Remarks

About the slides:
- They do not replace lecture notes (read both)
- Material written on the board also counts (e.g., we did the square problem on the white board)
- Read the reading assignments

How to prepare for the lab:
- Read the lecture notes / review slides
- Run the posted codes (this week: hypnotized.py and circle.py)
- Solve the homework (posted late Tuesdays)
Functions with parameters

Warm-up problem:

Define a function that draws a square with side length $S$ ($S$, the **parameter**, will be specified when calling the function).

Then, define a function that draws three squares with side-lengths 10, 20, 30.
Problem 1: Hypnotized

Draw a spiraling shape, starting at the center, increment the length of each side by 1, finish at length 20:

Pseudo code:

(Attempt 1: an infinite spiral.)

Brute force

```python
def spiral(S):
    fd(S)
    lt(90)
    spiral(S+1)

call spiral(1)
```

Note: recursion = a function calls itself (e.g. spiral)
Problem 1: Hypnotized

Stack trace diagram of the infinite spiral:

Remark: **Recursion** = when a function is calling itself
Fixing the pseudo code - introducing **conditionals**.

```python
def spiral(S):
    # draw a line of length S
    # turn left by 90 degrees
    if S < 21:
        call spiral(S+1)
    call spiral(1)
```

Types of conditions:

- `1 < 2`
- `1 == 2` (double equal)
- `1 != 2`
Problem 1: Hypnotized

Stack trace of the new pseudo code:

did on the board

Testing:

test for different values of $s$, e.g.

$s = 0, 1, 5, 10, 19, 20, 21, 25$

close to the "boundary" conditions

(we want to stop with $s = 20$ so we test values close to 20)
Problem 2: Circle

Draw a circle of radius 5.

Idea: draw a polygon with MANY sides

Pseudo code:

```python
def polygon(s, n, t):
    if t <= n:
        fd(s)
        lt(360/n)
        polygon(s, n, t+1)

call polygon(10, 9, 0)  # circle will need more than 3 sides, see the posted Python code
```

Introducing: functions with several parameters.
Problem 2: Circle

Stack trace diagram:

try on your own

Testing: