

Live de Python #58

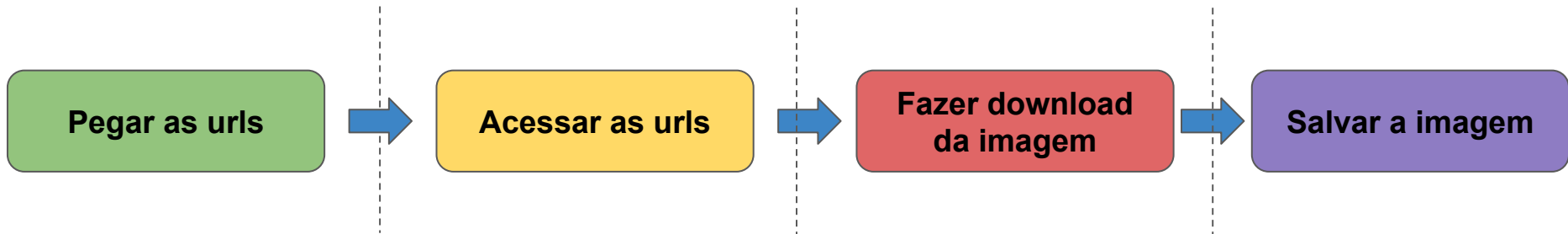
requisições web com Asyncio

Roteiro

- Revisão do problema
- Entender event loops
- Requests com asyncio

Relembrando o problema

Fazer o download de um sprite dos primeiros 100 pokémons da pokeapi



Glossário (vocabulário mais básico possível)

Se você entende do assunto, pode achar essa parte uma aberração, **CALMA**

Corrotinas: São funções assíncronas (ou seja, tem um decorador ou começam com `async`)

```
import asyncio

async def ping_server(ip):
    pass

@asyncio.coroutine
def ping_server(ip):
    pass
```

Glossário (vocabulário mais básico possível)

Se você entende do assunto, pode achar essa parte uma aberração, **CALMA**

yield from / await: É uma maneira de consumir uma corrotina. Lembre-se, para consumir uma corrotina, você precisa ser uma corrotina

```
@asyncio.coroutine
def parser_do_ping(path):
    yield from ping_server('127.0.0.1')

async def parser_do_ping(path):
    await ping_server('127.0.0.1')
```

Glossário (vocabulário mais básico possível)

Se você entende do assunto, pode achar essa parte uma aberração, **CALMA**

event_loop: loop de execução de tarefas/corrotinas

```
@asyncio.coroutine
def parser_do_ping(path):
    yield from ping_server('127.0.0.1')

async def parser_do_ping(path):
    await ping_server('127.0.0.1')
```