

how i learned to stop worrying and love to code

when do things go wrong?

before my code runs	at runtime	in my results
<p>what kind of error do i observe?</p> <ul style="list-style-type: none">▶ SyntaxError<ul style="list-style-type: none">▪ Check earlier lines for incomplete function statements or lists.▪ Make sure that paired language features have both of the pair:<ul style="list-style-type: none">▪ close strings ""▪ close parentheses ()▪ close brackets and braces▪ Make sure blocks begin with :.▶ IndentationError<ul style="list-style-type: none">▪ Use four spaces, not two or tabs.	<p>what kind of error do i observe?</p> <ul style="list-style-type: none">▶ NameError<ul style="list-style-type: none">▪ Check for missing variable or function definitions or imports.▪ Check for typos in variable names.▶ TypeError<ul style="list-style-type: none">▪ Check expression and variable types (coerce if necessary).▶ IndexError<ul style="list-style-type: none">▪ Don't refer to nonexistent list elements.▶ KeyError<ul style="list-style-type: none">▪ Don't refer to nonexistent dict keys.▶ ZeroDivisionError, etc.<ul style="list-style-type: none">▪ Infer from the error what has gone wrong.▪ Search online or ask a T.A.▶ does a function or method return a value? should it?<ul style="list-style-type: none">▪ Distinguish between output (print) and return value.▶ does an infinite loop occur?<ul style="list-style-type: none">▪ Press Ctrl + C to break out.▪ Check that loop conditions allow exit.	<p>how do i know the code is wrong?</p> <ul style="list-style-type: none">▶ AssertionError<ul style="list-style-type: none">▪ Print actual and expected results.▪ Figure out if issue is in format or type, or in the calculation.▪ Try Strategy #1.▶ some values work, some don't<ul style="list-style-type: none">▪ Check for edge cases (< v. <=).▪ Make sure right statements are inside of loop.▪ Check the range or loop exit conditions.▶ accumulator values change when the loop is run again<ul style="list-style-type: none">▪ Make sure that any accumulators are re-initialized.
<p>strategies</p> <ul style="list-style-type: none">#1 Re-read the problem statement carefully. Construct the expected output and compare to actual results. Make a simple test case.#2 Add <code>print</code> statements generously. Show results of intermediate variables and variables inside of loops.#3 Chart the flow of the program. Draw blocks and arrows, or print the code out and mark on it what happens.#4 Add comments to each part of the code to explain its behavior. Look for discrepancies. Explain it to someone else.#5 Run the code manually, if not too complicated. See where the program behavior diverges from expectations.#6 Make no assumptions! If your thinking is not precise, your code will not be precise.#7 Start over from scratch. Take a fresh look at the problem.	<p>reading tracebacks</p> <ul style="list-style-type: none">▪ Read from bottom to top: the first function which raised an exception is at the bottom. You normally only need the last 2-3 lines.▪ Read the message and <i>think</i> about it.▪ Use the given line number in the file to locate the bug (also the ^).▪ Repeat for next error up.	<p>common coding errors</p> <ul style="list-style-type: none">#1 The operation or method is not supported. (There is no <code>list.split</code> method.)#2 The expected behavior of operations or methods is not what you think it is. (<code>3*3 v. "3"*3; = v. ==</code>)#3 Variables get clobbered by mutable methods. (<code>x = x.sort()</code>)#4 A built-in function with a common name is overwritten. (<code>list = []</code>)#5 Variable names are inconsistent. (One line refers to <code>data</code>, the next to <code>entries</code>.)#6 Blocks aren't properly indented, or key statements are outside of the loop.#7 Input data aren't cleaned up—<code>strip</code>, remove %, convert to <code>float</code>, etc.