

# ifallfalse – Compare string against set of strings

Dennis Chen  
proofprogram@gmail.com

v2.0.0, 2021/07/22\*

## Abstract

The `ifallfalse` package is a package that allows you to check whether a string is contained within another set of strings, and perform an action if it is not.

## 1 Usage

The package provides an `allfalse` environment and a macro `\orcheck` to be used inside the `allfalse` environment.

`allfalse` To set up an `allfalse` environment, simply write

```
\begin{allfalse}{string}{true branch}{false branch}

\end{allfalse}
```

`string` will be compared to the set of strings (which we will declare via `\orcheck`), and if `string` does not match the set of strings, `false branch` will be executed. Otherwise, `true branch` will be executed.

`\orcheck`

To add strings to the set that `string` will be compared to, we must write `\orcheck{setstring}` inside the corresponding `allfalse` environment. Then, `action` will not execute if `string` matches `setstring` or any arguments of previous `\orcheck` declarations.

If no `\orcheck` declarations exist, then `action` will always be executed.

### 1.1 Error Checking

The package checks whether the macro `\orcheck` is used inside an `allfalse` environment. If it is not, the package throws an error.

---

\*<https://github.com/chennisden/ifallfalse>

## 2 Example

Here is a simple example to demonstrate how `allfalse` is used.

```
\documentclass{minimal}
\usepackage{ifallfalse}

\begin{document}

\begin{allfalse}{purple}{}{This color is not red, blue, or green!}
  \orcheck{red}
  \orcheck{blue}
  \orcheck{green}
\end{allfalse}

\end{document}
```

In this case, because purple does not match red, blue, or green, the false branch — which is `This color is not red, blue, or green!` — will execute at that location inside the document.

## 3 Implementation

These are the implementation details of package `allfalse`. Because the package is so short, we can explain everything.

`allfalse` When setting up `allfalse`, we locally define the `\comparedstring` macro with the first argument that the environment takes in. This is what will be compared against all the strings passed in through the `\orcheck` declarations inside the environment.

Then, we define our body of logic (which we will be adding onto through `\orcheck`) to just initially consist of the action we would like to perform if `\comparedstring` matches none of the strings passed in through `\orcheck`.

Finally, we execute our `logicbody`, which will change `\ifallfalse@branch` to be false if appropriate. Then, the appropriate action will be executed.

```
1 \newenvironment{allfalse}[3]
2 {%
3   \newif\ifallfalse@branch\allfalse@branchtrue%
4   \def\comparedstring{#1}%
5   \def\trueaction{#2}%
6   \def\falseaction{#3}%
7   \def\logicbody{\allfalse@branchfalse}%
8 }
9 {%
10  \logicbody%
11  \ifallfalse@branch
12    \trueaction%
```

```

13  \else
14      \falseaction%
15  \fi
16 }

```

`\orcheck` We first save `allfalse` to a macro so we can use `\ifx` to compare the current environment name against it. If we can, then we add some following (somewhat convoluted) code to `\logicbody`. I will explain what each piece of it does, though not in the order the pieces of code appear.

- `\ifx\@currenvir\@allfalsename` evaluates to true if the current environment (whose name is saved to the macro `\@currenvir`) matches the name of `\@allfalsename`, or `allfalse`.
- If it evaluates to false, the package throws an error.
- The line `\pdfstrcmp{\comparedstring}{#1}=0` evaluates to true when put with `\ifnum` if the two arguments passed into `\pdfstrcmp` are equal, because `pdfstrcmp` compares their lexicographical order and returns 0 if the two strings are lexicographically equivalent.
- Thus, we can treat `\ifnum\pdfstrcmp{\comparedstring}{#1}=0` as an expression that evaluates to true if `\comparedstring` and `#1` match, and false otherwise.
- When all is said and done, the logic reduces to something of the form

```

\if\else
\if\else
\ldots \allfalse@branchfalse
\fi\ldots \fi

```

Logically, `\allfalse@branchfalse` will only execute if all the conditions are false; in other words, it will only execute if `\comparedstring` does not match any of the strings passed in via `\orcheck`. This is because each `\else` branch must execute.

```

17 \newcommand*\@allfalsename{allfalse}
18
19 \newcommand{\orcheck}[1]{
20     \ifx\@currenvir\@allfalsename
21         \protected@edef\logicbody{
22             \ifnum\pdfstrcmp{\comparedstring}{#1}=0\else\logicbody\fi
23         }
24     \else
25         \PackageError{ifallfalse}{
26             \protect\orcheck\space should be nested within the allfalse environment
27         }{}
28     \fi
29 }

```

## 4 Limitations

This package cannot be used in the lualatex engine.