

语言本身: 变量、赋值、类、异常处理 **数据类型 流程控制 类 可执行对象**

特殊用法: 第三方库(C++)、推导式、装饰器、魔术方法(将实际工作处理内容封装成最普通的数据类型实现)

底层技能: 多进程、多线程、协程(aiohttp) **官方文档-标准库-并发执行**

底层理论: http协议、tcp协议

外延技能: 队列、redis缓冲、数据库(CMDB) mysql + flask + 搜索

行业技能: 运维CMDB、数据采集-清洗-存储-检索内容的处理-语义分析-展示

<b>数据类型</b>	<b>None</b>		<b>映射</b> dict	
	<b>数值</b>	int float complex	<b>集合</b> set	
	<b>序列</b> (可切片)	string list tuple range	<b>可调用类型</b>	function builtin function builtin method class - call - function
		buffer		instance module
	<b>类</b>	class metaclass		

结合应用场景: 重点、难点

装饰器模式如何把url和卡套在一起 - 用到 add-url-rule()

Mixin和其它类配合加强类

基本类型 数据类型、内置函数 - 使用魔术方法把问题转化为基本数据类型和内置函数上去

用魔术方法把数问题转化为基类型, 用内置函数处理

— getattr — 读取配置文件的值      — setattr — 设置配置文件的值