

# ASYNCIO

ASYNCHRONOUS IO SUPPORT REBOOTED

**GUREL ERCEIS**





The proposal includes a pluggable event loop, transport and protocol abstractions similar to those in Twisted, and a higher-level scheduler based on yield from ( PEP 380 ).

The proposed package name is `asyncio` .

**PEP 3156**

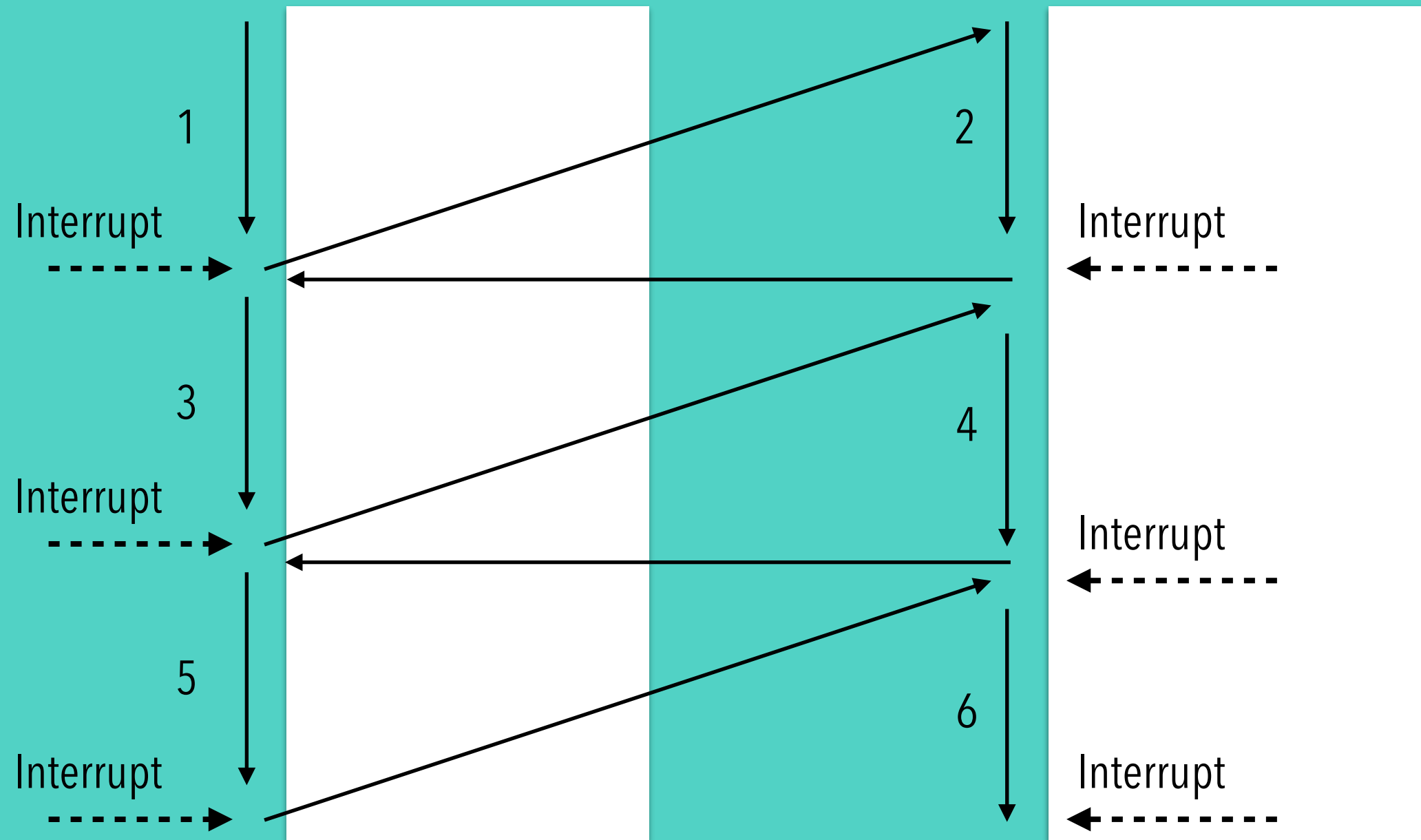
**DEVAM ETMEDEN**

**Concurrency != Paralellism**

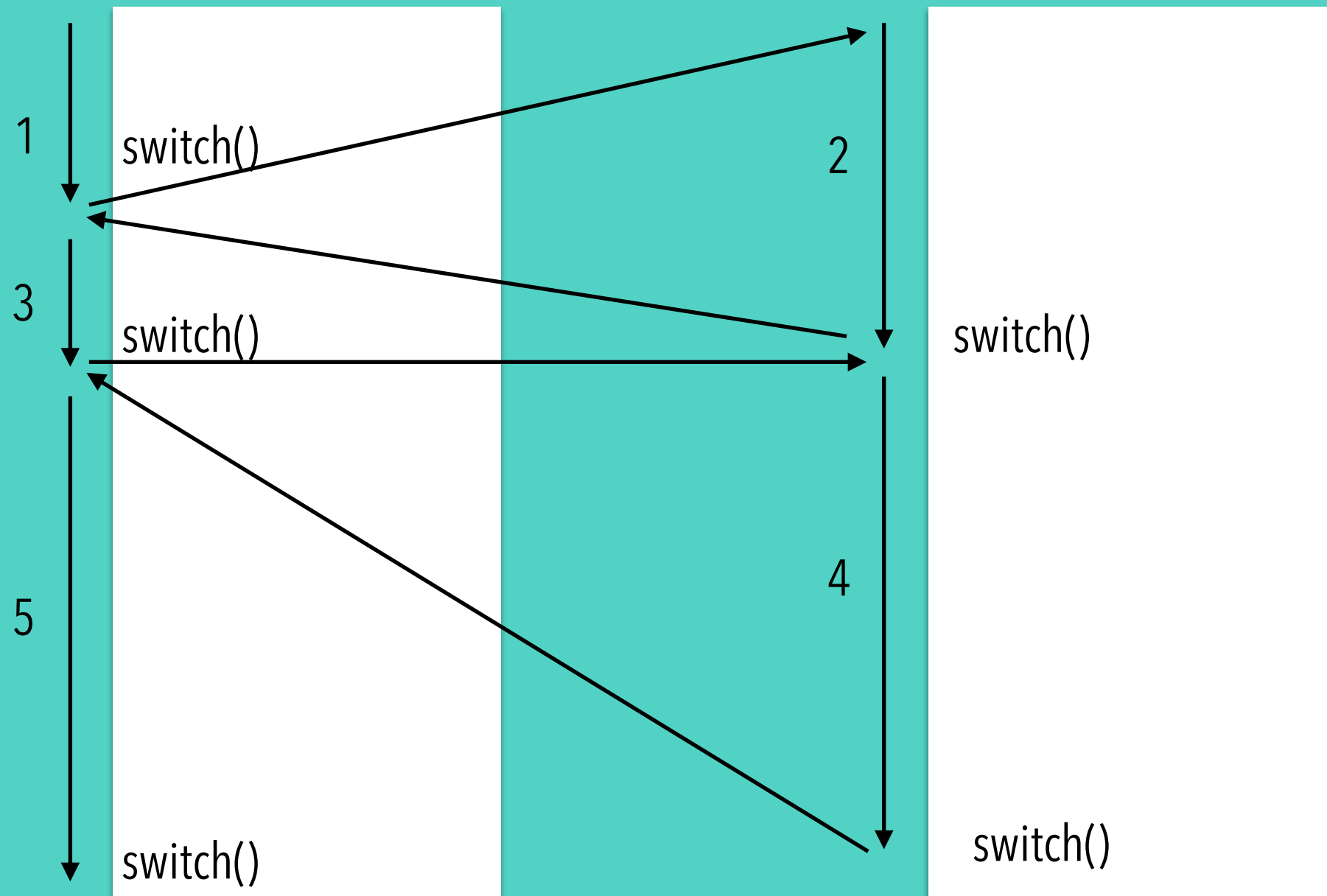
**Tek bir Thread**

**Birden fazla işlemci  
çekirdeğinin kullanımı**

# Pre-emptive multi tasking



# Cooperative Multi Tasking



## ASYNC CALLS

- Non-blocking operation
  - Genelde I/O
  - OS desteği gerekli
- Concurrency Sağlar, bazen de paralellik

**Demo**

# ARCHITECTURE

Event Loop

Coroutine, Future, Task

Transport, Protocol





## EVENT-LOOP

- Generally use default event loop
  - `asyncio.get_event_loop()`
- EventLoop uses selectors
- When event occurs, calls registered callback
- Monitors whether callback is done



