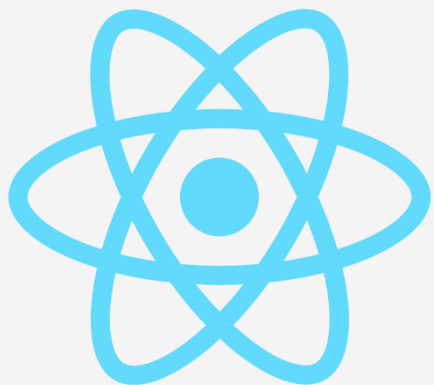


前端框架 是用来做什么的

@被删

热门前端框架



React



Vue



Angular

1

为什么要用前端框架

编写一个页面

1. 编写静态页面
2. 添加事件监听
3. 绑定事件
4. 更新页面内容

```
<form>
var nameValueEl = document.getElementById("name-value");
var emailValueEl = document.getElementById("email-value");
// 定义 updateValue 函数，用来更新页面内容
function updateNameValue(e) {
    nameValueEl.innerText = e.srcElement.value;
}
function updateEmailValue(e) {
    emailValueEl.innerText = e.srcElement.value;
}
```

如果是比较复杂的页面呢？

问答卡片

1. 新增一个卡片
2. 对输入框进行字数限制
3. 获取输入框内容

```
var index = 0;
// 用来新增一个卡片，卡片内需要填写一些内容
function addCard() {
    // 获取一个id为the-dom的元素
    var body
    // 从该元
    var addDom
    // theDom使用上面代码保存下来的引用哈
    // 问题绑定值
    theDom
    addDom.body.on("keyup", ".question", function(ev) {
        // 获取卡片的输入值
        // theDom使用上面代码保存下来的引用哈
    }
    function getCardValue(index) {
        var body = $("#the-dom");
        var theDom = body.find('[data-index="' + index + '"]');
        var questionName = theDom.find(".question").val();
        var optionA = theDom.find(".option-a").val();
        var optionB = theDom.find(".option-b").val();
        return { questionName, optionA, optionB };
    }
}
```

如果用前端框架呢？

```
<template>
  <div v-for="card in cards">
    <input
      type="text"
      class="form-control question"
      v-model="card.questionName"
      placeholder="你的问题"
    />
    <input
      type="text"
      class="form-control option-a"
      v-model="card.optionA"
      placeholder="回答1"
    />
    <input
      type="text"
      class="form-control option-b"
      v-model="card.optionB"
      placeholder="回答2"
    />
  </div>
</template>
```

```
<script>
  export default {
    name: "Cards",
    data() {
      return {
        cards: []
      };
    },
    methods: {
      // 添加一个卡片
      addCard() {
        this.cards.push({
          questionName: "",
          optionA: "",
          optionB: ""
        });
      },
      // 获取卡片的输入值
      getCardValue(index) {
        return this.cards[index];
      }
    }
  };
</script>
```


2

前端框架是如何做到的

模板引擎/渲染引擎

开发代码

```
<div>{{ data }}</div>
```

```
new Vue({  
  data: {  
    data: "测试文本"  
  }  
});
```

解析

AST对象

```
thisDiv = {  
  dom: {  
    type: "dom",  
    ele: "div",  
    nodeIndex: 0,  
    children: [{ type: "text", value: "" }]  
  },  
  binding: [{ type: "dom", nodeIndex: 0, valueName: "data" }]  
};
```

编译

更新页面

```
<div>测试文本</div>
```

编译后的结果
是怎么更新到页面中的？

更新页面

1. 绑定映射表方式
2. 直接替换内容方式

```
var a = document.getElementById("a");
```

```
// 1. 更改元素里面内容  
a.innerHTML = "<p>测试</p>";  
// 2. 插入一个<a>元素  
a.innerHTML = "<p>测试</p><a></a>";  
// 3. 删除第一个元素，在这里是前面的<p>测试</p>  
a.innerHTML = "<a></a>";
```

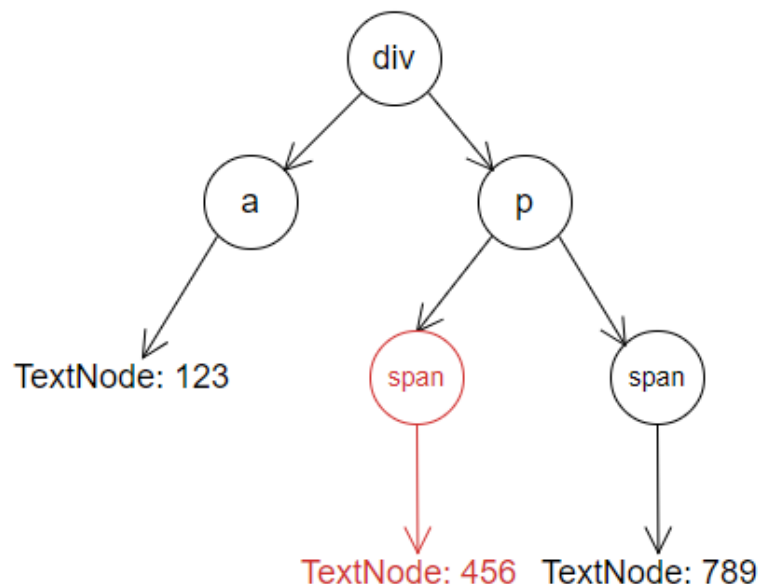
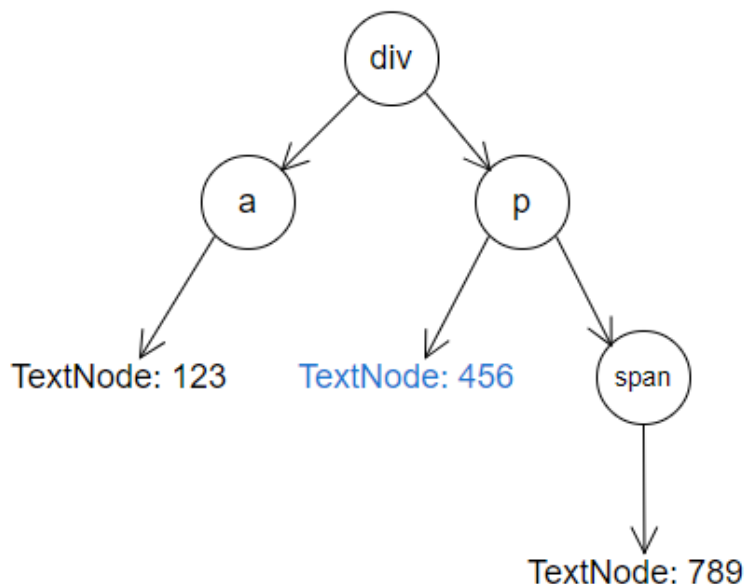
3

