Python Coding Style

Inspired by the official PEP8 style guide https://pep8.org/

Keypoints:
- Use 4 spaces per indentation level.
- Spaces are the preferred indentation method rather than tabs.
- Imports should usually be on separate lines.
- Don't use spaces around the = sign when used to indicate a keyword argument or a default parameter value.
- Inline comments should be separated by at least two spaces from the statement. They should start with a # and a single space.
- Class names should normally use the CapWords convention
- Function names should be lowercase, with words separated by underscores as necessary to improve readability.
- Use one leading underscore only for non-public methods and instance variables.
- Always use self for the first argument to instance methods
- Constants are usually defined on a module level and written in all capital letters with underscores separating words (e.g. MAX_NUM)

HTML Coding Style
- Templating system should be used intelligently, which means:
  - Prefer extending general templates when possible
  - Break down page elements into components as frequently as possible
  - Keep each single template page clean
- Avoid inline styles, prefer stylesheets instead
- Use css classes when possible, use IDs just when needed (for example unique page elements such as **map** or **event-form**)

CSS Coding Style

- Avoid using CSS stylesheets, prefer SASS instead (and pack it through the provided toolchain i.e. webpack)
- Selector names should be lowercase and bash-separated (for example **.event-owner** or **.map-tooltip-description**)
- Selector names should be as "nestable" as possible (use the & prefix in SASS to inherit the prefix and nest selectors)
- Create new **.scss** files when defining stylesheets for new components not previously styled
- Prevent styling the same selector in different files/nested classes

JS Coding Style
- Do not include additional .js files in the templates, extend the current ones through the supplied toolchain (webpack)

- **.js** files should be <u>required</u> in **app.js** file in an intelligent way (i.e. do not require a .js file that revolves around a certain html element if that element is not present in the current page)
- Function and variable names should be lower camel-case (with the first character in lowercase)
- Adhere to EcmaScript 6 standard (i.e. *let* and *const* instead of *var*, no semicolons, etc…) and prefer that over jQuery when possible