# 面试

**设计模式和继承？？？？？**

**H5新特性**

闭包（一句话概括：）

排序

对象排序

Jq动画

数组去重



Dpi：2x，3x

**冒泡捕获：**

捕获：从外到里

冒泡：从里到外

**盒子模型**

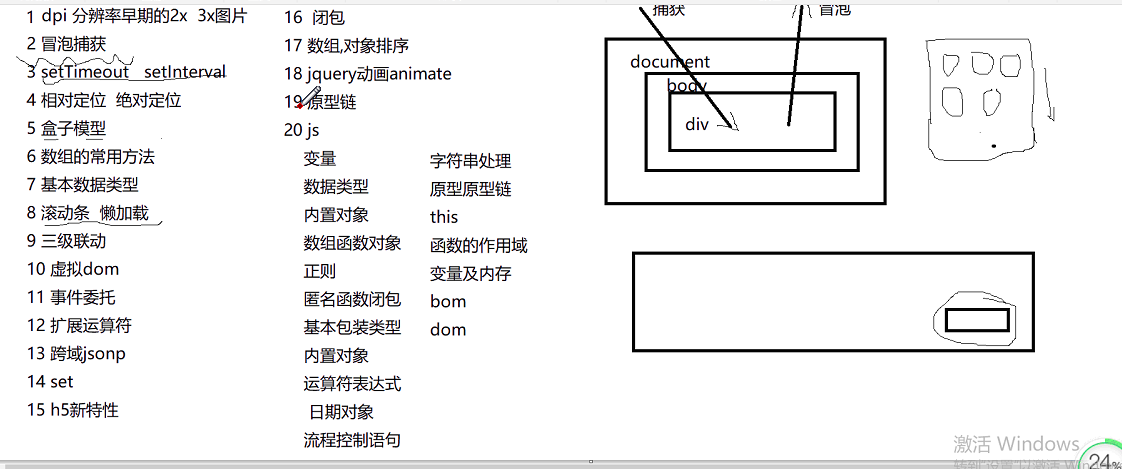
**margin+border+padding+content**

**数组的常用方法**

**滚动条+懒加载**

**三级联动**

扩展运算符：。。。

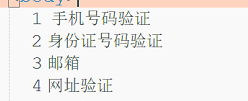


Js的封装继承多态

## PHP返回的类型



## 课堂作业



npm install---安装

## 看看：Css3写loading加载

background-position

### 小实验

绝对定位会不会依照浮动定位来定？？？？？

绝对定位若在浮动定位里面会根据其定位，反之不会

测试一下flex-shrink

**上拉下滑**



懒加载

# 事件

1. 移动端：
   1. **ontouchstact**
   2. **ontouchmove**
   3. **ontouchend**
   4. **ontouchcancel**
2. PC端：
   1. 鼠标事件
   2. 键盘事件
   3. 画笔

# 动画（animation）和过渡（transition）2D变形（transform）

animation（动画）：（由@keyframes创建的动画名称）（动画时间）（动画方式）（延迟时间）（动画的播放次数：（infinite：无限次））（动画的播放方向：（1.normal：循环播放2.alternate：偶数次向前播放））（动画的播放状态：（1.running：将暂停的开始播放2.paused：播放的暂停））（动画开始之前和结束之后发生的操作）

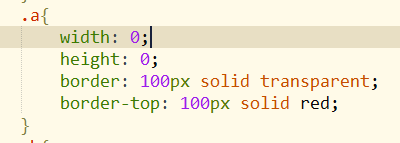
transition（过渡）：（过渡或动态模拟的css属性）（完成过渡所需要的时间）（指定的过渡函数：（1.ease：默认由快到慢2.linear：匀速3.ease-in：越来越快4.ease-out：越来慢5.ease-in-out：先快后慢））（过渡开始出现的延迟时间）