## COROUTINE

- coroutine: a function that can be suspended and resumed
  - use 'yield from' ('await' after 3.5)
  - decorator with '@asyncio.coroutine' ('async' after 3.5)
- Future: represents something running in background
- Task: subclass of future used for coroutines
  - Includes synchronization

## TRANSPORT, PROTOCOL

- Transport moves data (like Sockets)
- Protocol represents application (like HTTP)
- Can only support async transports (non-blocking)
- Methods on eventloop to create things
  - sockets, file handles, subprocess, signal

# PEP 255 -SIMPLE GENERATORS

```
def generator_function():  
    i = 2  
    for x in range(10):  
        yield i  
        i = i << 1  
  
for x in generator_function():  
    print(x)
```

# PEP 255 -SIMPLE GENERATORS

```
def generator_function():  
    i = 2  
    for x in range(10):  
        yield i  
        i = i << 1  
  
print(generator_function)  
print(generator_function())
```

# PEP 255 -SIMPLE GENERATORS

```
def generator_function():  
    i = 2  
    for x in range(10):  
        yield i  
        i = i << 1  
  
gen = generator_function()  
for x in range(10):  
    print(next(gen))  
    # print(gen.__next__())
```

# PEP 255 -SIMPLE GENERATORS

```
def generator_function():  
    i = 2  
    for x in range(10):  
        yield i  
        i = i << 1  
  
gen = generator_function()  
while True:  
    try:  
        print(next(gen))  
    except StopIteration:  
        break
```

# PEP 255 -SIMPLE GENERATORS

```
def grep(pattern):  
    while True:  
        line = (yield)  
        if line and pattern in line:  
            print(line)
```

```
g = grep("Gurel")  
g.send(None)  
g.send("Gurel topu at")  
g.send("Ali topu tut")  
g.close()
```

