



[Features](#) [Pricing](#) [Manual](#) [News](#) [Support](#)

Google Shell Style Guide Support

Google maintains a [comprehensive shell style guide](#). It defines how to write Bash scripts at Google. For everyone else, it's sound advice and a wonderful, clearly defined guide for better collaboration. If you want to establish a coding style for Bash scripts, then this is a great start.

BashSupport Pro provides a formatter preset and inspections to detect code not following the rules. All inspections provide quick fixes for the warnings they show in your editor. The coverage of the style guide will be extended in the future – for now the most important rules are implemented.

1. [Enabling Google Style Guide Support](#)
2. [Inspections](#)
 1. [File Name Style](#)
 2. [Function Comments](#)
 3. [Function Name Style](#)
 4. [Inconsistent Function Keyword Usage](#)
 5. [Inconsistent Function Parentheses Usage](#)
 6. [Missing main\(\) Function](#)
 7. [Incorrect main\(\) Declaration](#)
 8. [Shebang Definition](#)
 9. [Variable Name Style](#)

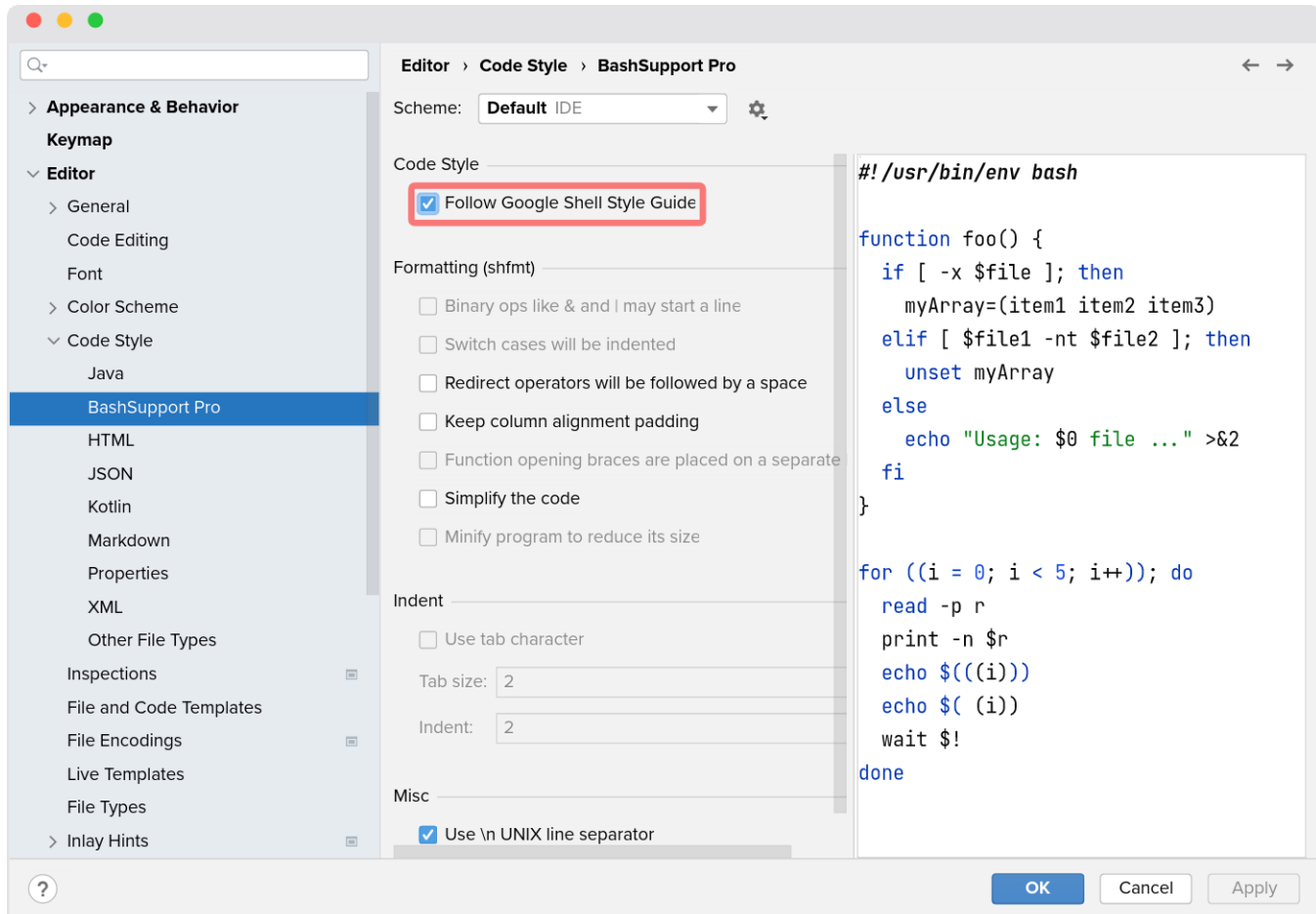
Enabling Google Style Guide Support

The support for the style guide is turned off by default. First, you need to enable it to

use it. You can find the settings at

Settings... → Editor → Code Style → BashSupport Pro.

Enabling the setting **Follow Google Shell Style Guide** turns on a formatter preset and also the inspections and quick fixes for the style guide.



Turning on the Google code style support

Inspections

The following inspections are only active with the enabled Google Style Guide support.

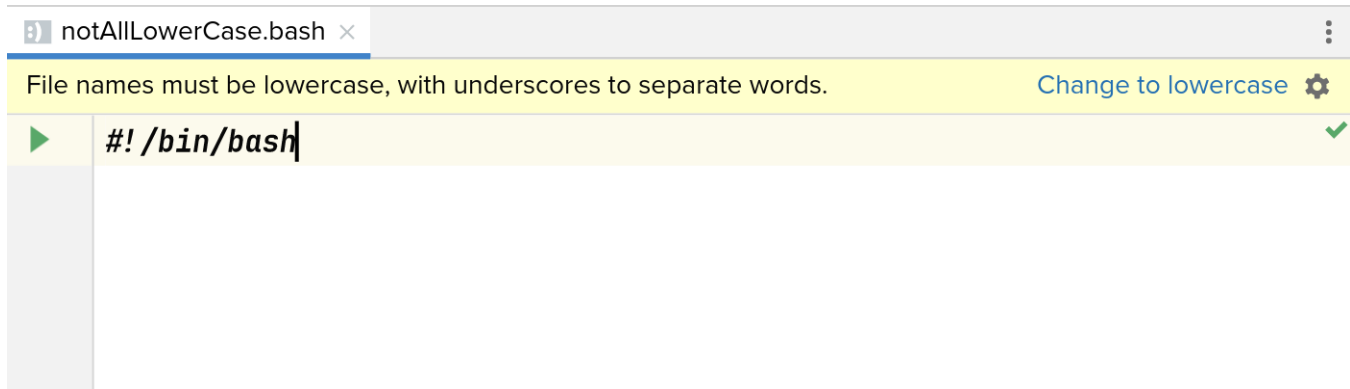
If you want to use an inspection without turning on the code style, then you can disable the inspection with configured Google shell code style” for this particular inspection. This setting is available for the style guide inspections at

Settings... → Editor → Inspections → BashSupport Pro → Google Shell

File Name Style

This inspection highlights incorrectly named files. A quick fix allows to automatically re-

