

Normativa de código



Samuel Fernández Fernández (100432070)

Alejandro Mokov Ivanchov (100432094)

Índice

Normativa de código

3

Normativa de código

Cambios aplicados:

- `module-naming-style`: Hemos cambiado de `snake_case` a `PascalCase` para un mejor estilo del nombre del archivo.
- `variable-naming-style`: Hemos cambiado de `snake_case` a `any` para que no haya restricciones a la hora de nombrar las variables.
- `method-naming-style`: Hemos cambiado de `snake_case` a `UPPER_CASE` para diferenciar entre métodos y funciones por el estilo de su nombre.
- `argument-naming-style`: Hemos cambiado de `snake_case` a `camelCase` para que los argumentos sean reconocibles por el estilo de su nombre.
- `function-naming-style`: Hemos cambiado de `snake_case` a `any` para que no haya restricciones a la hora de nombrar las funciones.
- `max-line-length`: Hemos cambiado de 100 a 200 la longitud de la línea para que, en caso de un programa complejo, el programador tenga más espacio para escribir en una misma línea.
- `max-module-lines`: Hemos cambiado de 1000 a 1500 para que el programador tenga más líneas en caso de hacer un programa muy largo.
- `attr-naming-style`: Hemos cambiado de `snake_case` a `camelCase` para que se sepa que es un atributo por la forma en la que está escrito.
- `class-attribute-naming-style`: Hemos cambiado de `any` a `UPPER_CASE` para diferenciar los atributos de las clases del resto de atributos.
- `class-const-naming-style`: Hemos cambiado de `UPPER_CASE` a `any` para que no haya restricciones a la hora de nombrar las constantes de las clases.
- `const-naming-style`: Hemos cambiado de `UPPER_CASE` a `PascalCase` para que haya una diferencia entre las constantes de una clase y las de una función, por ejemplo.

-
- `include-naming-hint`: Hemos cambiado de No a Yes porque queremos que Pylint de una pequeña ayuda al programador cuando se equivoque con el estilo de nombramiento de cada elemento del programa.