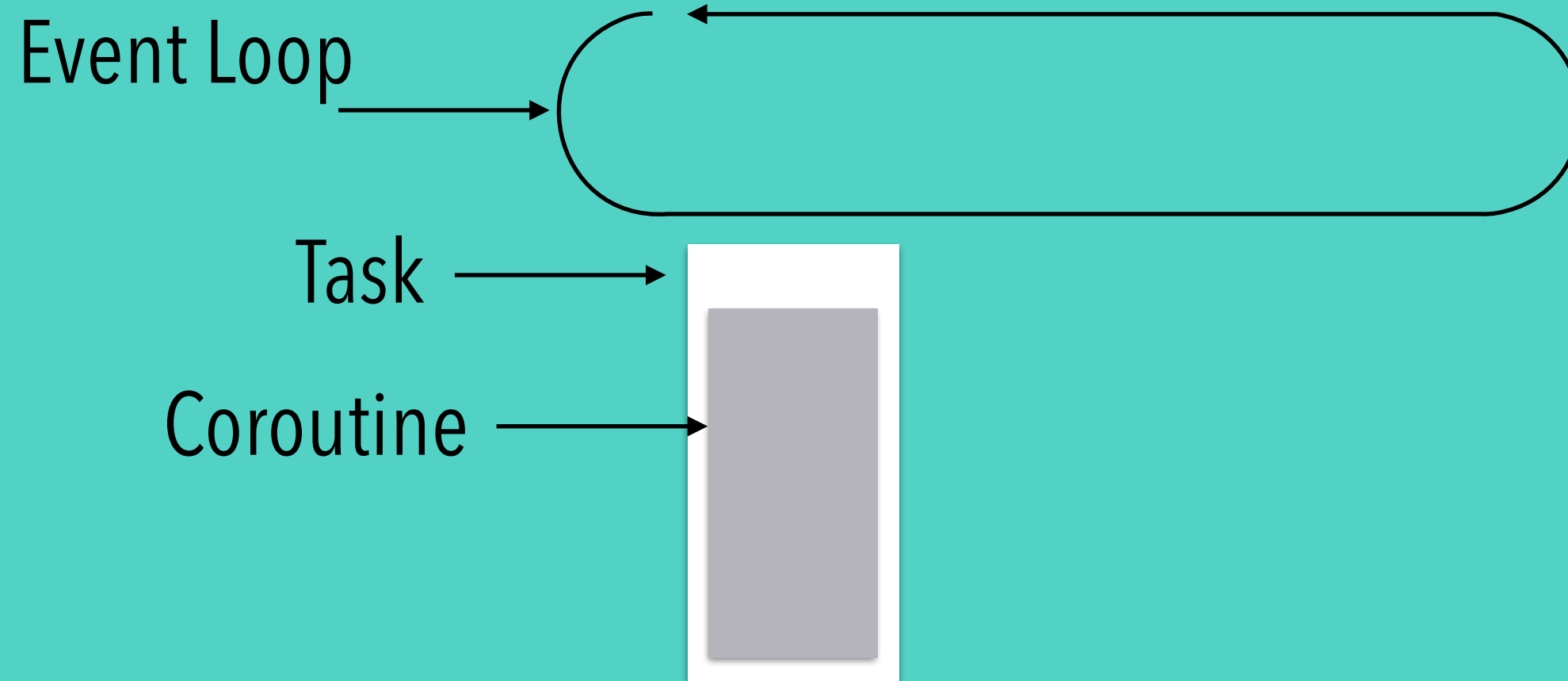


# PEP 255 -SIMPLE GENERATORS

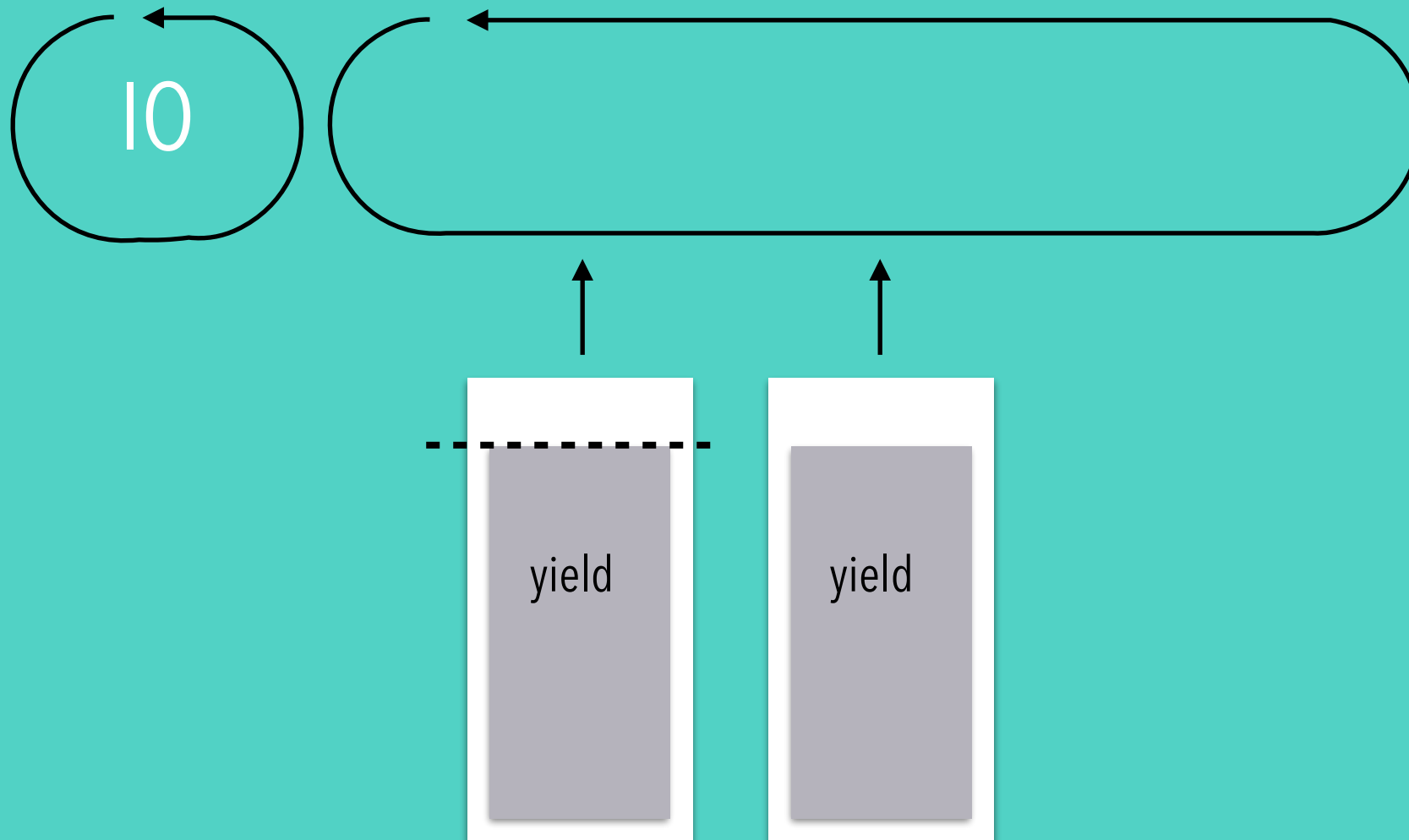
```
def grep(pattern):  
    last_success = None  
    while True:  
        line = (yield last_success)  
        if line:  
            if pattern in line:  
                last_success = True  
            else:  
                last_success = False
```

```
g = grep("Gurel")  
g.send(None)  
g.send("Gurel topu at")  
print(next(g))  
g.send("Ali topu tut")  
print(next(g))  
g.close()
```

