Exams

- Midterm Friday 6/2
- I will email you random seat numbers the night before, to your ucdavis.edu address
- TAs will have seating map

- If you are satisfied with your scores on the two midterms, you can skip the final
- As soon as your Photobooth and midterm are graded, I can give you your course grade (so far) so you can decide

Practice test

- Covers mostly AJAX, server, DB, API topics
- On Web site
- Open book, open notes
  - What to bring? Mostly code
  - All example programs I distributed
  - Your own code

Put together like legos

- I am mostly concerned with how you fit the specific operations we did (SQL operations, AJAX requests, changing the DOM, API reference, server…) into a larger whole.
- So I’d like you to be able to copy (and modify as necessary) individual operations.

1. There are a number of things the browser can do that the server cannot, and also a number of things the server can do that the browser cannot. Label each JavaScript line below as most likely browser code (B), server code (S), or growth code (G) that could be in either the browser or server. Add a sentence of explanation if you think the answer is ambiguous.

   ```javascript
   let request = new XMLHttpRequest();
   request.open('GET', 'http://data.instagram.com/api/v1/photos?no=9000000', false);
   request.send();
   if (request.status == 200) {
     let allAlbums = request.responseText;
   }
   ``

   ```javascript
   for (let id of ids) {
     let request = new XMLHttpRequest();
     request.open('GET', 'http://data.instagram.com/api/v1/photos?no=' + id, false);
     request.send();
     if (request.status == 200) {
       let photo = request.responseText;
     }
   }
   ```

   ```javascript
   let request = new XMLHttpRequest();
   request.open('GET', 'http://data.instagram.com/api/v1/photos?no=9000000', false);
   request.send();
   if (request.status == 200) {
     let photo = request.responseText;
   }
   ```

2. Answer True or False. Add a sentence of explanation only if you think the question is ambiguous.

   a) Anonymous functions have to be defined inside other functions.
   b) The Same Origin Policy is implemented by the browser.
   c) An AJAX request should always define a callback.
   d) A server can send an HTTP GET request to a browser.
3. Multiple choice: Which of the following statements about our Photobooth project is true? Add a sentence of explanation if you think the question is ambiguous.
(a) We sent an HTTP request from the browser to the Google Cloud Vision API using an XMLHttpRequest object.
(b) We sent an HTTP request from the server to the Google Cloud Vision API using code from a Node module.
(c) We sent an HTTP response from the server to the Google Cloud Vision API using the response object.

Hard Javascript problem

5. Write out the contents of x, as a Javascript literal. What is its data type?

```javascript
function showItemPrice(str) {
  console.log(str);
}

function createNewAction(id, price) {
  return function() {
    showItemPrice('item .+' + id + ' .price = ' + price);
  }
}
x = createNewAction(0, "$37.99");
```

System problem

6. In an effort to provide some privacy for photos on our server, we store a photo not under its real name but using a 14-digit ID number, which is stored in the database. So when the browser wants to display a photo, instead of bala.jpg it needs to know the ID number for because it would access balal.jpg at 138.48.25.50:/9779/module/home/bala.jpg. The browser can find out the ID number of a photo using an AJAX request to the server. We have decided that the format of this query should be something like this:

138.48.25.50:/9779/query/getphoto(123456789012345678901234567890).

Because that our colleagues have already written the server code to correctly handle this operation, it returns the ID number in the body of the response, or the string "not found" if the photo is not in the database.

```javascript
function getPhoto(id) {
  var url = "http://138.48.25.50:60421/";
  var filename = document.getElementById("photoQuery").value;
  var xhr = new XMLHttpRequest;

  xhr.onreadystatechange = function() {
    if (xhr.readyState == 4) {
      if (xhr.status == 200) {
        var content = xhr.responseText;
        var id = parseInt(content); // Extract the ID number from the response
      } else {
        alert("Error: " + xhr.statusText);
      }
    }
  }

  xhr.open("GET", url + "query/getphoto(123456789012345678901234567890).jpg");
  xhr.send(null);
}
```
function displayPhoto(idNum) {
  var body = document.getElementsByTagName("body");
  var newDiv = document.createElement("div");
  newDiv.id = "photoDiv";

  var newImg = document.createElement("img")
}