Submitted by **bikingman**
over 11 years

"Unindent does not match any outer indentation level"

Can anyone explain to me what this message actually means? I have fixed many a problem after getting this error message, mostly by dumb luck. Often I indent the offending line another way and get a new error message like "so and so not defined".

What process does a good coder follow when he or she gets a message lijke that?

---

0 votes

Permalink

If you post your code, we can try to find the problem. So that we can see the indentation you used, be sure to format it according to the instructions that you can see be clicking the link that looks like this:

Show formatting instructions ▼

Indentation problems often occur when the code is not indented that is intended to be controlled by a function header, a loop header, an `if` block header, or another type of header. The controlled block of code needs to be indented one level to the right, with respect to its header.

One significant problem is the mixing of tabs and spaces for indenting code. Use either tabs or spaces, but not both. Visually, code indented with tabs or spaces is indistinguishable, but you can check the spacing by clicking on it with your mouse to discern the width of the whitespace units that are in the code.

Most Python editors have a default mode of spacing, and include some degree of automatic indentation formatting. Codecademy appears to be using spaces, at present, and automatically

converts tabs used for formatting into four spaces. The IDLE Python editor that you might have on your local computer includes menu items that help with indenting.

Sometimes, when I cannot tell whether a sample of Python code on Codecademy or elsewhere has mixed spaces and tabs, I'll remove the indentation from each indented line, starting from the beginning, remove the line feed at the beginning of the line, then hit `Enter` to have Codecademy provide default indentation. Corrections concerning the level of indentation may have to be made along the way, though, so that each line belongs to the proper block. You can experiment with this on Codecademy to find out what works best for you.

Submitted by AppylPye
over 11 years

0 votes

Permalink

OK, here is my latest code starting with lines 11 through 14 which passed on the last exercise: class PartTimeEmployee(Employee): def calculate_wage(self,hours): self.hours=hours return hours * 12.00 def full_time_wage(self,hours): self.hours=hours return super(PartTimeEmployee,self).calculate_wage(self,hours) Milton = PartTimeEmpolyee("Milton")

print full_time_wage(milton,10)

and that returns:

"syntax error. return outside function"

That's my latest code. It doesn't live up to the original title of this thread but I still have a problem getting my brain around what these error messages mean and what you think about when you get certan ones (especially ones like "expected indent, unexpected indent, not defined, etc.). I get those all the time. I understand the indent from the standpoint of if and else statements in a function. You do this if it's inside the function. That makes sense. But how the indent determines whether or not something is defined or not (when I can see a statement I just wrote that says "def xyz"), that just baffles me.

Submitted by bikingman
over 11 years

0 votes

Permalink

Thank you Glenn, I finally solved the problem. I always try to use a hard return when possible and I always use four spaces per indent. That was really good advice you gave about that. But sometimes the error implicates indents when I think the interpreter is confused by some other bug. You fix the indent like it likes and that reveals some subtler flaw that makes it think something is not defined for example when you can see in the text that it is defined (at least in the human sense of a statement like def xyz:

What got me through this one was changing my return super code to ... (PartTimeEmployee, self).calculate_wage(hours)

[instead of

0 votes

instead of (self, hours) and I also changed the last line to print milton.full_time_wage(10)

0 votes

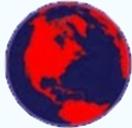And I forgot to ask Glenn: "How the heck did you ever amass 7177 points?(!)

0 votes

In part, the points are from maintaining this list:

A List of Great Beta Python Courses

Codecademy offers lots of content that users have contributed beyond the standard tracks.

Popular free courses

Free course

Learn SQL

In this SQL course, you'll learn how to manage large datasets and analyze real data using the standard data management language.

Beginner Friendly   **4** Lessons

Free course

Learn JavaScript

Learn how to use JavaScript — a powerful and flexible programming language for adding website interactivity.

Beginner Friendly   **11** Lessons

Free course

Learn HTML

Start at the beginning by learning HTML basics — an important foundation for building and editing web pages.

Beginner Friendly   **6** Lessons

Explore full catalog