Announcements…

- If I still have your exam, please come to the front to get it now.

- Project #5 has been posted

Two ways to "repeat"

- **Loops ("Iteration")**
- **Recursion**

Are there situations where only one of these works?

Technically, no. Any problem involving repetition could be solved either way.

Are there situations where one is more natural (easier to code) than the other?

Some people find loops easier for simple tasks.

There are **many** situations where recursion is more natural. We haven't seen them (yet).

Intro to Recursion

Suppose I have to wash a pile of dishes...

- How does this process look with a loop (iteration)?

    Allow me to illustrate...

Intro to Recursion

How would this process look if done "recursively"?

- If there is only ONE plate, I will just wash it. (Remember this.)

- If there is more than one plate... Things get interesting. (Next slide please...)

Use Your Imagination…

What if…. I have a machine that I can use to make a clone (copy) of myself. Whenever I have a job to do, I can create a "helper"!

Rules:

1. Clone must be assigned a *portion* of the same job that I am working on. (Not the whole thing, and not some other kind of job).

2. We cannot do the work at the same time.

3. When clone is done with his *portion* of the job, he tells me he is done, then vanishes, leaving me to complete the job. (Poof!)

Allow me (us) to demonstrate…

Let's write some code…

Let's write this twice: Once with iteration (looping) and once with recursion:

var function washDishes(numberOfDishes) {

   …

}

Factorial Example

Recall:

> **5! = 5 * 4 * 3 * 2**
>
> > **= 120**

Consider:

```
function factorial(n) {
    …
}
```

Let's solve it with a loop first.

How would this work with recursion?