

极客大学前端训练营第10期

第1周：浏览器工作原理（一）

Zola Wang

第10期领教

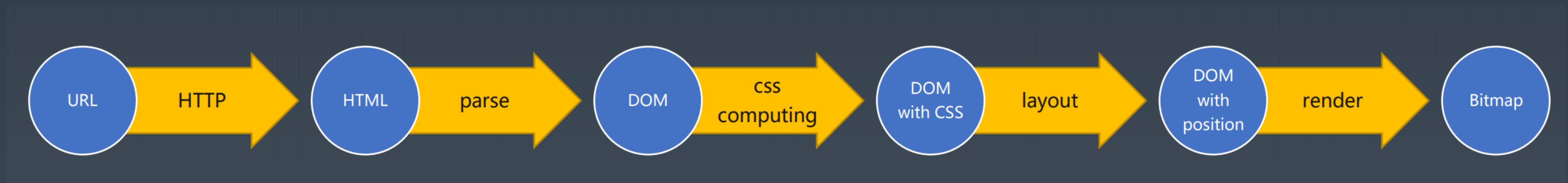
目录

1. 知识点回顾
2. 代码和习题讲解
3. 答疑

知识点回顾

1. 浏览器的渲染流水线

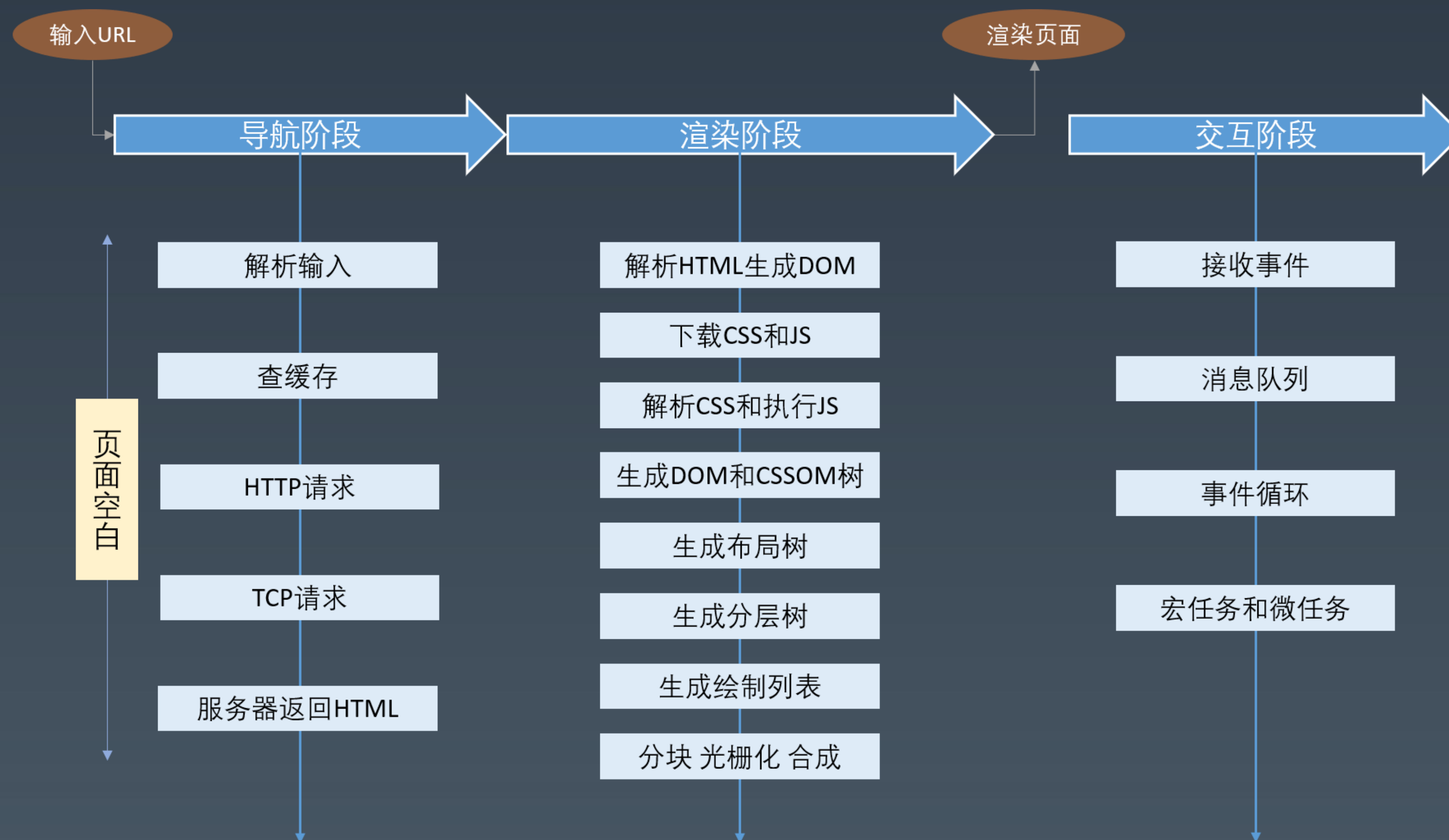
- 过程



- 经典面试题：
 - 从输入URL到最终渲染，浏览器都做了什么

知识点回顾

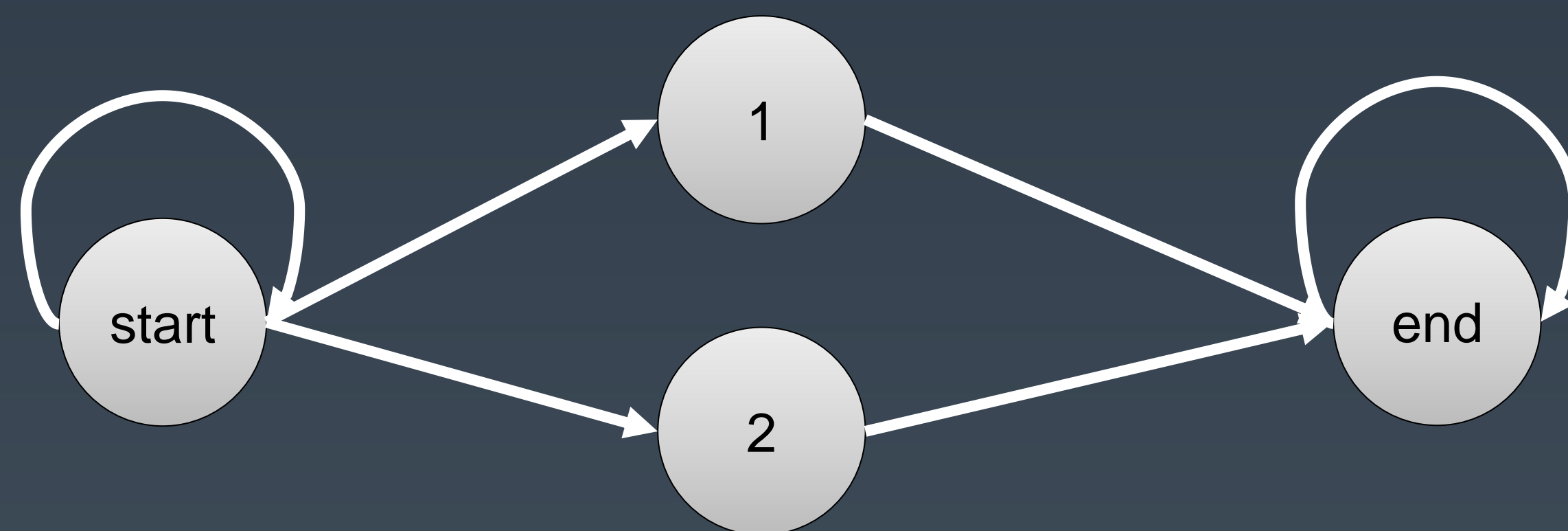
1. 浏览器的渲染流水线



知识点回顾

2. 状态机

- 过程



知识点回顾

2. 状态机

- 代码

//每个函数是一个状态

function **state**(input) //函数参数就是输入

{

//在函数中，可以自由地编写代码，处理每个状态的逻辑

return next;//返回值作为下一个状态

}

//////////以下是调用//////////

while(input) {

//获取输入

state = state(input); //把状态机的返回值作为下一个状态

}

知识点回顾

3. 网络

- 网络模型：7层
- 网络协议：HTTP / TCP / IP

知识点回顾

4. HTTP协议

- Request
 - Line
 - Method
 - Path
 - Protocol version
 - Header
 - Body
- Response
 - Status line
 - Protocol version
 - Status code
 - Status text
 - Header
 - Body

代码和习题讲解

1. 代码

- HTTP Server
- Chunk data

```
// 1. 通过require("http")来引入http对象
const http = require("http");

// 2. 利用http自带方法createServer启动一个后台服务
http.createServer((req, res) => {
  let body = [];
  req.on("error", err => {
    console.log(err);
  }).on("data", chunk => {

    console.log("chunk:\n", chunk);
    // body.push(chunk.toString());
    body.push(chunk);
  }).on("end", () => {

    console.log("end");
    // Buffer类作为是处理二进制数据的缓冲区
    body = Buffer.concat(body).toString();
    console.log("body:", body);
    res.writeHead(200, {"Content-Type": "text/html"});
    res.end(" Hello world\n");
  })
})
```

代码和习题讲解

1. 代码

- HTTP Request Header
