Documentation style guide

This style guide helps us create organized and easy-to-read documentation. It provides guidelines for organization, spelling, formatting, and more.

These are guidelines rather than strict rules. Content is more important than formatting. You do not need to learn the entire style guide before contributing to documentation. Someone can always edit your material later to conform with this guide.

- Documentation is in markdown files with names formatted as `lowercase-with-dashes.md`.
  - Use an underscore in the filename only if the underscore is part of the topic name (e.g., `child_process`).
  - Some files, such as top-level markdown files, are exceptions.
- Documents should be word-wrapped at 80 characters.
- `.editorconfig` describes the preferred formatting.
  - A [plugin](#) is available for some editors to apply these rules.
- Check changes to documentation with `make test-doc -j` or `vcbuild test-doc`.
- [Use US spelling](#).
- [Use serial commas](#).
- Avoid first-person pronouns (*I*, *we*).
  - Exception: *we recommend foo* is preferable to *foo is recommended*.
- Use gender-neutral pronouns and gender-neutral plural nouns.
  - OK: *they*, *their*, *them*, *folks*, *people*, *developers*
  - NOT OK: *his*, *hers*, *him*, *her*, *guys*, *dudes*
- When combining wrapping elements (parentheses and quotes), place terminal punctuation:
  - Inside the wrapping element if the wrapping element contains a complete clause.
  - Outside of the wrapping element if the wrapping element contains only a fragment of a clause.
- Documents must start with a level-one heading.
- Prefer affixing links (`[a link][]`) to inlining links (`[a link](http://example.com)`).
- When documenting APIs, update the YAML comment associated with the API as appropriate. This is especially true when introducing or deprecating an API.
- When documenting APIs, every function should have a usage example or link to an example that uses the function.
- For code blocks:
  - Use [language](#)-aware fences. (` ```js `)

  - For the [info string](#), use one of the following.

Meaning	Info string
Bash	`bash`
C	`c`
C++	`cpp`
CoffeeScript	`coffee`
Diff	`diff`
HTTP	`http`

JavaScript	`js`
JSON	`json`
Markdown	`markdown`
Plaintext	`text`
Powershell	`powershell`
R	`r`
Shell Session	`console`

If one of your language-aware fences needs an info string that is not already on this list, you may use `text` until the grammar gets added to [remark-preset-lint-node](#).

- Code need not be complete. Treat code blocks as an illustration or aid to your point, not as complete running programs. If a complete running program is necessary, include it as an asset in `assets/code-examples` and link to it.

- When using underscores, asterisks, and backticks, please use backslash-escaping: `\_` , `\*` , and `\`` .
- Constructors should use PascalCase.
- Instances should use camelCase.
- Denote methods with parentheses: `socket.end()` instead of `socket.end` .
- Function arguments or object properties should use the following format:
  - `* `name` {type|type2} Optional description. **Default:** `value`.`
  - For example: `* ` `byteOffset` ` {integer} Index of first byte to expose.` **Default:** `0` .
  - The `type` should refer to a Node.js type or a [JavaScript type](#).
- Function returns should use the following format:
  - `* Returns: {type|type2} Optional description.`
  - E.g. `* Returns: {AsyncHook} A reference to` `asyncHook` .
- Use official styling for capitalization in products and projects.
  - OK: JavaScript, Google's V8
  - NOT OK: Javascript, Google's v8

- Use *Node.js* and not *Node*, *NodeJS*, or similar variants.
  - When referring to the executable, `node` is acceptable.

- [Be direct](#).

- When referring to a version of Node.js in prose, use *Node.js* and the version number. Do not prefix the version number with *v* in prose. This is to avoid confusion about whether *v8* refers to Node.js 8.x or the V8 JavaScript engine.
  - OK: *Node.js 14.x*, *Node.js 14.3.1*
  - NOT OK: *Node.js v14*
- [Use sentence-style capitalization for headings](#).

See also API documentation structure overview in [doctools README](#).

For topics not covered here, refer to the [Microsoft Writing Style Guide](#).