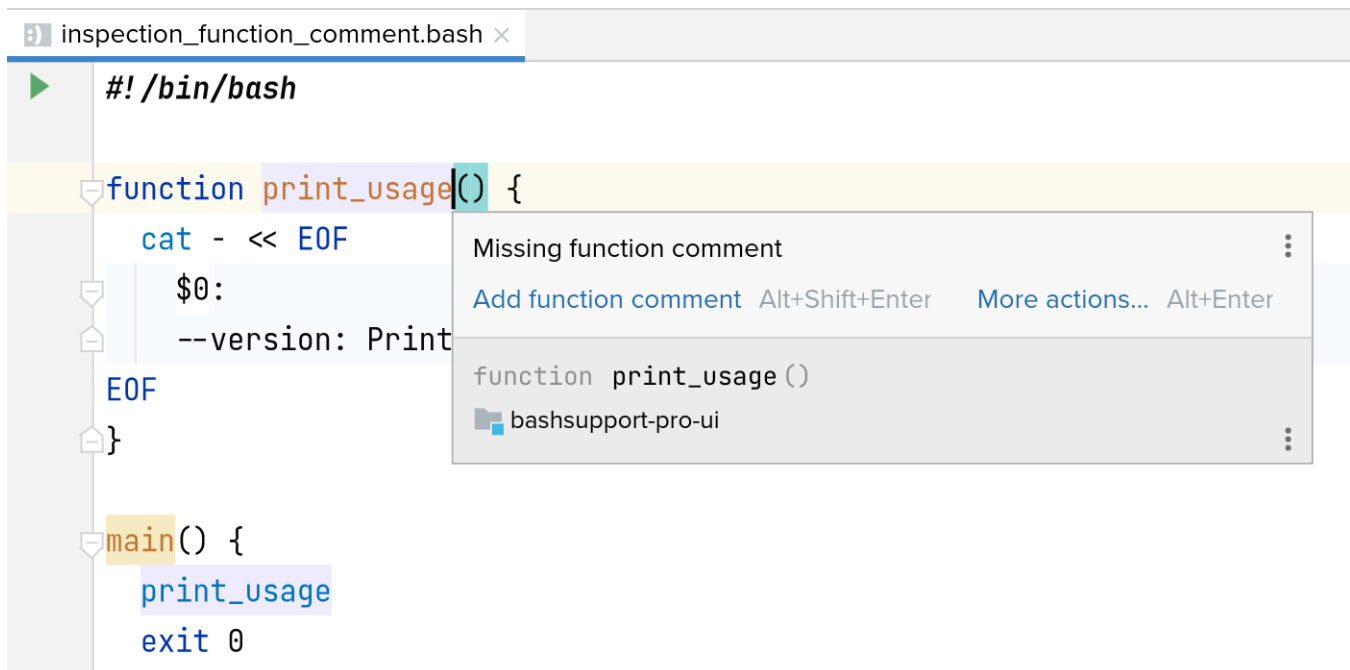
name the file and all references to it.



File name inspection

Function Comments

This inspection highlights functions, which don't have the required comments. A quick fix will automatically create it for you.

