**Announcements**

- No class on Wednesday (University Day)
  - Friday recitations in Chapman
- HW 2 solutions posted online
- Mid-term next Wednesday (in-class)
  - Bring only your pens (I provide the paper)
  - No notes, books, computers, calculators, aid, allowed
- HW 3 deadline next Monday
  - If want it to be moved to Wednesday, speak up NOW!
  - Make extensive use of office hours this week!
- Mid-semester anonymous survey posted online
  - Please take a few minutes to fill it out now
- 2-min presentations by:
  - Hayley Nicole Hoyle
  - Thomas Edward Seelinger

**HW 3 approach**

- Keep collecting input till user wants --- how?
  - While loop that checks what value is returned by `confirm`
- Draw all valid inputs on chess-board
  - Only one chance to draw board --- so?
  - Collect all inputs before attempting to draw chess-board
- If any invalid input field, do not draw chess-board --- so?
  - Same as above
- Detect overlapping inputs --- how?
- Remember all non-overlapping moves (and whether valid move, what piece, etc) --- how?
  - Use 2-D arrays for representing properties of the 64 squares
    - `colorArray`, `chessContentArray`
  - For each move, remember:
    - The colors of starting and ending squares (in `colorArray`)
    - The image to be displayed (piece image, blank image, or ?!)
    - Overlaps: look up if starting or ending cell is already colored
Example

function makeTable()
{
  document.write("<table border="3" >");
  document.write("<tr><td> $$$ </td>");
  document.write("<td> $$$ </td>");
  document.write("<td> $$$ </td></tr>");
  document.write("</table>");
}

- JavascriptFunctionExamples.html
  - Page 1: Write once, call twice!

Communication to functions

function incrementNumber()
{
  x++;  
}

...  
var x = 5;
incrementNumber();

- Using global variables:
  - Subroutines have access to all variables declared in main
    - JavascriptFunctionExamples.html: Page 2
      - info is known throughout the script
  - Requires function to know variable name
  - ☹️ harder to (re-)use with variables other than x
Communication to functions

  - Story with place holders
  - Replace place holders with actual data & read!

- Using function parameters
  - Actual parameter value is substituted for formal; function executed

```javascript
function incrementNumber(n) {
    n++;
    document.write("In the function, \+ n +\"<br>\"); 
}
```

```javascript
var x = 5;
incrementNumber(x);
var y = 10;
incrementNumber(y);
document.write("After the function, \+ x +\"<br>\"); 
```

---

Function parameters

```javascript
function makeTable(d) {
    document.write("<table border="3">\n    document.write("<tr><td>" + d + "</td>\n    document.write("<td>" + d + "</td>\n    document.write("<td>" + d + "</td></tr>\n    }
```

- JavascriptFunctionExamples.html
  - Page 3: use parameter to determine table content
    - At function call,
      1. actual parameter value is substitute for formal, and
      2. function code is executed
  - Better than using global variable
    - Don't need to know variable name
    - Works for any variable
Multiple parameters

function makeTable(x,y,z)
{ document.write(“<table border="3" >”);
  document.write(“<tr><td>”+x+”</td>”);
  document.write(“<td>”+y+”</td>”);
  document.write(“<td>”+z+”</td></tr>”);
}

makeTable(“Raleigh”, “Durham”, “Chapel Hill”);

- Parameters are matched by position
  - Parameters values are substituted by order
  - JavascriptFunctionExamples.html
    - Page 4: use multiple parameters

Function return value

function double(x)
{ return x+x; }

... 
document.write("Double of “+ x +" is “+ double(x));
salary = normal + double(otime);
document.write(double(“1”)); // what’ll this do?

- Functions can also return a value
  - At function call:
    1. Values is substituted for formal
    2. Function is executed
    3. Value is returned
  - Since value is returned, function call is an expression!