

day91 flask

内容回顾

1. django和flask的区别？
2. django和flask哪个好？

都好

3. django的app和flask的蓝图的区别？
4. 装饰器的应用

今日概要

- 面试题
- 数据库连接池
- flask配置
- flask路由
- before_request/after_request

今日详细

1. 面试题

1. 编程语言及他们之间的区别？

C/C++ ,很多语言的底层实现都是与C, 代码执行效率高, 自己做内存管理, 对代码要求比较高, 很多功能需要手动试下。

Java, 比较好的编程语言, 很多企业级应用都会选择java。

C#, 是微软开始的编程语言, 部署时需要放在window server 上, 最大弊端是window系统花钱。

PHP, 一般用于快速搭建网站。

Python, 简洁且入门简单, 很少的代码就可以做很多事情。

Go lang, 语法和C比较接近, 处理并发时比较有优势 + docker。

2. 构造函数和析构函数

构造函数，用于创建对象的函数。
析构函数，用于销毁对象的函数。

```
class Foo(object):
    def __new__(self,*args,**kwargs): # 构造函数
        pass

    def __del__(self,*args,**kwargs): # 析构函数
        pass

obj = Foo()
del obj
```

3. 重写和重载的区别?

```
class Foo(object):

    def f1(self,int a1):
        return 123

    def f1(self,string a1):
        pass
```

重载，函数名相同而参数类型/个数/返回值不同。

```
class Foo(object):
    def f1(self):
        print(123)
```

```
class Bar(Foo):
    def f1(self):
        print(666)
```

重写，在子类中对父类中的方法进行重写。

4. 什么是接口?

```
Interface IPerson:

    def f1(self,a1)

    def f2(self,a1)

public interface Predator {
    boolean chasePrey(Prey p);
    void eatPrey(Prey p);
}
```

接口是以interface关键字开头，内部可以定义方法，但方法中不用写具体实现，他的作用：专门用于约束实现他的类，Python中没有接口类型。

```
class UserInfo(IPerson): # UserInfo类实现了IPerson接口

    def f1(self,a1):
        return 123

    def f2(self,a1):
```

```
print(123123)
return 666
```

5. Python的类执行多继承/ 其他语言不支持多继承

```
# Python
class Foo(Base,NewBase):
    pass

# 其他
class Base:
    pass

class NewBase(Base):
    pass

class Foo(NewBase):
    pass
```

6. 在其他语言中 可以实现多个接口 / 不可以继承多个类

```
# Java
interface IBase:
    def f1(sef)

interface INewBase:
    def f2(sef)

class Foo(IBase,INewBase):
    def f1(self):
        pass
    def f2(self):
        pass
```

7. 抽象类和抽象方法

```
#java
abstrac class Base(object): # 抽象类

    def f1(self): # 普通方法
        print(123)

    def abstract f2(self): # 抽象方法，内部不能写任何代码
        pass

class Foo(Base):

    def f2(sef):
        print(123)

obj = Foo()
obj.f2()
obj.f1()
```

8. 总结

- 重载
- 重写
- 构造函数

Python

```
class Foo:
    def __new__(self):
        pass
```

```
public class Foo{
```

```
    # 函数和类名相同
```

```
    public void Foo(){
```

```
    }
```

```
}
```

- 析构函数

```
__del__
```

```
finalize
```

- 接口
- 抽象类/抽象方法

9. 三元运算

10. lambda 表达式

11. yield关键字

12. 进程和线程的区别?

13. 数据库

数据库设计：博客系统 / 呼啦圈

SQL语句：

```
select * from user left join depart on user.depart_id = depart.id
```

```
select * from user left join depart on user.depart_id = depart.id order
by id desc
```

inner join 和 left join的区别？

```
select gender,count(1) from user group by gender
```

```
select gender,count(1) from user group by gender having count(1) > 2
```

视图：是一个虚拟的表。

为某个sql语句取名字： `select * from user where id<100` =====> `t1`

`select * from t1 where name = '成说'、`

存储过程：是大堆存储在数据库中的SQL语句。

```
create producer pro(x1):
```

```
    update ...
```

```
    delete ...
```

```
    select ...
```

触发器：存储在数据中的一个特殊的東西。

```
create trigger x1 before insert 表:
....
create trigger x1 after insert 表:
```

14. 在浏览器上输入 www.xxx.com 发生了什么?

- dns解析: www.xxx.com 解析成 IP
- 本质通过socket进行通信
- 请求到达框架, 以django框架为例:

15. http和https的区别?

http, 默认端口: 80
https, 默认端口: 443

http的数据是基于明文传输。
https的数据是基于密文传输。

<https://www.cnblogs.com/wupeiqi/articles/11647089.html>

16. cookie和session的区别?

17. get和post的区别?

18. 数据结构

- o 链表
 - 单项
 - 双向
- o 树
 - 二叉树前序遍历
 - 二叉树中序遍历
 - 二叉树后序遍历
- o 面试题
 - 伪代码实现一个链表
 - 对链表进行反转 (逆置)
 - 两个队列生成一个栈
 - 前序遍历 / 中序遍历 / 后序遍历

<https://www.cnblogs.com/wupeiqi/p/11604077.html>

19. 算法

- o 冒泡排序
- o 快速排序
- o 二分查找

