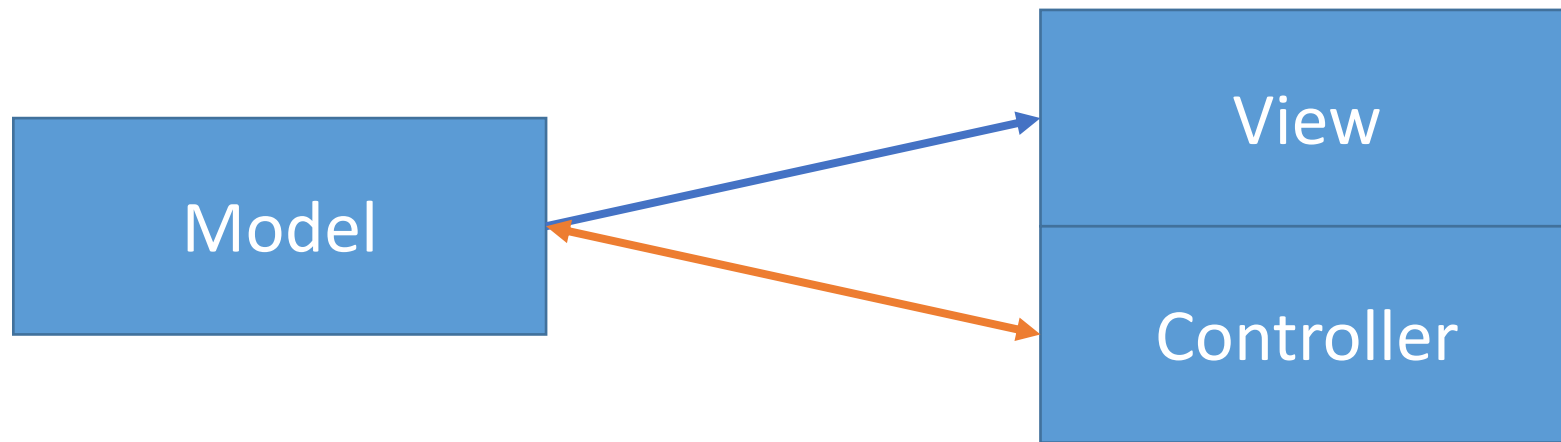
Theoretical MVC Flow



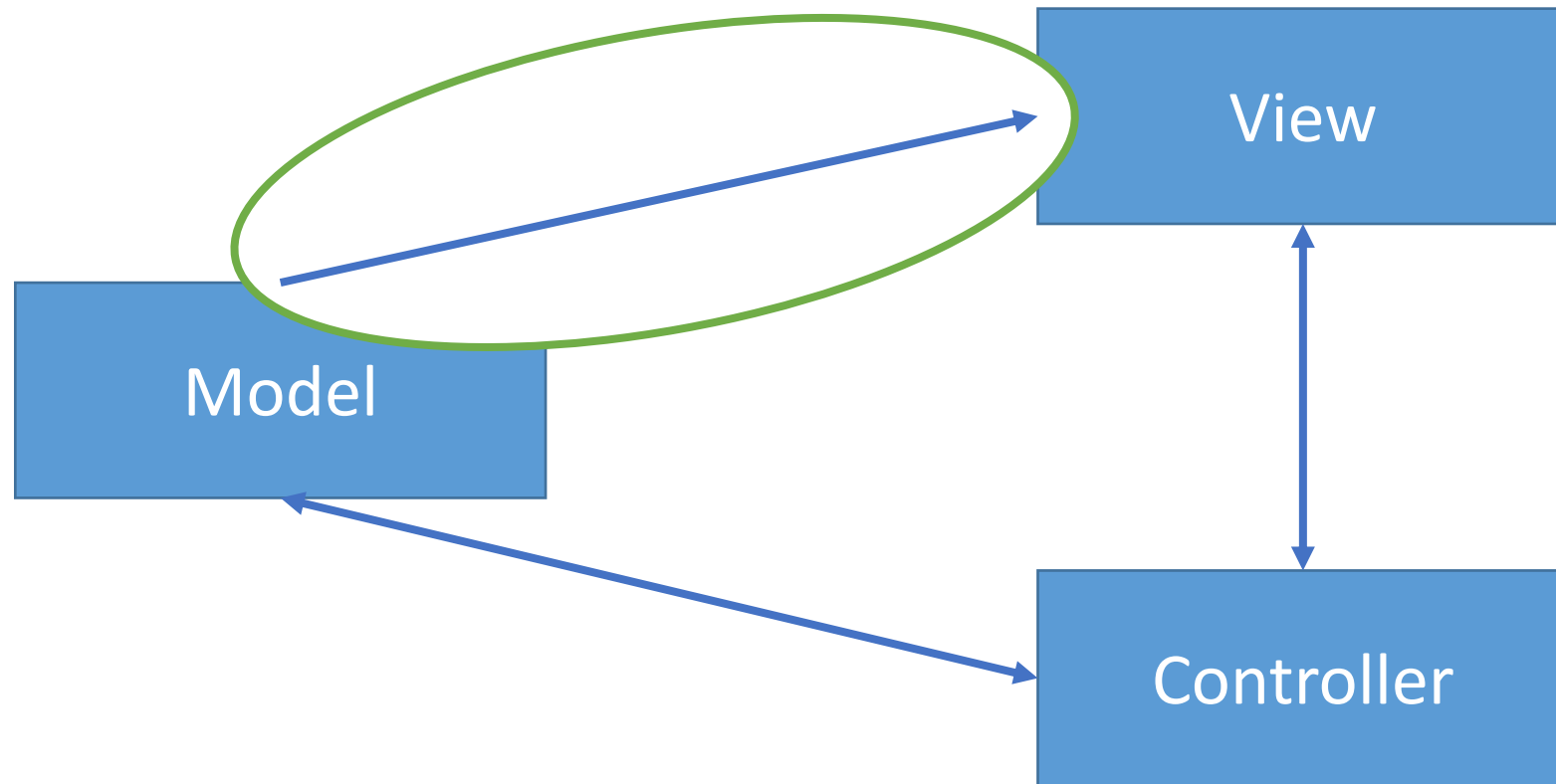
MVC Flow

- In theory
 - Controller is separate from the View
 - Controller contacts view to interpret what input events should mean in the context of the view
- In practice
 - View and controller are so intertwined that they almost always occur in matched pairs (ex: command line interface)
 - Many architectures combine the two

Used MVC Flow



Push vs Pull



Push vs Pull

Push Architecture

- As soon as the model changes, it notifies all of the views
- Eg. Gmail automatically shows new emails as soon as one comes up without refreshing the page
- Guaranteed to have latest data in case something goes wrong later on

Pull Architecture

- When a view needs to be updated, it asks the model for new data
- Eg. **What do you think is an example of Pull Architecture?**
- Avoid unnecessary updates, not nearly as intensive on the view

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			
Display price/details of a product			
Store product/inventory details			
Purchase items in shopping cart			
Record customer transactions			
User sign-in			
Authenticate user sign-in attempt			
Check user credentials			

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			X
Display price/details of a product			
Store product/inventory details			
Purchase items in shopping cart			
Record customer transactions			
User sign-in			
Authenticate user sign-in attempt			
Check user credentials			

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			X
Display price/details of a product		X	
Store product/inventory details			
Purchase items in shopping cart			
Record customer transactions			
User sign-in			
Authenticate user sign-in attempt			
Check user credentials			

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			X
Display price/details of a product		X	
