

tf.contrib.framework.deprecated_args

```
deprecated_args(  
    date,  
    instructions,  
    *deprecated_arg_names_or_tuples,  
    **kwargs  
)
```

Defined in [tensorflow/python/util/deprecation.py](#).

See the guide: [Framework \(contrib\) > Deprecation](#)

Decorator for marking specific function arguments as deprecated.

This decorator logs a deprecation warning whenever the decorated function is called with the deprecated argument. It has the following format:

Calling (from) with is deprecated and will be removed after . Instructions for updating:

If **date** is None, 'after ' is replaced with 'in a future version'. includes the class name if it is a method.

It also edits the docstring of the function: ' (deprecated arguments)' is appended to the first line of the docstring and a deprecation notice is prepended to the rest of the docstring.

Args:

- **date** : String or None. The date the function is scheduled to be removed. Must be ISO 8601 (YYYY-MM-DD), or None.
- **instructions** : String. Instructions on how to update code using the deprecated function.
- ***deprecated_arg_names_or_tuples** : String or 2-Tuple(String, [ok_vals]). The string is the deprecated argument name. Optionally, an ok-value may be provided. If the user provided argument equals this value, the warning is suppressed.
- ****kwargs** : If **warn_once=False** is passed, every call with a deprecated argument will log a warning. The default behavior is to only warn the first time the function is called with any given deprecated argument. All other kwargs raise **ValueError** .

Returns:

Decorated function or method.

Raises:

- **ValueError** : If date is not None or in ISO 8601 format, instructions are empty, the deprecated arguments are not present in the function signature, the second element of a deprecated_tuple is not a list, or if a kwarg other than **warn_once** is passed.

Stay Connected

- Blog
- GitHub
- Twitter

Support

- Issue Tracker
- Release Notes
- Stack Overflow

English

Terms | Privacy