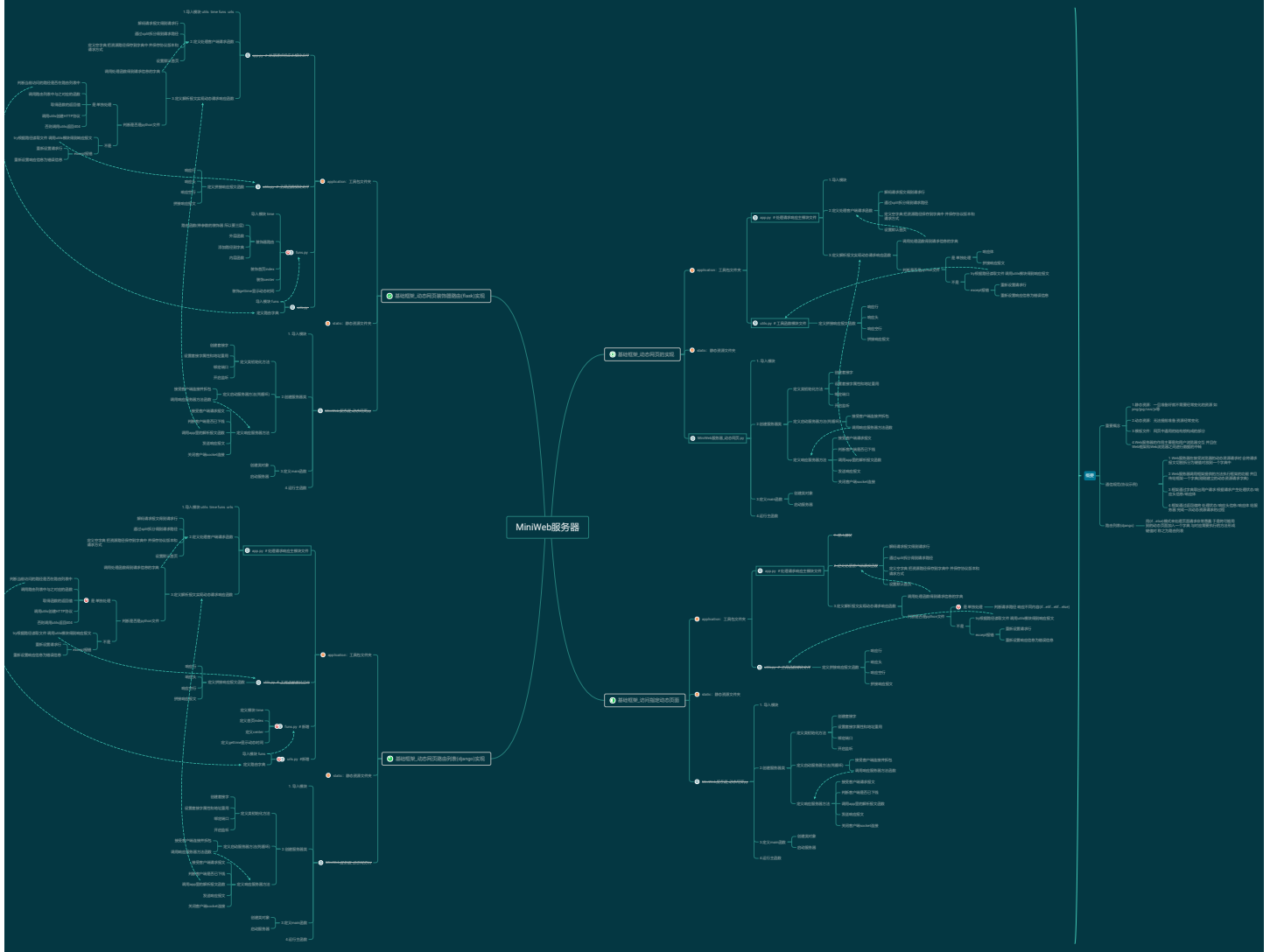
MiniWeb服务器
基础框架_动态网页装饰器路由(flask)实现
application：工具包文件夹
static：静态资源文件夹
基础框架_动态网页路由列表(django)实现
基础框架_动态网页的实现
基础框架_动态网页的实现
概念
1.静态资源：一切准备好能不需要任何变化的资源 如 jhq.jpg/index.html
2.动态资源：无法提前准备 资源任何变化
3.模板文件：网页中通用的结构性的部分
4.Web服务器的功能主要是处理产生的资源文件 并且在Web应用中调用正确的工具进行渲染
通信规范(协议示例)
1.Web服务器在等待浏览器的连接请求时 会向请求报文加固定的三行换行符 作为一个请求
2.Web服务器解析报文时 会先读取报文的第一行 并以换行符为一个字典(键值对)进行拆分成 请求字典)
3.根据请求字典取出客户端请求 将请求路径产生处理方法 响应内容
4.将响应内容进行组装 包括状态 响应头 信息/响应体 将报文转换 成一个完成的 响应请求报文
简单的服务器http：
用户、Web服务器和应用服务器等角色 于 各种功能等 封装动态内容部分 一个文件 如何处理解析的方法和规 解析的 和 之间   的
1.导入模块 urls time func urls
解析请求报文/响应请求行
通过urlrf的分/响应请求路径
2.定义/响应产/响请求函数
定义文字典 把浏览器传过来的字典中 并保存协议版本和 请求方式
设置默认值元
调用处理函数响应请求的字典
3.定义解析/定义实现动态请求响应函数
判断是否是urlrf文件
是 单独处理
判断请求路径 响应不同内容的if、elif、elif、else)
响应内容urlrf文件
调用urlrf模块响应报文文件
调用urlrf模块响应报文
不是
mime类型
图标/设置请求行
图标/设置响应合作方响应信息
1.导入模块 urlrf time func urlrf
解析请求报文/响应请求行
通过urlrf的分/响应请求路径
2.定义/响应产/响请求函数
