

---

# **tryagain Documentation**

***Release 0.9***

**Thomas Feldmann**

February 12, 2016



<b>1</b>	<b>Indices and tables</b>	<b>3</b>
----------	---------------------------	----------



```
tryagain.call(func, max_attempts=None, exceptions=<class 'Exception'>, wait=0.0,  
              cleanup_hook=None, pre_retry_hook=None)
```

**Parameters**

- **(callable)** (*func*) – The function to retry. No arguments are passed to this function. If your function requires arguments, consider defining a separate function or use `functools.partial` / a lambda function.
- **max\_attempts** – Any integer number to limit the maximum number of attempts. Set to `None` for unlimited retries.
- **exceptions** – An iterable of exceptions that should result in a retry.
- **wait** – This can be an integer / float to specify the waittime in seconds before the next attempt. You can also pass a function which accepts a single argument ‘attempt’.
- **cleanup\_hook** – Can be set to a callable and will be called after an exception is raised from calling *func*. No arguments are passed to this function. If your function requires arguments, consider defining a separate function or use `functools.partial` / a lambda function.
- **pre\_retry\_hook** – Can be set to any callable that will be called before *function* is called. No arguments are passed to this function. If your function requires arguments, consider defining a separate function or use `functools.partial` / a lambda function. If *wait* is set, *pre\_retry\_hook* will be called before the waittime. Exceptions that are raised when calling this hook are not caught.

**Returns** The result of calling the given *func*.

**Raises****Any exception which is**

- not in the given *exceptions*
- raised in *pre\_retry\_hook* or in *cleanup\_hook*
- raised in the last attempt at calling *func*



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`





## C

`call()` (in module `tryagain`), [1](#)