

hello, section!

week 8

pset 7 recap

MVC

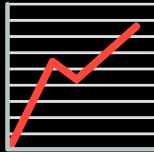
MVC

organization of a web application:

1. Model
2. View
3. Controller

MVC

organization of a web application:

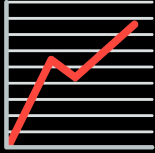

1. Model → Data 

2. View → Interface 

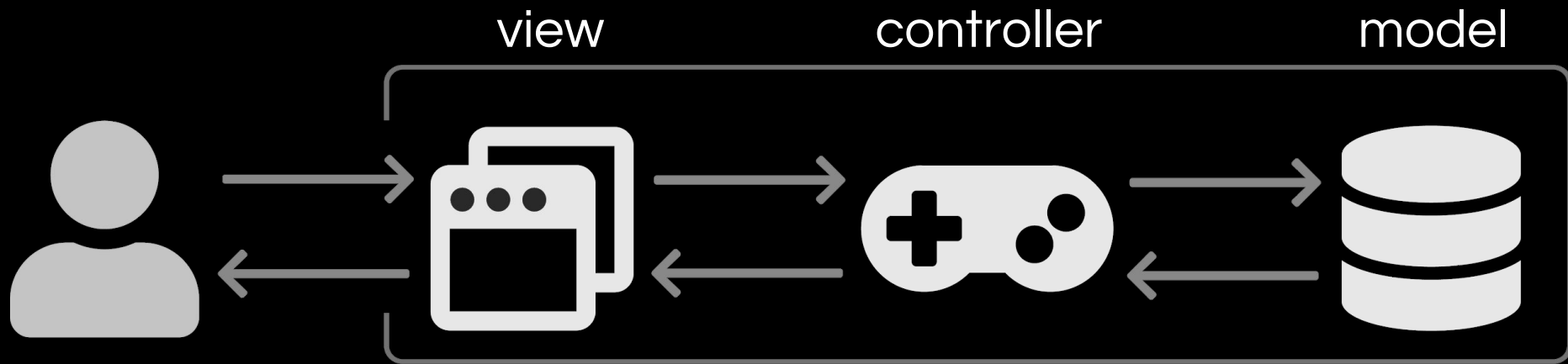
3. Controller → Logic 

MVC

organization of a web application:

1. Model → SQL 
2. View → HTML/CSS 
3. Controller → Python 

MVC



Flask/Jinja review

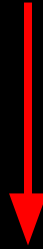
SQL

SQL

structured query language

SQL

structured query language



store and manage information in
the form of a database

SQL

why use a database?

- persistent memory
- efficiency
- accuracy
- security

SQL

- **INTEGER**: smallint, integer, bigint
- **NUMERIC**: boolean, date, datetime, numeric(scale, precision), time, timestamp
- **REAL**: real, double precision
- **TEXT**: char(n), varchar(n), text

SQL

database review

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	3	Adams
11250023	Jay Doe	4	Currier

classes

id	name	class
1048	James Doe	CS50
1049	James Doe	HUM10
1050	Jay Doe	STAT110

database

SQL

students

tables

classes

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	3	Adams
11250023	Jay Doe	4	Currier

id	name	class
1048	James Doe	CS50
1049	James Doe	HUM10
1050	Jay Doe	STAT110

database

SQL

columns

students



The diagram shows the word 'columns' in orange at the top. Three orange arrows point from it to the column headers of two tables below. The first arrow points to the 'id' column of the 'students' table. The second arrow points to the 'name' column of the 'students' table. The third arrow points to the 'class' column of the 'classes' table.

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	3	Adams
11250023	Jay Doe	4	Currier

classes

id	name	class
1048	James Doe	CS50
1049	James Doe	HUM10
1050	Jay Doe	STAT110

database

SQL

primary keys

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	3	Adams
11250023	Jay Doe	4	Currier

classes

id	name	class
1048	James Doe	CS50
1049	James Doe	HUM10
1050	Jay Doe	STAT110

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	3	Adams
11250023	Jay Doe	4	Currier

classes

id	name	class
1048	James Doe	CS50
1049	James Doe	HUM10
1050	Jay Doe	STAT110

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	3	Adams
11250023	Jay Doe	4	Currier

what is
wrong here?

classes

id	name	class
1048	James Doe	CS50
1049	James Doe	HUM10
1050	Jay Doe	STAT110

database

SQL


students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	3	Adams
11250023	Jay Doe	4	Currier

classes

id	name	class
1048	James Doe	CS50
1049	James Doe	HUM10
1050	Jay Doe	STAT110

name is
not a good
identifier!