Store product/inventory details	X		
Purchase items in shopping cart			
Record customer transactions			
User sign-in			
Authenticate user sign-in attempt			
Check user credentials			

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			X
Display price/details of a product		X	
Store product/inventory details	X		
Purchase items in shopping cart			X
Record customer transactions			
User sign-in			
Authenticate user sign-in attempt			
Check user credentials			

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			X
Display price/details of a product		X	
Store product/inventory details	X		
Purchase items in shopping cart			X
Record customer transactions	X		
User sign-in			
Authenticate user sign-in attempt			
Check user credentials			

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			X
Display price/details of a product		X	
Store product/inventory details	X		
Purchase items in shopping cart			X
Record customer transactions	X		
User sign-in		X	
Authenticate user sign-in attempt			
Check user credentials			

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			X
Display price/details of a product		X	
Store product/inventory details	X		
Purchase items in shopping cart			X
Record customer transactions	X		
User sign-in		X	
Authenticate user sign-in attempt			X
Check user credentials			

MVC Example - Amazon

	Model	View	Controller
Update user's Shopping Cart			X
Display price/details of a product		X	
Store product/inventory details	X		
Purchase items in shopping cart			X
Record customer transactions	X		
User sign-in		X	
Authenticate user sign-in attempt			X
Check user credentials	X		

HW8 Data Parsing

- 2 files to be parsed:
 - campus_buildings.tsv
 - campus_paths.tsv
- Both are tsv files that can be parsed with opencsv (or similar)
- Similar to the demo. Consider looking at the parse() method of CsvToBean

HW7 and HW8

- HW7 Questions
- HW8