

Function Use & Declaration

- Use a single space between function name and opening parenthesis as well as between separating semicolons and within parentheses. This facilitates easy selections within both Windows and Macintosh when double-clicking to highlight

| | |
|-------------------------------------|------|
| Function Name (with ; parameters) | good |
| Function Name(with; parameters) | bad |

- Do not use a space at the end of lines of code closed by the semicolon.

| | |
|--|------|
| <pre>Let ([~replaced = Substitute (FunctionalArea » Tablename::fieldName ; "foo" ; "bar"); ~anotherVar = "literal string"]; ~replaced & ~anotherVar)</pre> | good |
| <pre>Let ([~replaced = Substitute (FunctionalArea » Tablename::fieldName ; "foo" ; "bar") ; ~anotherVar = "literal string"] ; ~replaced & ~anotherVar)</pre> | bad |

- Empty or no-parameter based custom functions can take the **optional** opening and closing parentheses despite FileMaker's ability to accept calculation code without them. Doing so does increase readability and enforces proper syntax checking by FileMaker when a similarly named variable, global or field may conflict.

Note: certain custom functions, such as **Null** are [Reserved elements](#) and do not need to follow this convention.

| | |
|--|------------|
| Let (~something = FunctionName () ; ~something) | good |
| Let (~something = FunctionName ; ~something) | acceptable |

Single platform development

In order to stay cross-platform compatible with regards to code editing, the space on either side of the semicolon facilitates easier parameter selection using the mouse. However, if you are working within a team of developers or all developers are working on the Macintosh, it is acceptable to remove the space between the parameter and the semicolon. This increases readability, but does force the developer to be familiar with using keyboard selection strategies, such as advancing to the next word and using the arrow keys and shift to navigate and select text.