database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	3	Adams
11250023	Jay Doe	4	Currier

instead...

classes

id	name	class
1048	James Doe	CS50
1049	James Doe	HUM10
1050	Jay Doe	STAT110

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier

foreign key



classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110

database

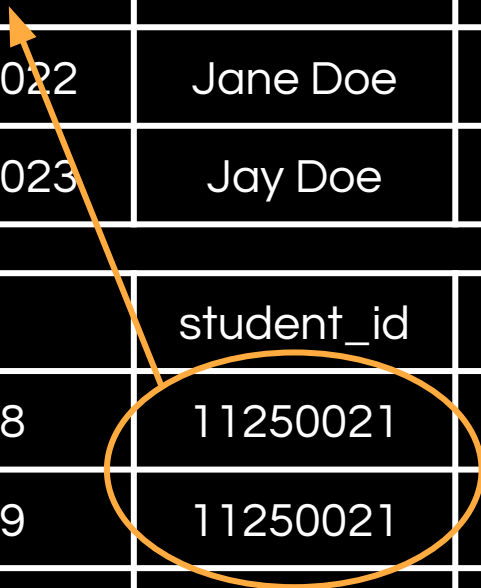
SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110



database

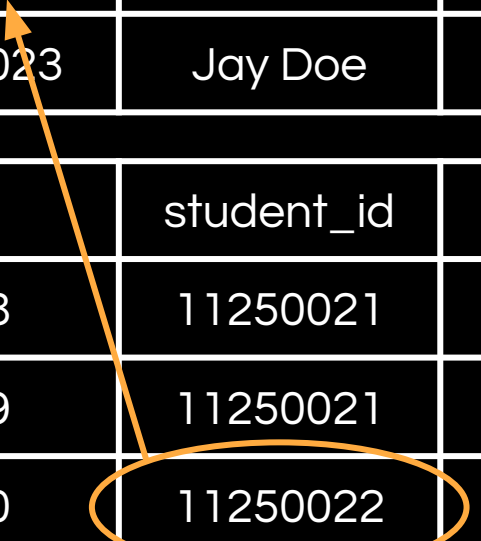
SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110



SQL

SQL syntax

SQL

INSERT
SELECT
UPDATE
DELETE

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110

SQL

the INSERT query adds information to table

INSERT INTO

<table>

(<columns>)

VALUES

(<values>)

SQL

the INSERT query adds information to table

```
INSERT INTO
```

```
classes
```

```
(student_id, class)
```

```
VALUES
```

```
('11250022', 'CS61')
```

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110
1051	11250022	CS61

SQL

the INSERT query adds information to table

```
INSERT INTO
```

```
students
```

```
(name, year)
```

```
VALUES
```

```
('John Doe', 1)
```

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	1	None

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110
1051	11250022	CS61

SQL

the SELECT query extracts information from table

```
SELECT  
<columns>  
FROM  
<table>  
WHERE  
<predicate>
```

SQL

the SELECT query extracts information from table

```
SELECT  
name, year  
FROM  
students
```

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	1	None

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110
1051	11250022	CS61

SQL

the SELECT query extracts information from table

```
SELECT  
name, year  
FROM  
students  
WHERE  
house = 'Thayer'
```

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	1	None

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110
1051	11250022	CS61

SQL

the SELECT query extracts information from table

```
SELECT
```

```
*
```

```
FROM
```

```
students
```

```
WHERE
```

```
year < 2
```


database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	1	None

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110
1051	11250022	CS61

SQL

the UPDATE query modifies information in a table

UPDATE

<table>

SET

<column> = <value>

WHERE

<predicate>

SQL

the UPDATE query modifies information in a table

UPDATE

students

SET

year = 2, house = 'Winthrop'

WHERE

name = 'John Doe'

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	2	Winthrop

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110
1051	11250022	CS61

SQL

the DELETE query removes information from table

DELETE FROM

<table>

WHERE

<predicate>

SQL

the DELETE query removes information from table

```
DELETE FROM
```

```
classes
```

```
WHERE
```

```
student_id = 11250022
```

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	2	Winthrop

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10

SQL

SQL with Flask

SQL

we will use a **SQLite** database and access it as:

```
db = SQL("sqlite:///database.db")  
db.execute(query)
```

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	2	Winthrop

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110
1051	11250022	CS61

SQL

```
db = SQL("sqlite:///database.db")
```

```
...
```

```
item = db.execute("SELECT * FROM students  
WHERE name = 'John Doe'")
```