定义文字典 把浏览器传过来的字典中 并保存协议版本和 请求方式
设置默认值元
调用处理函数响应请求的字典
3.定义解析/定义实现动态请求响应函数
判断是否是urlrf文件
是 单独处理
判断请求路径 响应不同内容的if、elif、elif、else)
响应内容urlrf文件
调用urlrf模块响应报文文件
调用urlrf模块响应报文
不是
mime类型
图标/设置请求行
图标/设置响应合作方响应信息
urlrf.py 不是请求响应主要文件
定义判断响应报文函数
响应行
响应头
响应空行
判断响应报文
application：工具包文件夹
static：静态资源文件夹
1.导入模块
创建数据字
设置数据字属性响应重用
响应接口
开启监听
使用客户端连接开始
调用响应服务函数方法
使用客户端请求报文
判断客户端是否已下线
调用Web服务解析报文函数
发送响应报文
关闭客户端socket连接
2.创建服务器类
定义实例初化方法
定义启动服务器方法用循环
定义响应服务方法
创建实例对象
启动服务器
3.定义main函数
4.运行主函数
1.导入模块
创建数据字
设置数据字属性响应重用
响应接口
开启监听
使用客户端连接开始
调用响应服务函数方法
使用客户端请求报文
判断客户端是否已下线
调用Web服务解析报文函数
发送响应报文
关闭客户端socket连接
2.创建服务器类
定义实例初化方法
定义启动服务器方法用循环
定义响应服务方法
创建实例对象
启动服务器
3.定义main函数
4.运行主函数
urlrf.py 不是请求响应主要文件
定义判断响应报文函数
响应行
响应头
响应空行
判断响应报文
application：工具包文件夹
static：静态资源文件夹
1.导入模块
创建数据字
设置数据字属性响应重用
响应接口
开启监听
使用客户端连接开始
调用响应服务函数方法
使用客户端请求报文
判断客户端是否已下线
调用Web服务解析报文函数
发送响应报文
关闭客户端socket连接
2.创建服务器类
定义实例初化方法
定义启动服务器方法用循环
定义响应服务方法
创建实例对象
启动服务器
3.定义main函数
4.运行主函数