

▼ Preamble

- ▼ I'm excited to share my passion for CS with you
 - but you don't need to be a programmer or love programming to find it a handy tool to have in your toolbox
 - preview labs
 - POLL: what are the two things the CPU does in our notional machine?

▼ Problem: what icon to display in weather app?

- Computer needs to make a decision at a particular point in time
- ▼ Preparation for future learning
 - diagram what computer needs to know (input) and what the possible outcomes could be (output)
 - sketch the operations computer must do and their ordering
 - computer scientists borrowed a lot of math terms and symbols for only somewhat similar concepts
- ▼ Simplified problem: difference between current and “hot” temperature (i.e., should the app show the current temperature in **red**)
 - outline: get input data, calculate difference, output result
 - ▼ have the values we want: `get_cur_temp()` and **80 F**
 - ▼ don't worry about exactly what's going on with `get_cur_temp()`
 - we'll get into it next week, for now, abstraction!

- ▼ need to make these values available and give them useful names—assignment!

- ▼ **variables**

- = and variables have different meaning than in math
- ▼ assignment statement: `<variable name> = <expression>`
 - `cur_temp = get_cur_temp()`
 - `hot_temp = 80`
- assignment copies value to memory and gives it a label (the variable name)

- ▼ variable names follow certain rules

- must begin with a letter or an underscore, can only contain letters, number, and underscores
- no spaces
- case sensitive (e.g., `cur_temp` and `Cur_temp` are different names)

- ▼ to make variable names easier to read, convention is to use underscores or capitalization

- `cur_temp` or `curTemp`
- *snake case* and *camel case*

- ▼ just got off the phone with Current Temperature Inc., `get_cur_temp()` is in C

- ▼ **arithmetic**

- ▼ Python provides the standard mathematical operators

- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (remainder or modulo)
- ** (exponent)

- ▼ Follows the order of operations you might have seen in math class

- PEMDAS
- parens, exponents, multiplication & division, addition & subtraction
- minus signs happen **after** exponents, so $-2^{**}2$ evaluates to -4
- POLL: $3 * 5 - 10^{**}2 \% 9$
- $C = (F - 32) * 5 / 9$
- ▼ $cur_temp_F = (cur_temp - 32) * 5 / 9$
 - when a variable appears on the right side of an =, it is interpreted as the value it labels in memory
 - when a variable appears on the left side of an =, it is the label where the value computed on the right side will be stored
- *quick check: write Python code to compute a lower bound on hours I spent watching Fellowship of the Ring (2 hours 58 minutes) when it came out on VHS (I watched it once per day for a week), the final result should be assigned to a variable*
- ▼ **calculate difference**
 - $difference = hot_temp - cur_temp_F$
- **make a diagram of memory after each step (line of code)**
- ▼ **diagram the following steps**
 - ▼ $x = 10$
 $y = x$
 $x = x + 5$
 - y and x label separate locations in memory, the second step copies the value stored as x to a new location and labels it y
 - so the third line overwrites the value for x, but not for y
- ▼ **[NOW ON SCREEN] output result (three mistakes: missing “, missing), typo in variable name)**
 - Forgetting a close “ around text causes a SyntaxError
- ▼ **anatomy of the error message**
 - File where the error occurred, line where the error occurred, and the error type and message
 - it may feel like it’s Python yelling at you for your mistakes

- really Python catching on fire and saying “I don’t understand, aaaaaaaahhhhhhhhhh.”
- ▼ **how would you demonstrate to me this is correct (or incorrect)**
 - print everything, check for known results
- ▼ **compare to one-line version**
 - `print(“difference between hot and current temp in Celsius:”,
(80 - 32) / 9 * 5 - get_cur_temp())`
- ▼ **comments**
 - any line that starts with a # is a *comment*
 - ignored by Python, useful for explaining and documenting your code