# Mclab Variable Naming Conventions

1. **All "units" SHOULD NOT use underbars to separate words contained within** (e.g. "probetype"). It's up to you to decide on what a "unit" is. When in doubt, use an underbar. Most importantly, BE CONSISTENT within your code and with respect to code others have written.
2. **All other words and all “units” SHOULD be separated by an underbar** (e.g. "p\_logsubs\_sex"). This improves readability.
3. **If variable length exceeds 32 characters, REMOVE ALL VOWELS from the variable name** (e.g. "this\_is\_a\_super\_very\_long\_ variable\_name" becomes "ths\_s\_a\_spr\_vry\_lng\_vrbl\_nm"). Removing vowels from a word does not significantly diminish readability. If a word becomes unreadable when vowels are removed, leave vowels in at your discretion (e.g. “hawaii”  “hawaii" or “hwi”, but probably not “hw”). If a word is all vowels, leave the word intact (e.g. “a”). Never remove vowels from the suffix “id”.
4. **All characters should be LOWER CASE.** (Exception – camelCase allowed for python). Upper case variables have special purposes in many languages, so avoid those unless they’re being used for that purpose.
5. **No special punctuation characters** (except underbar).
6. **If there are multiple versions of a variable, identify by using the syntax: 2 digit # followed by** (e.g. "baseread\_v1", "baseread\_v2", etc). Differentiation always goes at the END of the variable.
7. **IN DATASETS, Boolean ("yes/no", "true/false", etc) data identifiers should include a descriptive word or unit** (e.g. “constitutive”, “common”, “alternative”, etc). Do not use “question” prefixes. The “flag” descriptor must be used as a prefix, if used at all. Store a value of “0” for false/no and “1” for true/yes in Boolean variables.
8. **INSIDE SCRIPTS AND PROGRAMS, Boolean ("yes/no", "true/false", etc) variables should start with a “question” prefix** such as "is", "does", “can”, “should”, etc (e.g. "is\_valid\_read", "does\_probetype\_match", etc). Do not use the word “flag” in your Boolean variables—the “question” prefix is easier to read, and helps to avoid situations where two variable names are similar (e.g. “probetype” and “probetype\_flag”). Store a value of “0” for false/no and “1” for true/yes in Boolean variables. Example flag\_female a 1 is female.
9. **IN DATASETS, multivariable data identifiers should include a descriptive word or unit**. Do not use “question” prefixes. The “ind” descriptor must be used as a prefix followed by the # of categories and the descriptive word/unit.
10. **Avoid ambiguous variable names** (e.g. "temp", "array\_read", etc).
11. **Avoid one character variables EXCEPT as iterators in loops** (using them as parts of combining units, e.g. “p\_logsubs\_sex”, is OK).
12. **Variable names should be descriptive, but not verbose.**
13. **Avoid using abbreviations.** Some exceptions: “nucleic acid”  “nacid”, “identifier”  “id”, “background subtracted mean”  “bgsubmean”.
14. **Do NOT use plurals in variable names.** Arrays should use the singular form of the items they contain (e.g. “@gene”, not “@genes”). Exception – lists in python
15. **If a variable requires an unusual name, document it in the code comments**

**Mclab VARIABLE NAMING CONVENTIONS**

1. All "units" **SHOULD NOT** use underbars to separate words contained within (e.g. "probetype").
2. All other words and all “units” **SHOULD** be separated by an **underbar** (e.g. "p\_logsubs\_sex").
3. If variable length exceeds 32 characters, **REMOVE ALL VOWELS** from the variable name.
4. All characters should be **LOWER CASE**.
5. No special punctuation characters (except underbar).
6. If there are multiple versions of a variable, identify by using the syntax **varname\_identifier**.
7. **IN DATASETS**, Boolean data identifiers should include a descriptive word or unit (e.g. “constitutive”).
8. **INSIDE SCRIPTS AND PROGRAMS**, Boolean variables should start with a “question” prefix such as "is", "does", “can”, “should”, etc (e.g. “is\_valid\_read”).
9. Avoid ambiguous variable names (e.g. "temp", "array\_read").
10. Avoid one character variables **EXCEPT** as iterators in loops (using them as parts of combining units, e.g. “p\_logsubs\_sex”, is OK).
11. Variable names should be descriptive, but not verbose.
12. Avoid using abbreviations.
13. Do **NOT** use plurals in variable names.
14. If a variable requires an unusual name, document it in the code comments.