- Announcements
- Questions
- objects and methods
- Functions
  - possible errors
    - wrong number of arguments
    - unindenting something that should be part of the function
  - can we use `temp_F` or `temp_C` outside definition?
    - no, function bodies are their own little world
    - this is called **scope** and we'll talk more about next week
  - ```
    1    def C_to_F(temp_C):
    2        temp_F = temp_C * 9 / 5 + 32
    3        return temp_F
    4
    5    cur_temp_F = C_to_F(cur_temp)
    6    <rest of program>
    ```
    - In what order does the computer execute the above lines?
    - 1 -> 5 -> 2 -> 3 -> 6
    - remember the instruction pointer? A function call and `return` are instructions that modify it
  - Mysteries
    - ```
      def f(x):
          return x + 2
      def g(y):
      ```

```
    return y * 3
a = 1
print(f(g(a)), g(f(a)))
```

- ```
  def f(x):
      return x + 2
  def g(y):
      return f(y) * f(y + 1)
  a = 1
  print(g(a))
  ```

Conditionals

boolean expressions

relational operators

- less than (<), less than or equal to (<=)
- greater than (>), greater than or equal to (>=)
- equal (==)
- not equal (!=)

can use and, or to combine expressions

- check if variable x is positive number at most 10: `x > 0 and x <= 10`
- can use parentheses to control evaluation
  - `current_temp > hot_temp and chance_of_rain > 0.4 or chance_of_clouds > 0.5`
  - `(current_temp > hot_temp and chance_of_rain > 0.4) or chance_of_clouds > 0.5`
    - same as first version, by default ands and ors are evaluated left to right
  - `current_temp > hot_temp and (chance_of_rain > 0.4 or chance_of_clouds > 0.5)`

Practice: write an absolute value function

- ```
  def abs(x):
      if x < 0:
          return -x
  ```

```
    else:
        return x
```

- Mysteries
  - ```
    def is_even(x):
        return x % 2 == 0
    turns = 7
    if turns == 0:
        print("first turn")
    elif is_even(turns):
        print("even turn")
    else:
        print("odd turn")
    ```
    - boolean expression for odd number?
      - `not is_even(turns)`
      - `x % 2 == 1`
- Practice: number guessing game
  - **given a player's guess and a secret number, print "cold" "warm" or "lava"**
    - only print one hint

Extras

- randomness
- numeric data types
    ```