前端框架还做了什么事情

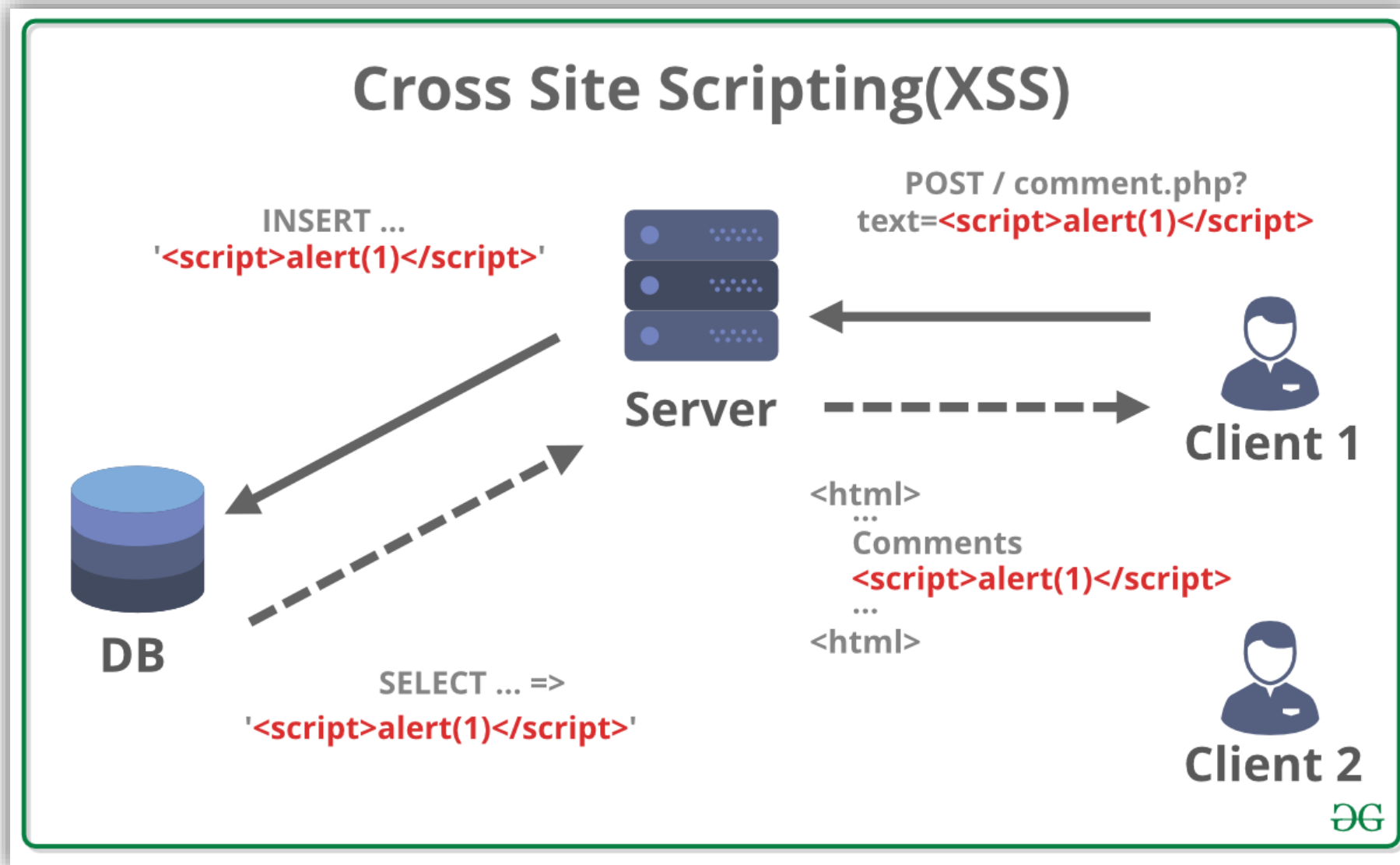
虚拟DOM

1. 用 JS 对象模拟 DOM 树，得到一棵虚拟 DOM 树。
2. 当页面数据变更时，生成新的虚拟 DOM 树，比较新旧两棵虚拟 DOM 树的差异。

3.



避免 XSS 漏洞





谢谢



Github: godbasin
@被删