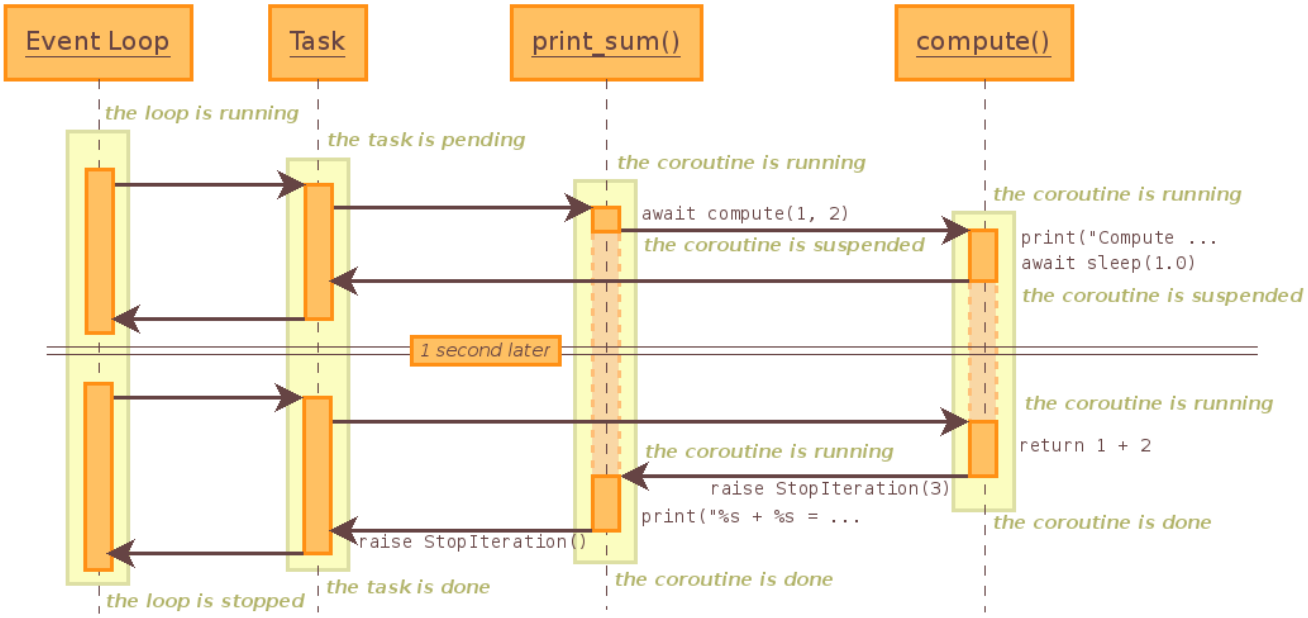
python with actor

-

- ,
-
-



3.5 Async(coroutines) ,C# Async Task

. .

.

http://thespianpy.com/doc/in_depth#outline-container-hH-1b595f63-447b-40d1-ad52-582e4ac7a536

```
import asyncio
from thespian.actors import *
import time


class EngActor(Actor):
    def __init__(self):
        self.greeting='hi there'

    def receiveMessage(self, message, sender):
        print(self.greeting)
        self.send(sender, self.greeting )

class KorActor(Actor):
    def __init__(self):
        self.greeting=''

    def receiveMessage(self, message, sender):
        time.sleep(1);
        print(self.greeting)
        self.send(sender, self.greeting )

class HelloAsync:
    def __init__(self):
        self.greeting = "hi there2"

    async def SayHello(self,message):
        await asyncio.sleep(1)
        print(self.greeting)

#
def say_hello():
    actorSys = ActorSystem()
    engActor = actorSys.createActor( EngActor )
    korActor = actorSys.createActor( KorActor )
    actorSys.tell(korActor, '~')
    actorSys.tell(engActor, 'are you there?')

say_hello();


#
async def asay_hello():
    asyncHello = HelloAsync()
    await asyncHello.SayHello('are you there?')

loop = asyncio.get_event_loop()
loop.run_until_complete(asay_hello())
loop.close()


print('============= wait for test ==================')
time.sleep(3);

## Result -   ~
#
# hi there
```

pykka

```
import pykka
import time


class Greeter(pykka.ThreadingActor):
    def __init__(self, greeting='Hi there!'):
        super(Greeter, self).__init__()
        self.greeting = greeting
    def on_receive(self, message):
        if self.greeting=='Hi you!':   #   .
            time.sleep(0.5);

        print(self.greeting)
        return "OK"


#  ,   .
actor_ref = Greeter.start(greeting='Hi you!')
actor_kor_ref = Greeter.start(greeting='')


#   .  -
answer = actor_ref.ask({'msg': 'Hi?'}, timeout=3)
answer = actor_kor_ref.ask({'msg': '?'}, timeout=3)


#  . - (   )
actor_ref.tell({'msg': 'Hi?'} )
actor_kor_ref.tell({'msg': '?'} )

print('============= wait for test ==================')
time.sleep(1.5);

## Result
#
#Hi you!
```

.

.

C#

```
public class MyActor : ReceiveActor
{

    public MyActor()
    {
        Receive<string>(message => {
            Sender.Tell("RE:" + message);
        });

        Receive<SomeMessage>(message => {
            Sender.Tell("RE:" + message.message);
        });
    }
}
```

Scala/Java

```
class Greeter extends Actor {

  var greeting = "default"

  def receive = {

    case WhoToGreet(who) => greeting = s"hello, $who"

    case Greet           => sender ! Greeting(greeting) // Send the current greeting back to the sender
  }
}
```

( ),

- https://docs.python.org/3/library/asyncio-task.html
- https://godaddy.github.io/Thespian/doc/blog/async_blog_2016Feb/async_2016Feb.html (async vs actor)