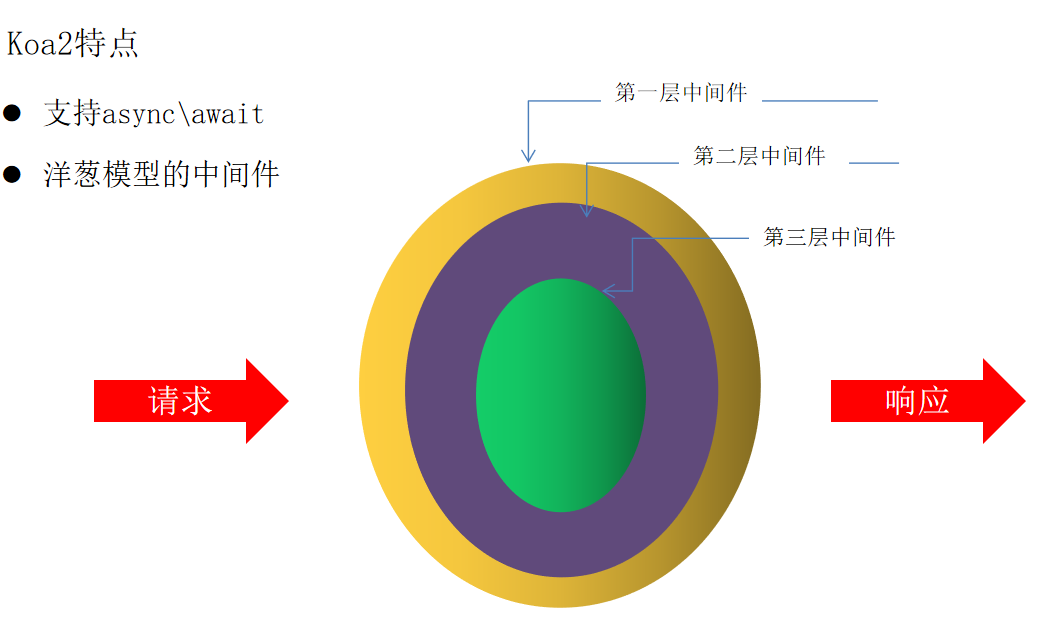
## 洋葱模型



第一层中间往下执行，执行到底层中间件，然后再往上执行

中间件回调函数中需要加next(),否则不会传递到下层中间件

const koa = require("koa");

const app = new koa();

app.use((ctx, next) => {

  ctx.response.body = "hello";

  console.log("第一层中间件--1");

  next();

  console.log("第一层中间件--2");

});

app.use((ctx, next) => {

  console.log("第二层中间件--1");

  next();

  console.log("第二层中间件--2");

});

app.use((ctx, next) => {

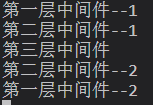
  console.log("第三层中间件");

});

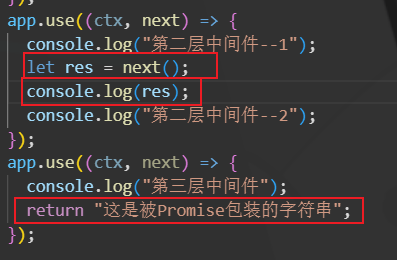
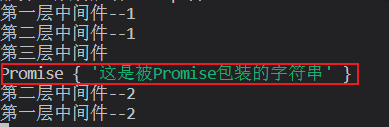
app.listen(3333, () => {

  console.log("服务器成功启动，http://127.0.0.1:3333");

});



## next()返回的是一个Promise对象

所以可以通过async/await接收结果