- Content-Type 和 Content-Length

```
if(!this.headers["Content-Type"]) {  
    this.headers["Content-Type"] = "application/x-www-form-urlencoded";  
}  
  
if(this.headers["Content-Type"] === "application/json")  
    this.bodyText = JSON.stringify(this.body);  
else if(this.headers["Content-Type"] === "application/x-www-form-urlencoded")  
    this.bodyText = Object.keys(this.body).map(key => `${key}=${encodeURIComponent}`);  
  
this.headers["Content-Length"] = this.bodyText.length;
```


代码和习题讲解

1. 代码

- HTTP Request Send
- toString

```
    toString(){  
        return `${this.method} ${this.path} HTTP/1.1\r  
${Object.keys(this.headers).map(key => `${key}: ${this.headers[key]}`).join('\r\n')}\r  
\r  
${this.bodyText}`  
    }
```

代码和习题讲解

1. 代码

- HTTP Response Body Parser
- Body length

```
if(this.current === this.WAITING_LENGTH) {  
    if(char === '\r') {  
        if(this.length === 0) {  
            this.isFinished = true;  
        }  
        this.current = this.WAITING_LENGTH_LINE_END;  
    } else {  
        this.length *= 16;  
        this.length += parseInt(char, 16);  
    }  
}
```

代码和习题讲解

1. 代码

- 建议
 - 不要照着抄 要理解之后自己写
 - 耐心很重要
 - 锻炼自己debug的能力

代码和习题讲解

2. 习题

- 在一个字符串中，找到字符“a”
- 不准使用正则表达式，纯粹用 JavaScript 的逻辑实现：在一个字符串中，找到字符“ab”
- 不准使用正则表达式，纯粹用 JavaScript 的逻辑实现：在一个字符串中，找到字符“abcdef”
- 用状态机实现：字符串“abcabx”的解析

答疑

1. 收集问题

- 问题池: <https://shimo.im/sheets/KdjCdhPvYKgTGCvq/MODOC>

答疑

2. 自由问答

THANKS! |  极客大学