

Pylint Changes

Specific Changes:

Maximum number of characters on a single line.

max-line-length=150

- We changed the maximum length of a single line of code to be 150 characters instead of 100. We felt that when there is too much shortening of a line, as is the case with the base presets of pylint, code can become a bit jumbled on the screen and harder to read. We feel this change makes it easier for coders to understand each other's work, even if that sometimes means scrolling slightly horizontally when they would not have before.
- There are multiple examples of this in our project/

Minimum line length for functions/classes that require docstrings, shorter

ones are exempt.

docstring-min-length=10

- We changed the minimum docstring length to be 10 characters, as opposed to -1 where it started. This means that no person can write a short docstring, likely just to fulfill the pylint requirement, that doesn't have enough characters to actually explain the function. This makes it so lazy coders can't get away with bad docstrings that will punish new coders later.

Ex: "Test for Function 2: hotel_stay" rather than "Test"

Naming style matching correct argument names.

argument-naming-style=camelCase

Naming style matching correct function names.

function-naming-style=camelCase

- Changed argument and function naming style from snake_case to camelCase. This is because camelCase is a preferred style by our code team and we believe it will help distinguish the arguments in a more compact way. Deciding between snake_case and camelCase is the less important aspect of this, compared to just having one in general

Maximum number of locals for function / method body.

max-locals=20

- Changed the maximum number of local variables from 15 to 20. This was because although making sure that functions aren't too long and use sub-functions when it makes sense, we felt limiting ourselves to 15 local variables was too restrictive and a negative to our coding rather than a positive. With proper spacing and logical uses of local

variables (ex. not reusing code multiple times with local variables), this rule made our coding process much easier.

- Ex. Used in hotel_stay.py function “guest_arrival”

Maximum number of parents for a class (see R0901).

max-parents=4

- Changed the maximum number of parents for a class to be 4, rather than the 7 it started. We felt that 7 parent classes was simply too much for logical code, and as such, have lowered it to ensure the inherited code did not get too convoluted.

Maximum number of nested blocks.

max-nested-blocks=4

- Reduced the maximum number of nested blocks from 5 to 4. Limiting nested blocks helps maintain code readability and prevents excessive indentation levels, promoting clearer code structure and easier debugging.
- Nesting is used in most functions, but not to an extraordinary extent, as this rule promotes.

Line similarities.

min-similarity-lines=6

- Adjusted the maximum line similarities from 5 to 6. This change allows for more tolerance in detecting similar lines, accommodating code patterns such as repetitive imports or function calls without triggering unnecessary warnings.

Maximum spelling mistakes suggestions.

max-spelling-suggestions=8

- Increased the maximum spelling mistake suggestions from 4 to 8. This adjustment provides a higher threshold for spelling checks, reducing false positives and improving productivity by minimizing interruptions during code review.
- We had to use funky words like the words produced by localizers in this project, so more tolerance was necessary.

Words that should not be checked in spelling.

spelling-dict-words=conmattcon

- Added "conmattcon" to the dictionary of words that should not be checked during spelling checks. This ensures that specific project-related terms or identifiers are not flagged as spelling mistakes, reducing noise in spell checking results. This word is the combination of all of the key contributors names, and as such, is a necessary word for giving adequate credit in our project.

Maximum number of statements in a function.

max-statements=40

- Changed the maximum number of statements in a function from 30 to 40. This adjustment allows for more comprehensive functions while still encouraging modular and readable code. It strikes a balance between function complexity and maintainability.
- `guest_arrival()` is a function that uses many statements, but not too many.

Maximum number of returns in a function.

max-returns=5

- Modified the maximum number of returns in a function from 4 to 5. This change permits slightly more flexibility in function design, accommodating scenarios where multiple return statements may enhance code clarity and expressiveness.