

▼ Announcements

- pair programming survey
- lab 0 due 9pm tonight
- ▼ lab 1 out
 - a similar amount of code to how much you wrote for lab 0
 - but now I'm asking you to translate more complex ideas into code than simple formulae
 - quick demo of how to get set up to do the lab
- quiz on Friday, includes today but not Wednesday
- lab assistants in CMC 102 from 6pm until 9pm or later Sunday–Thursday

▼ Making change

- an example of where `//` and `%` can be useful

▼ Practice: order the following lines to correctly print the area of a circle with radius `r`

- ```
1 print("The area of the circle is", area)
2 area = math.pi * r**2
3 import math
4 r = 7.5
```

- ▼ Consider the documentation for `math.pow`

- `math.pow(x, y)`  
Return `x` raised to the power `y`.
- How would we change the above code to use `math.pow` instead of `**`

## ▼ Functions!

- ▼ returning to the `get_CCD_temp()` mystery
  - it's called a **function**, not like in math class
  - separates definition and execution (you will see what this means in a moment)
- ▼ why should Current Temperature Inc. have all the fun? Let's make our own function!
  - ▼ **C\_to\_F** function
    - start with diagram of input, output, operations
    - ▼ all **definitions** start out the same way (name, parameters, colon)
      - `def <function name>(<parameter1>, <parameter2>, ...):`
      - function names follow the same rules as variable names
    - ▼ **parameters**
      - number of parameters and their order
      - the labels they get inside the function
    - indent determines function body
    - ▼ using a function looks like  
`<function name>(<value for parameter1>, <value for parameter2>, ...)`
      - referred to as a **function call**
    - ▼ Aaron's function manifesto
      - only operate on parameters
    - ▼ always **return** result at the end
      - marks the end of a function and determines its value

- what happens if these laws are disobeyed?
- ▼ class discussion: why might we want to use functions?
  - Our weather app won't convert from Fahrenheit to Celsius just once, but many many times
- ▼ quick check: define a function that takes the radius and returns the area of a circle, use the function to print the area of a circle of radius 10
  - ```
import math
def circle_area(radius):
    area = math.pi * radius**2
    return area
print("Area of a circle with radius 10 is", circle_area(10))
```

▼ Decision time

- ▼ we can do this with Python's **if**
 - ▼ anatomy of if
 - if <condition>:
 - <indented lines only executed when condition is True>
 - need to make program behavior **conditional** on state of the world
 - diagram possible states and actions
 - ▼ let's say text has more than two settings (yellow, orange, red)
 - **if/elif/else** nice way to structure this decision
 - ▼ **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 $x \leq 10$
- ▼ 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
```

## ▼ Extras

- randomness
- numeric data types