今天学习的内容

1:复习上周重点

2:今天的目标

2.1:完成worker 练习

完成用户输入数值计算累加和

2.2:html5新特性 websocket(重点)--多对多聊天室

HTTP:超文本传输协议

HTTP作用:传输网页中资源(html;css;js;image;video;..)

HTTP是浏览器搬运工

HTTP 工作方式:请示与响应

浏览器发服务器信息(请求数据)

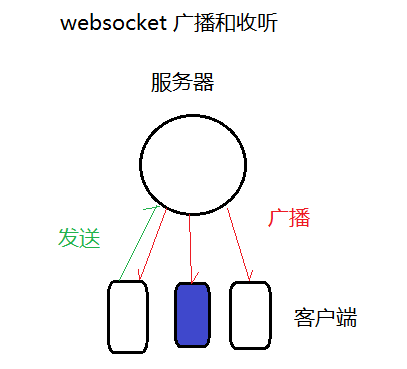
服务器发数据给浏览器(响应消息)

#一次请求对应一次响应，如果没有请求就没有响应

#金融行业(股票/期货/汇率/...) 多长时间请求一次

新解决方案方式:新协议 websocket

websocket工作方式:广播和收听



适用范围:金融;弹幕;聊天室;客服

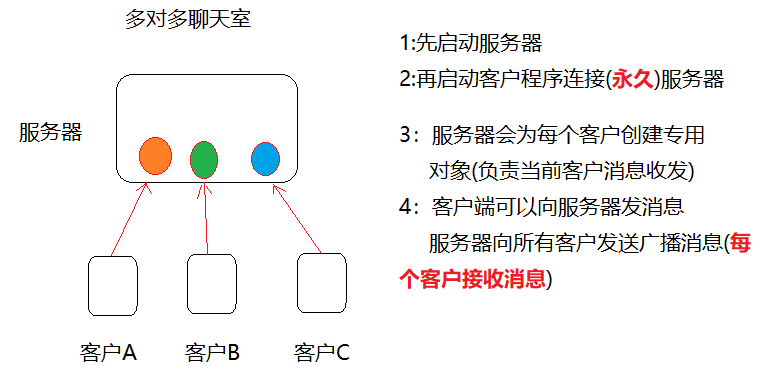
2.3:html5新特性 websocket(重点)--使用

服务器端:node.js scoket.io 最方便最简单

客户端:浏览器原生对象 websocket功能较少

第三方:socket.io

#socket.io 多对多聊天室



#服务器端 下载第三方模块 (socket.io) 02\_socketapp.js

npm i socket.io

#客户端 下载第三方模块 (socket.io) 02\_socket.html

https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.0.3/socket.io.js

#服务器端程序

(1)创建web 服务器

var app = require("http").createServer();

(2)创建socketio 服务器端对象

var io = require("socket.io")(app)

(3)监听端口

app.listen(3000);

(4)为io对象绑定事件 connection (当客户连接服务器触发此事件)

io.on("connection",(socket)=>{

#console.log("有客户端连接服务器....");

#socket 为当前客户创建专属服务员...

})

#客户端程序

(1)创建socket对象并且连接服务器

var clientSocket = io("ws://127.0.0.1:3000");

#ws: websocket

#127.0.0.1 服务器端程序所有机器地址

#3000 服务器运行端口号

#==========================================

#客户端发送数据给服务器 (一对一)

#socket.io 发送与接收数据采用方式事件绑定与事件触发机制

服务器(收)

socket.on("message",(data)=>{console.log(data);})

客户端(发)

clientSocket.emit("message","初始见面多关照");

#emit:触发指定事件

#==========================================

#服务器发送数据给客户端(一对一)

客户端(收)

clientSocket.on("server",(data)=>{})

服务器(发)

socket.emit("server","发送消息");

#==========================================

#广播消息:服务器向所有客户发送消息

客户端(收)

clientSocket.on("list",(data)=>{})

服务器(发)

io.emit("list","明天开始大降价");

#==========================================

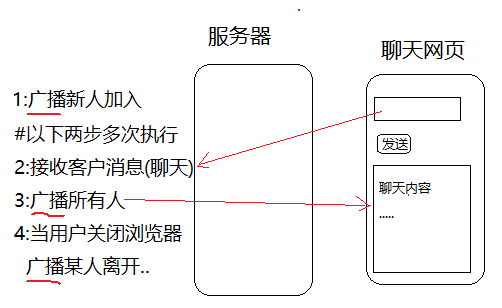
服务器端默认事件:disconnect

#当客户端关闭浏览器默认触发事件(客户端离开了)

2.4:html5新特性 websocket(重点)--多对多聊天室

server/03\_chart\_app.js

client/03\_chart.html



2.4:html5 新特性 webStroage

解决问题:在客户端浏览器中保存用户专用数据

作用:登录用户昵称;购物车购物车;...

#注意事项:webStorage 客户端不要存储安全级别高数据

#安全级别高数据 用户密码;uid

|  |  |
| --- | --- |
| 客户端存储方式 | 服务器端存储方式 |
| (不安全;数据量很少) | (安全/海量/高速查询) |
| (1)cookie  优点:兼容性好  缺点:操作复杂;1kb | (1)数据库 |
| (2)Flash  缺点:依赖flash播放器 | (2)session 对象(数据量少) |
| (3)WebStorage  优点:操作简单;8MB  缺点:兼容性 | (3)文件 data.txt |
| (4)IndexDB 不标准 | (4)内存数据库,文档数据库 |

WebStorage 存储数据依赖二个对象

#sessionStorage

sessionStorage对象存储范围在当前会话中.当前会话中所有网页

都可以使用sessionStorage 保存,但是如果会话结束数据丢失!

注意事项:会话结束(关闭浏览器-关闭标签),退出浏览器生效

#localStorage

此对象保存数据永久保存

WebStorage 操作数据方法与属性以上两个对象通用

(1)保存数据 setItem(key,value); #key键 value 值

(2)获取数据 var value = getItem(key);

(3)删除指定数据 removeItem(key);

(4)清空所有数据 clear()

(5)数据长度 length

练习:02\_save.html 保存数据 03\_get.html 获取数据

2.5:html5 新特性 网页游戏

#分析功能实现

-蓝色大海背景

-紫色海葵左右摆动(重点-难点)

-海葵创建食物(难点)

-大鱼负责吃食物 吃蓝色100分 吃橙色200分

-小鱼跟着大鱼游戏

2.6:目录结构

game 游戏主目录

+ js 游戏代码

+ src 游戏所有图片

+ index.html 入口文件

2.7:创建游戏基本代码

-复制游戏中所有图片保存src目录

-创建index.html

画布一:(前)大鱼;小鱼;分数 800\*600 z-index:1

画布二:(后)背景;海葵;食物 800\*600 z-index:0

作业:参数提供示例代码:--背景大海--海葵