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	2	Winthrop

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110
1051	11250022	CS61

SQL

```
db = SQL("sqlite:///database.db")
```

```
...
```

```
deleted_course = "CS61"
```

```
item = db.execute("DELETE FROM classes WHERE  
class = :course", course=deleted_course)
```

database

SQL

students

id	name	year	house
11250021	James Doe	1	Thayer
11250022	Jane Doe	2	Adams
11250023	Jay Doe	4	Currier
11250024	John Doe	2	Winthrop

classes

id	student_id	class
1048	11250021	CS50
1049	11250021	HUM10
1050	11250022	STAT110

Quiz Review

Ajax

Ajax

asynchronous Javascript and XML

Ajax

asynchronous Javascript and
XML



constantly updating webpage
w/o needing to refresh

Ajax

examples

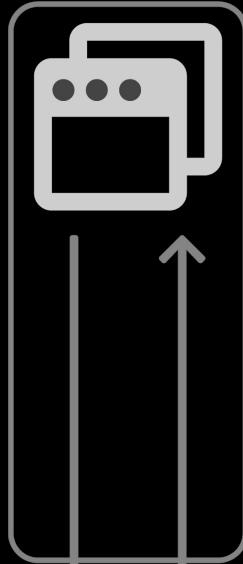
sports scores

emails

what are other examples?

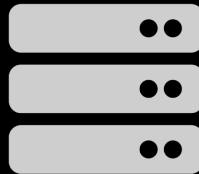
w/o Ajax

web browser



HTTP
request

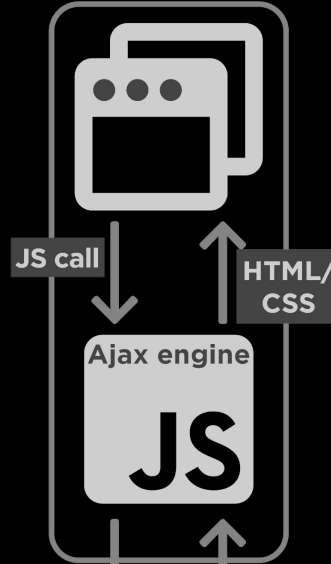
HTML/
CSS
data



server

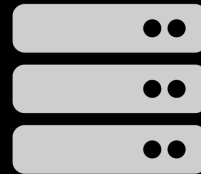
w/ Ajax

web browser



HTTP
request

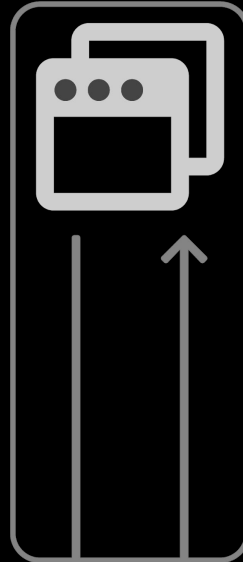
JSON/
HTML
data



server

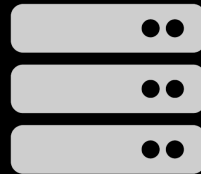
w/o Ajax

web browser



HTTP
request

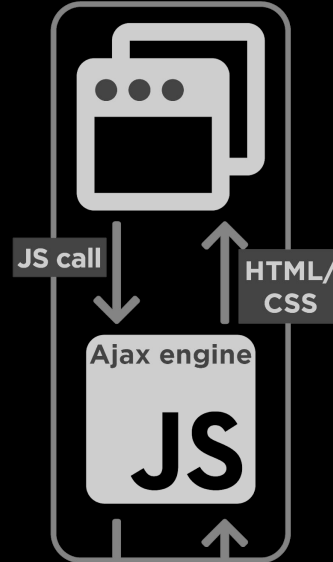
HTML/
CSS
data



server

w/ Ajax

web browser

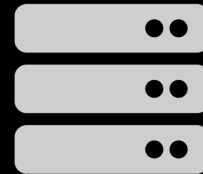


JS call

HTML/
CSS

HTTP
request

JSON/
HTML
data



server

Javascript
"client side"

Python
"server side"

pset 8 tips & tricks

- make sure you understand the workflow of **MVC**
- name your columns in SQL appropriately!

final project to-dos

- Preproposals due by 11:59pm on Tue 11/6
- If collaborating with 1 or 2 classmates, each of you should submit a preproposal, even if identical.
- Proposals due by 11:59pm on Tue 11/13

After CS50

1. CS20
2. CS51
3. CS124