

Announcements

- **Quiz #5 on Wednesday 12/9**
- **Project #5 due on Thursday 12/10**
- **Final Exam: Monday 12/14**
4:00PM to 6:00PM
TYD 0130 (Tydings Hall)

2-Dimensional Arrays

Another example: [MultiDimensionalArray2.html](#)

Other Shortcuts

$x = x + 7;$	$x += 7;$
$x = x - 7;$	$x -= 7;$
$x = x * 7;$	$x *= 7;$
$x = x / 7;$	$x /= 7;$

Loop Control Statements

- **break** – immediately terminates the loop
- **continue** – immediately goes to the top of the loop

Example: BreakAndContinue1.html

BreakAndContinue2.html

Recursion

There are two ways to repeat tasks:

- 1. Loops**
- 2. Recursion**

The defining characteristic of “recursion” is a function that calls itself!

In theory, any problem that can be solved with loops could also be solved with recursion (and vice versa), but some lend themselves more to one technique or the other.

Recursion – Main Idea

If (problem is really simple)

solve it directly and return the answer

else

- 1. Break the problem into smaller instance(s) of the original**
- 2. Solve smaller instances by calling this function again**
- 3. Combine solution(s) to solve original problem**
- 4. Return the answer**

Recursion Examples

1. `function factorial(n)`

Returns `n factorial`.