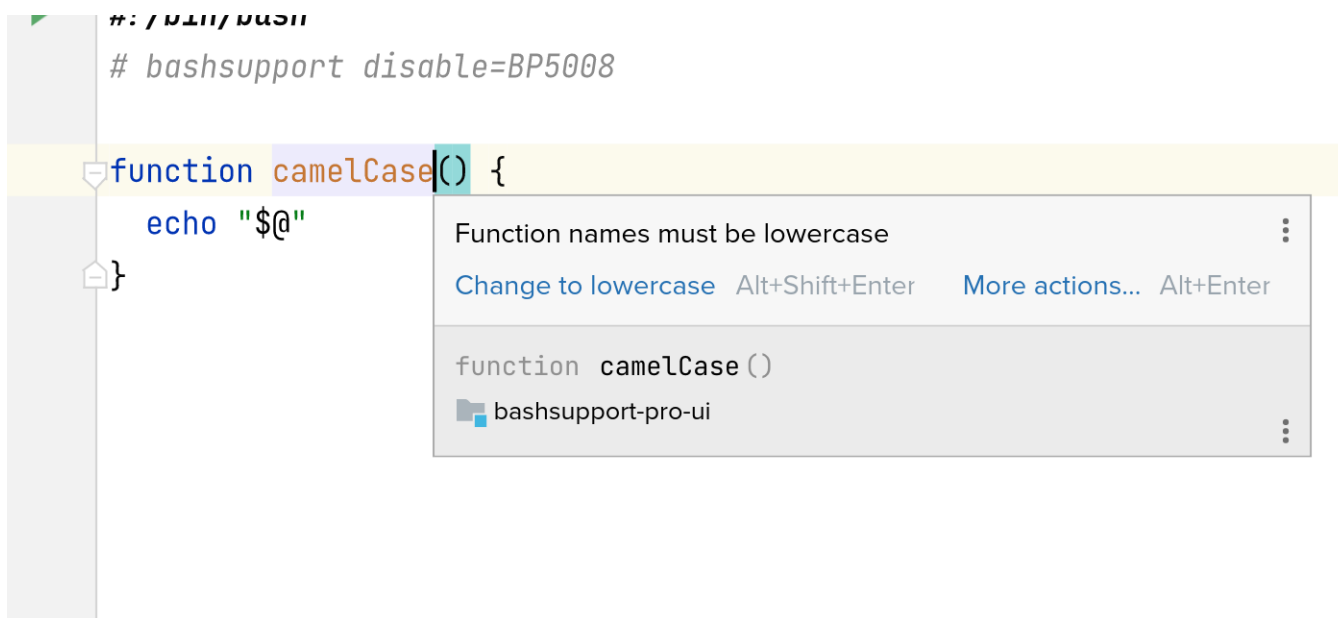


Function without the required comment

Function Name Style

This inspection highlights incorrectly named functions. A quick fix allows to rename the function and all references to it.

