

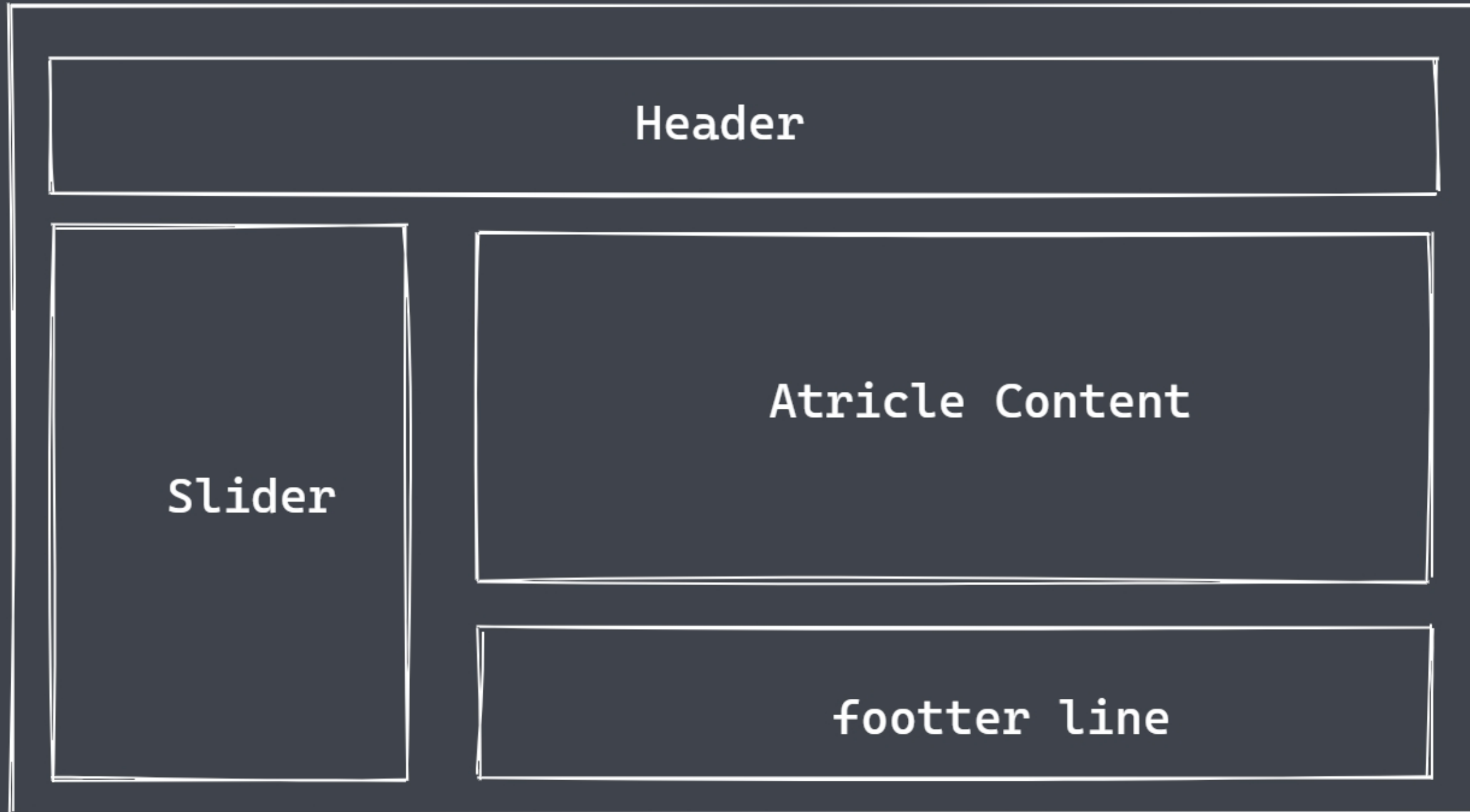
# React 介绍

# React 是什么？

- React 是 facebook 推出的用于构建用户界面的前端 javascript 框架
- <https://reactjs.org/>
- React 使用组件构建用户界面

# 组件是什么？

Components?



一块区域，包含了 html css 及  
javascript

# 组件开发的优势

- 将一个庞大复杂的应用程序拆分成多个独立单元
- 组件之间互相独立，有利于应用程序的维护
- 组件可以重用，一次编写多地复用

# React 开发环境搭建

# JSX 语法

# JSX 事件操作



# 循环数据

# 设置内联样式

# 设置外联样式

# 创建组件

# 向组件传递数据

# 设置 props 默认值

# 验证 props 数据类型

# 向组件传递 JSX



# 组件布局实例

Header 内容

首页内容 | 列表页内容

Footer 内容

# 组件状态

# setState使用

# 组件中的 this

# 单向数据流动

# 受控表单值绑定与更新

# 受控表单之下拉框



# 受控表单之单选框

# 受控表单之复选框

# 非受控表单

# 管理组件之系统拆分

# 添加学员信息

# 学生信息展示

# 学生信息删除

# 计算平均值等



# React 生命周期

# React 组件也有自己的生命周期

# 组件的生命周期大致分为三个部分

# 组件挂载阶段

# 组件更新

# 组件卸载

# React 发送 Ajax请求

# React 请求转发



# Mock 数据

# Redux 工作流程

# 创建 store 及 reducer

# 获取 store 数据

# 组件修改 store 数据

# Action 传递参数

# 提取action代码为函数

# 自动生成action触发函数



# 设置action类型常量

# Reducer 拆分与合并

# redux 工作流程梳理

# redux 使用代码优化

# redux 中间件工作流程

# redux-thunk异步解决方案

# redux-saga异步解决方案

# redux-saga拆分与合并



# 简化action与reducer

# todos组件拆分

# 任务列表展示

# 新增任务

# 删除任务

# 任务状态更新

# 计算未完成任务数据

# 任务筛选



# 删除已完成任务

# 修改任务名称

# 实例代码优化

# immutable 使用

# 路由基本使用

# exact 属性

# 路由匹配规则

# 动态路由传参



# 查询参数传参

# 路由嵌套

# 路由跳转及重定向

# 路由守卫

# 路由守卫组件

# 路由模块懒加载

# 非路由组件传递路由信息

# useState 设置状态



# useState 注意点

# useEffect 使用

# useEffect 优点

# useContext 使用

# useReducer 使用

# useCalllback 优化代码

# useMemo 使用

# useRef 使用



# useImperativeHandle 使用

# useLayoutEffect 使用

# 自定义 Hook 实现



扫码联系老师

技能评估、福利资料、课程优惠

Made with ❤️ by LagouFed