

# Live de Python #59

Objetos assíncronos

# Live de Python #59

`collections.abc` - Parte 2

Continuação espiritual da Live 32

# Roteiro

- Corrotinas
  - Awaitable objects
  - asynchronous iterator
  - asynchronous context manager
  - Coroutines objects (**se der tempo**)
- Links úteis para complementar, vai faltar conteúdo, **não diga que não avisei:**
  - <https://docs.python.org/3/reference/datamodel.html#coroutines>
  - <https://www.python.org/dev/peps/pep-0492/>
  - <https://docs.python.org/3/library/collections.abc.html#>

# Awaitable

Awaitable ou “aguardável” são objetos que implementam o método `__await__`. Funções declaradas com `async def` são exemplos de awaitables.

```
class AsyncRandom(Awaitable):
    def __await__(self):
        return self.randint(1, 10).__await__()

    async def randint(self, start, stop):
        value = randint(start, stop)
        return await sleep(0.2, result=value)
```

# Awaitable (OBS\*)

Objetos de corrotinas podem ser definidas sem ser awaitable, em **CPython**



```
In [1]: from types import coroutine
```

```
In [2]: @coroutine
...: def batatinhas():
...:     yield from 'batatinhas'
...:
```

```
In [3]: from collections.abc import Awaitable
```

```
In [4]: isinstance(batatinhas, Awaitable)
Out[4]: False
```

# Awaitable (OBS\*\*)

Objetos de corrotinas podem ser definidas sem ser awaitable, em **CPython**



```
In [1]: from asyncio import coroutine
```

```
In [2]: @coroutine
...: def batatinhas():
...:     yield from 'batatinhas'
...:
```

```
In [3]: from collections.abc import Awaitable
```

```
In [4]: isinstance(batatinhas, Awaitable)
Out[4]: False
```

# Awaitable (OBS\*\*\*)

```
In [1]: from asyncio import coroutine as a_coro
```

```
In [2]: from types import coroutine as t_coro
```

```
In [3]: @t_coro
...: def t_batatinhas():
...:     yield 'batatinhas'
...:
```

```
In [4]: @a_coro
...: def a_batatinhas():
...:     yield 'batatinhas'
...:
```

```
In [5]: from inspect import isawaitable
```

```
In [6]: isawaitable(a_batatinhas()) and \
...: isawaitable(t_batatinhas())
Out[6]: True
```

Corrotinas são aguardáveis mesmo sem `__await__`

# Asynchronous Iterator