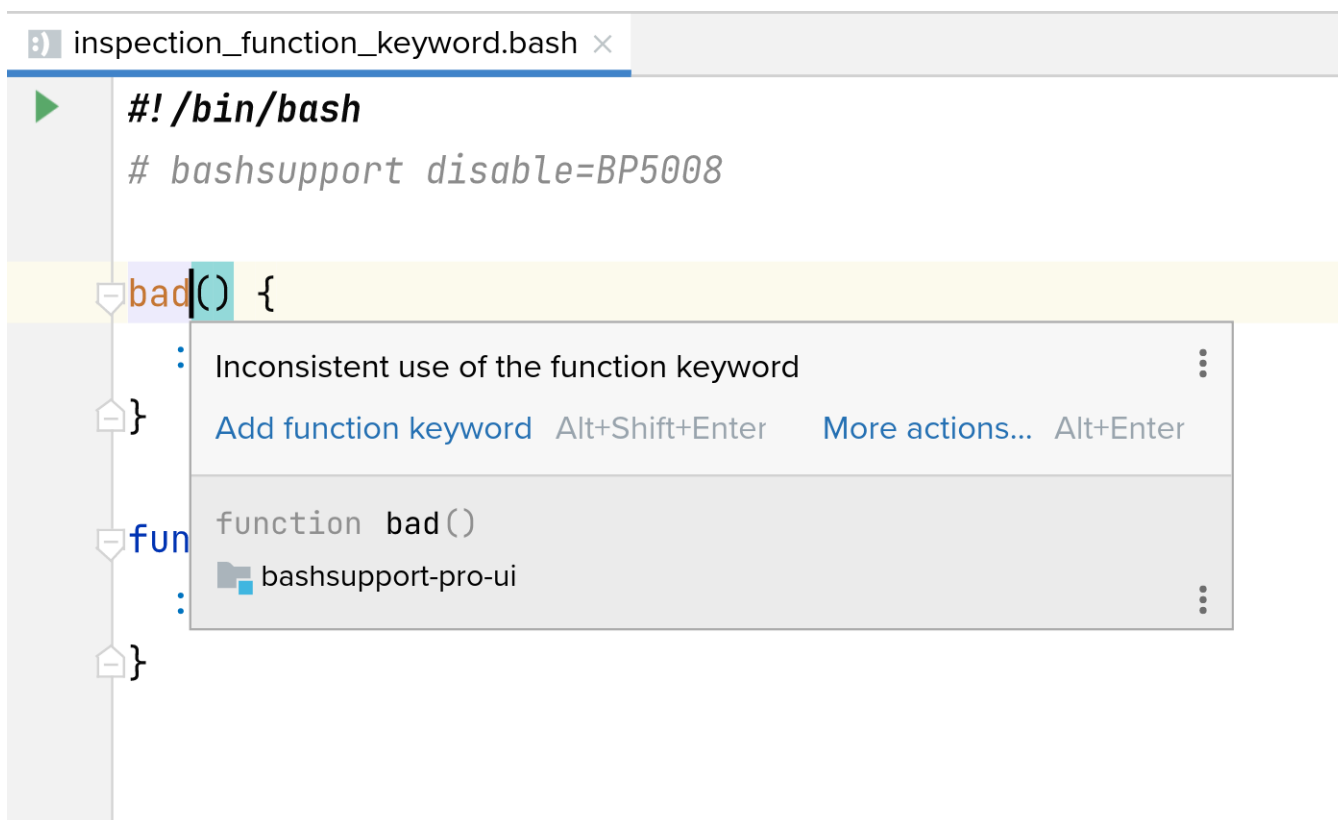




Incorrect function name

Inconsistent Function Keyword Usage

This inspection highlights inconsistent use of the `function` keyword. A quick fix allows to add the keyword.