# EVENT LOOP

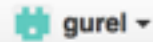
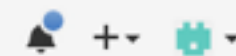


# EVENT LOOP



# – Either `_fut_waiter` is `None`, and `_step()` is scheduled;  
# – or `_fut_waiter` is some `Future`, and `_step()` is *\*not\** scheduled.

```
def _step(self, value=None, exc=None):
    assert not self.done()
    try:
        if exc is not None:
            result = coro.throw(exc)
        elif value is not None:
            result = coro.send(value)
        else:
            result = next(coro)
    except StopIteration as exc:
        self.set_result(exc.value)
    except BaseException as exc:
        self.set_exception(exc)
        raise
    else:
        if isinstance(result, futures.Future):
            result.add_done_callback(self._wakeup)
            self._fut_waiter = result
        elif result is None:
            # Bare yield relinquishes control for one event loop iteration.
            self._loop.call_soon(self._step)
    finally:
        self.__class__.__current_tasks.pop(self._loop)
        self = None # Needed to break cycles when an exception occurs.
```



gurel ▾



6 days ago

**davelab6** commented on issue [zeynepakay/Rakkas#9](#)

Great :) On 4 April 2016 at 07:19, zeynepakay [notifications@github.com](#) wrote: Still working on it. Should have something this week. :) On 30 Mar ...



6 days ago

**zeynepakay** commented on issue [zeynepakay/Rakkas#9](#)

Still working on it. Should have something this week. :) On 30 Mar 2016, at 16:18, Dave Crossland [notifications@github.com](#) wrote: @zeynepakay [http...](#)



11 days ago

**davelab6** commented on issue [zeynepakay/Rakkas#9](#)

@zeynepakay any news on this? :)

**burakcan** starred [gurel/marvin](#) 17 days ago

18 days ago

**mertsalik** pushed to [develop](#) at [gurel/Kutumubu](#)

**b57f23c** added missing chroma\_key effect



18 days ago

**mertsalik** pushed to [develop](#) at [gurel/Kutumubu](#)

**ce0a4f2** added missing converting helpers



18 days ago

**mertsalik** pushed to [develop](#) at [gurel/Kutumubu](#)

**820d45f** added missing printing helper

**brtn** starred [gurel/robotjs](#) 19 days ago

23 days ago

**mertsalik** pushed to [develop](#) at [gurel/Kutumubu](#)

**3013a2f** key color parameter



23 days ago

**mertsalik** opened issue [mertsalik/BrandWall#62](#)

Default hashtagwall ön sayfasının tekrar yazılması

**GPG signature verification** ✕

You can now verify GPG signed commits and tags!

[View 17 new broadcasts](#)**Repositories you contribute to**

<a href="#">zeynepakay/Rakkas</a>	1 ★
<a href="#">fatiherikli/fil</a>	486 ★
<a href="#">jviotti/electron-json-storage</a>	75 ★
<a href="#">octalmage/robotjs</a>	4,719 ★

**Your repositories** 14[+ New repository](#)

Find a repository...

[All](#) [Public](#) [Private](#) [Sources](#) [Forks](#) [marvin](#) [Kutumubu](#) [robotjs](#) [mertsalik/BrandWall](#) [zeynepakay/Rakkas](#) [geolocation\\_search](#) [BrandWall](#) [react-dnd](#) [mertsalik/Kutumubu](#) [fil](#) [panel.kutumubu.com](#) [mertsalik/panel.kutumubu.com](#) [is.js](#) [gurel.github.io](#) [Subscribe to your news feed](#)

# THANK YOU

ASYNCIO 101