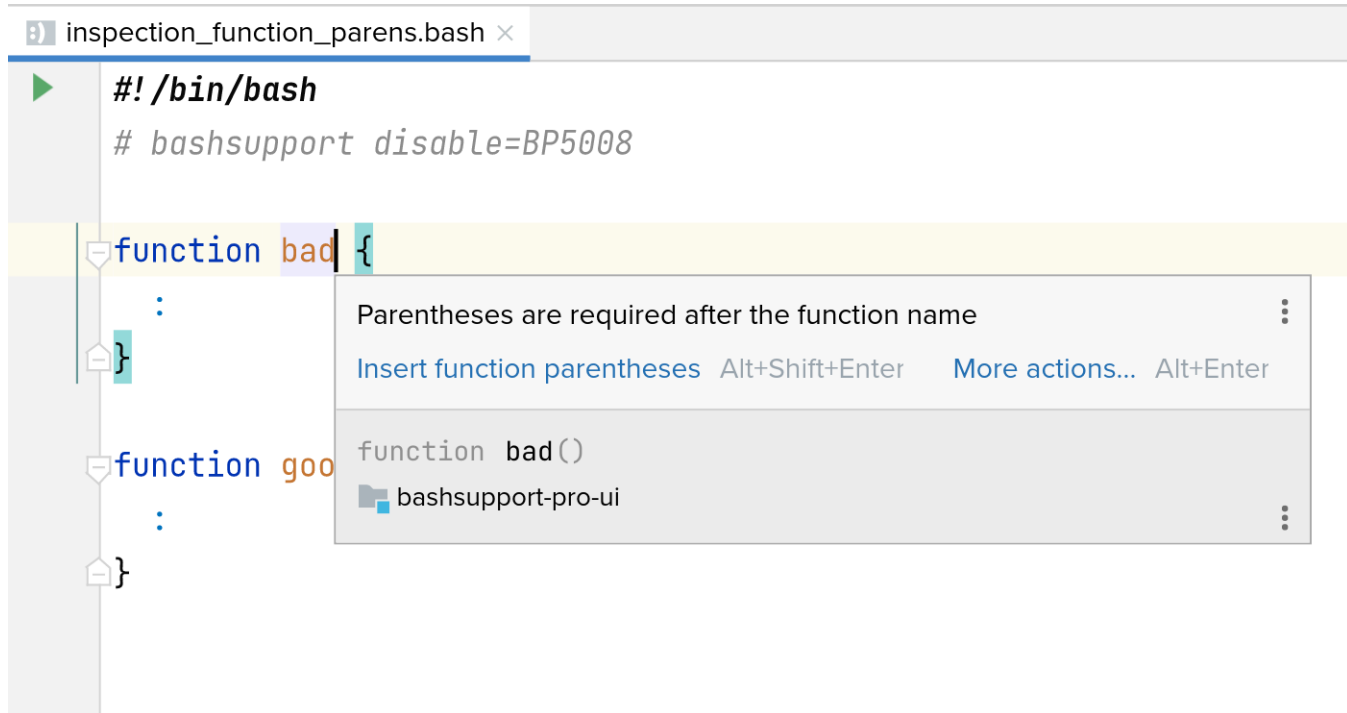


Inconsistent use of the function keyword

Inconsistent Function Parentheses Usage

Inconsistent Function Parentheses Usage

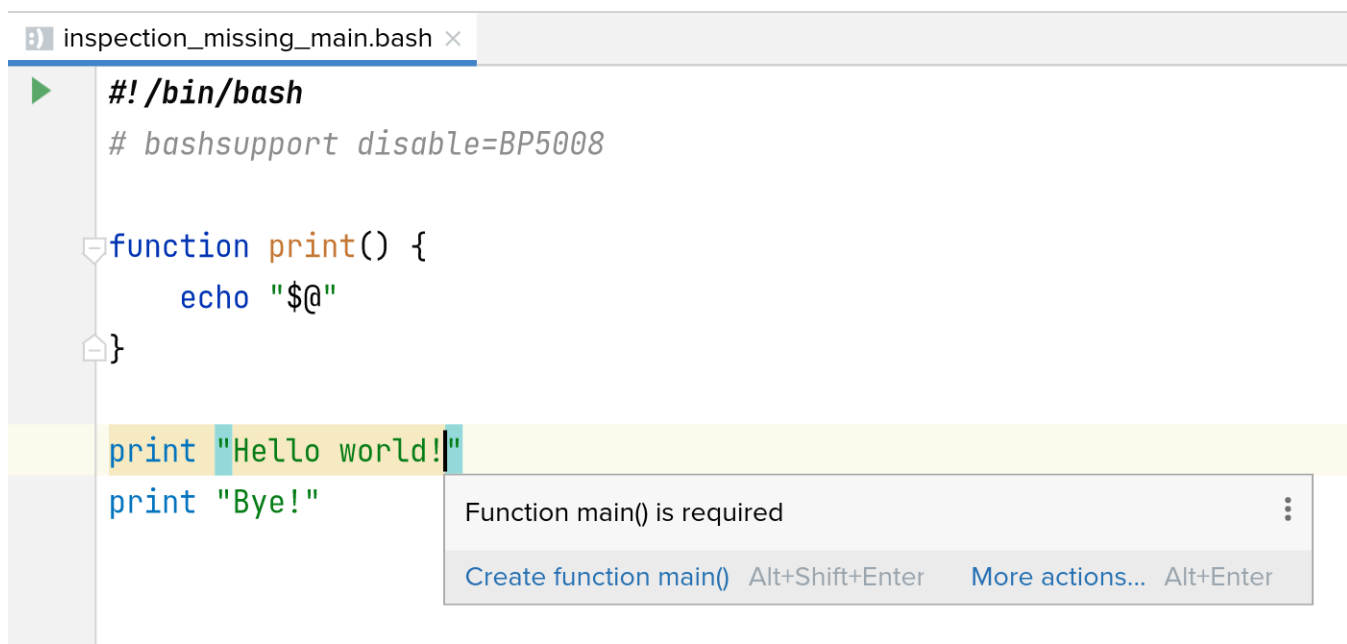
This inspection highlights inconsistent use of the `()` parentheses after a function name. A quick fix allows to add the parentheses to the file.



Inconsistent use of `()`

Missing main() Function

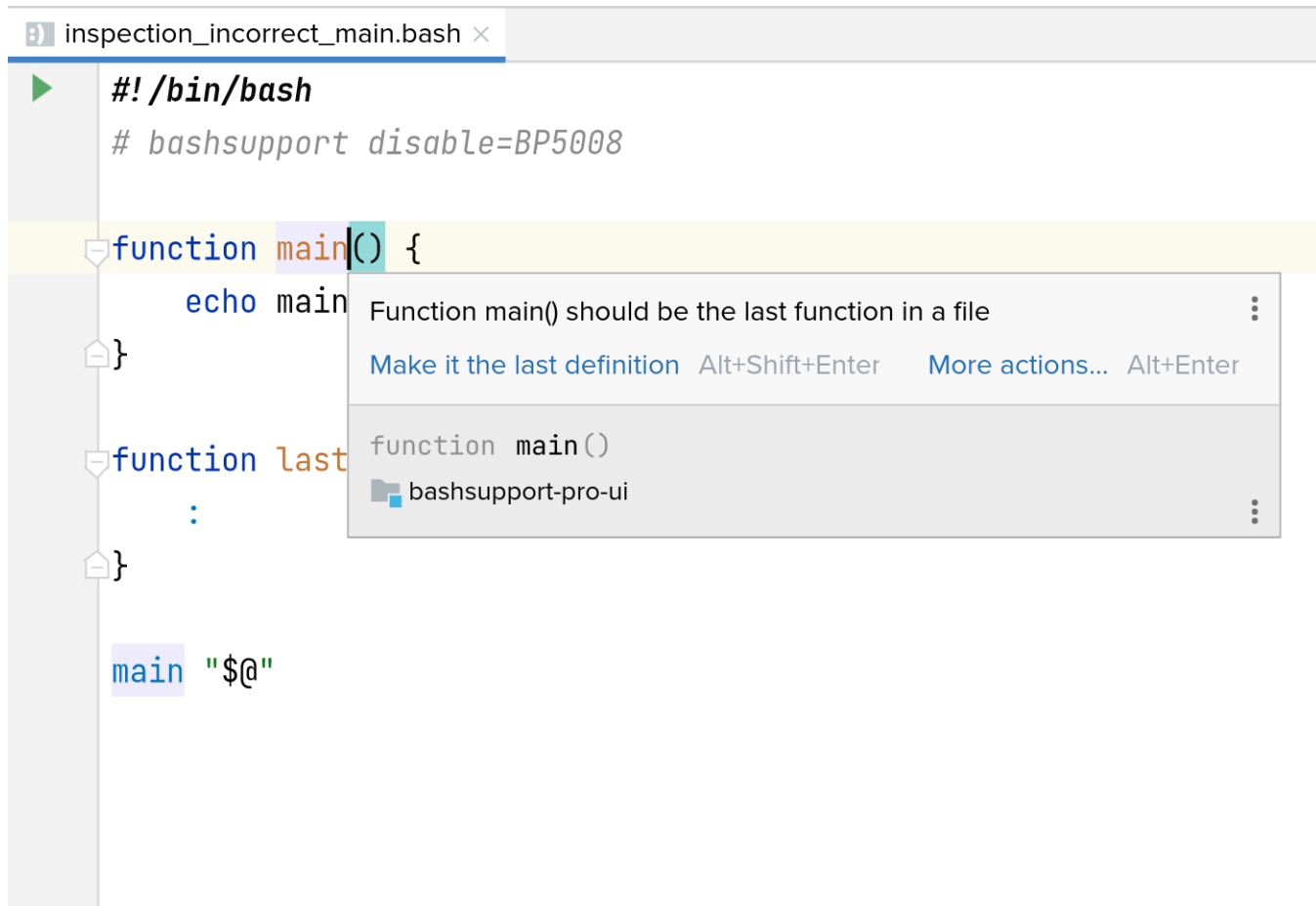
This inspection highlights the first instruction at the top-level of a file, which should be wrapped into a `main()` function. The provided quick fix will automatically create a `main()` function in the file and move all top-level instructions into it.



Instruction which should be inside main()

Incorrect main() Declaration

This inspection highlights declarations of `main()` which are declared at the wrong position in a file. A quick fix allows to move the declaration and all calls inside the file to the end of the file.



```
inspection_incorrect_main.bash x
#!/bin/bash
# bashsupport disable=BP5008

function main() {
    echo main
}

function last {
    :
}

main "$@"
```

Function main() should be the last function in a file

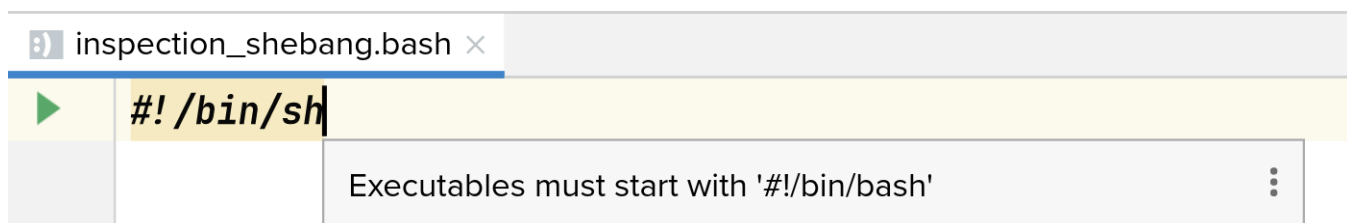
[Make it the last definition](#) Alt+Shift+Enter [More actions...](#) Alt+Enter

function main()
bashsupport-pro-ui

main() declared at the wrong position

Shebang Definition

This inspection highlights incorrectly defined shebang lines. A quick fix allows to change the shebang line to the required `#!/bin/bash`.



```
inspection_shebang.bash x
#!/bin/sh
```

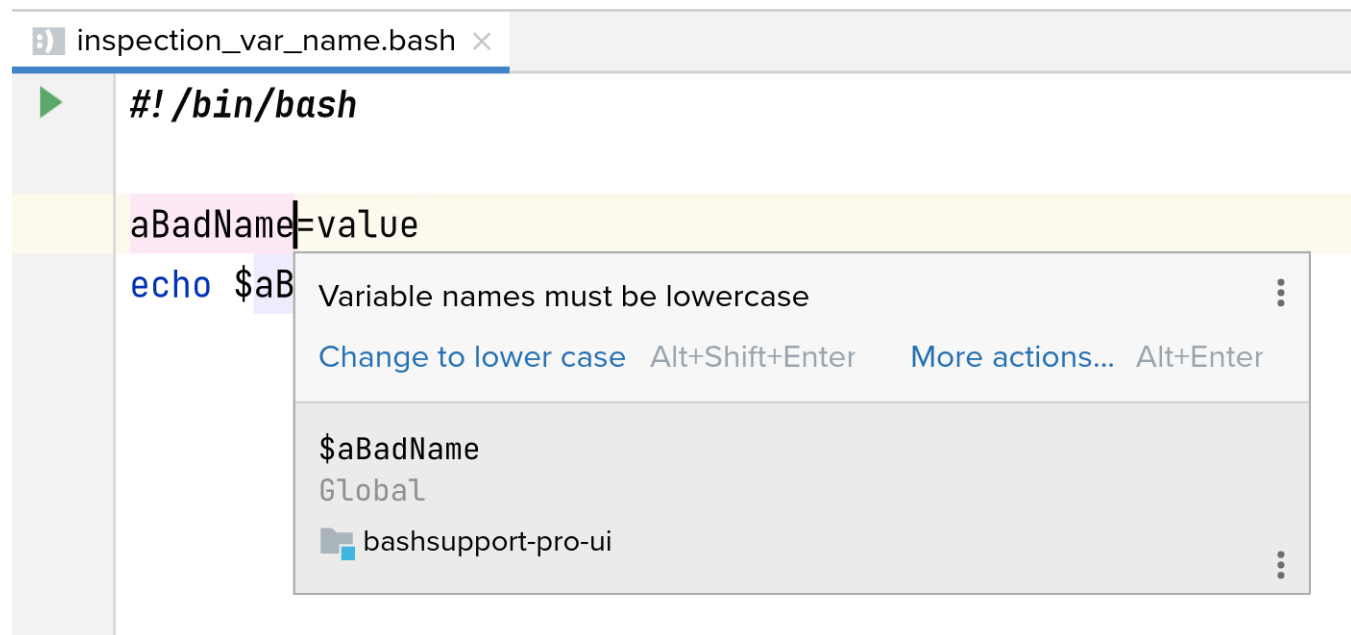
Executables must start with '#!/bin/bash'

Update shebang Alt+Shift+Enter More actions... Alt+Enter

Wrong shebang line

Variable Name Style

This inspection highlights bad variable names. A quick fix allows to change the variable name and all references to the required lowercase name.



Incorrectly named variable