transform：（2D变形）：（平移：translate）（缩放函数：scale()）（旋转函数：rotate()）（倾斜函数：skew()）

# 作业：

**css3做三角形、动画（自荐）、自己的项目**

**三角**：宽高为0，transparent：透明色



# 布局(横向排列)

1. 浮动
2. flex布局
   1. flex：1；等分比例
   2. 垂直对齐：align-
   3. 两端对齐

3.一个布局，一个浮动（浮动那个用margin-left==第一个的宽，让个位置）

# a标签如何清除默认跳转行为

使用 **javascript：;** 或者 **javascript：void(0);**

# from表单取消默认行为

**e.preventDefault()**

# 阻止冒泡

**e.stopPropagation()**

# 清除浮动

1. 给父级添加高
2. 给父级添加伪类
3. 最后一个浮动元素后面添加空div
4. 父类添加overflow：hidden

# 定位区别

1. 绝对定位（absolute）：根据自己定位



1. 相对定位（relative）：根据父级
2. 固定定位（fiexd）：根据浏览器

## **Rem**

1. **rem就是基于html的** 100px=1rem
2. rem主要针对Html依照比例为font-size

# Filx布局

1. 容器的属性
   1. **flex-direction**：排列的方向
      1. row：默认值：主轴为水平方向，起点在左端
      2. row-reverse：主轴为水平方向，起点在右端
      3. column：主轴为垂直方向，起点在上沿
      4. column-reverse：主轴为垂直方向，起点在下沿
   2. **flex-wrap**：怎样换行
      1. nowrap：默认，不换行
      2. wrap：换行，第一行在上方
      3. wrap-reverse：换行，第一行在下方
   3. **flex-flow**：
      1. 是flex-direction和flex-wrap属性的简写，默认值为row nowrap
   4. **justify-content**：项目在主轴上的对齐方式
      1. flex-start：默认：左对齐
      2. flex-end：右对齐
      3. center：居中
      4. space-between：两端对齐，项目之间的间隔都相等
      5. space-around：每个项目两侧的间隔相等
   5. **align-items**：项目在交叉轴上如何对齐
      1. flex-start：交叉轴起点对齐
      2. flex-end：交叉轴终点对齐
      3. center：交叉轴终点对齐
      4. Baseline：项目的第一行文字的基线对齐
      5. Stretch：默认值，若设置高度或auto，将占据整个容器高度
   6. **align-content**：多根轴线对齐方式，若只有一根轴线，不起作用
      1. flex-stare：与交叉轴起点对齐
      2. flex-end：与交叉轴终点对齐
      3. center：与交叉轴中点对齐
      4. space-between：与交叉轴两端对齐，轴线之间的间隔平均分布
      5. Space-around：每根轴线两侧的间距相等
      6. stretch：默认，轴线占满整个交叉轴
2. 项目的属性
   1. **order**：排列顺序，数值越小，排列越靠前，默认0
   2. **flex-grow**：放大比例，默认0，若果存在剩余空间，也不会放大
   3. **flex-shrink**：缩小比例，默认1，若空间不足，该项目将缩小
      1. 若一个项目的属性为0，其他项都为1，则空间不足时，前者不缩小
   4. **flex-basis**：默认值auto，即项目本来大小
      1. 在分配多余空间之前，项目占据的主轴空间。浏览器根据这个属性，计算主轴是否有多余空间。
      2. 可设为跟width或height属性一样的值，则项目将占据固定空间
   5. **flex**：
      1. 是flex-grow、flex-shrink、flex-basis的简写，默认值为0 1 auto。后两个属性可选
      2. 该属性有两个快捷值：auto(1 1 auto)和none(0 0 auto)
      3. 建议优先使用
   6. **align-self**：
      1. 允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items。
      2. 默认auto，表示继承父元素的align-items属性，若没父元素，则等同于stretch

# display：none和visibility：hidden的区别？

都是

**dispalay**：让元素不可见

1. 该元素以及所有的后代元素都会隐藏；
2. 隐藏后的元素无法点击，无法使用屏幕阅读器等辅助设备访问，占据的空间消失

**visibility**：

1. 隐藏之后的元素仍占用与未隐藏是一样的空间
2. 具有继承性：父元素设置之后visibility之后子元素会继承这个属性，重新为子元素设置成visibility：visibility之后就会显示出来
3. 不会影响计时器计数
4. 过渡（transition）支持visibility属性，不支持display，可使用纯css实现hover延时显示效果。提高用户体验

# 音频：audio

1. 循环：loop
2. 静音：muted
3. 显示组件：controls
4. autoplay：出现该属性，音频在就绪后马上播放
5. preload：出现该属性，音频在页面加载是进行加载，并预备播放，若使用autoplay忽略

## H5兼容

视频不兼容

## H5语义化

理解起来比较方便，结构清晰，对搜索引擎友好

## 对WEB和W3C的

## Layer弹框

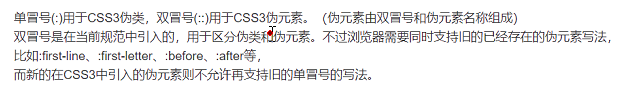
# link和@import区别

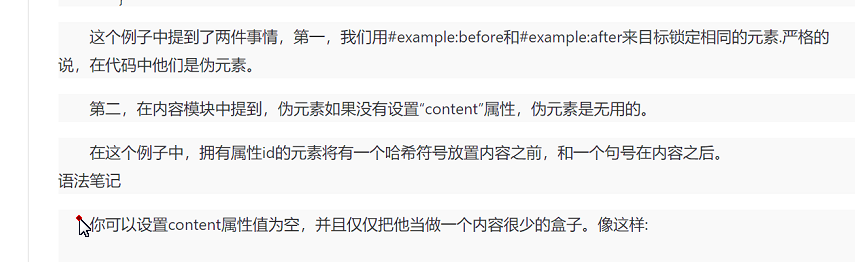
1.link引入的css会同时加载；@import引入的css将在页面加载完后被加载



# 伪类中单冒号和双冒号

1. 区别
   1. 单冒号：伪类
   2. 双冒号：伪元素
2. 现在规范用双冒号，其中有一个content属性





# Html5新增标签

布局用的：header、aside、section、footer、nav、artive

1. 不继承父亲

slick官网

# window.onload：dom加载之后再去执行

# Window.onresize：窗口监听大小（**页面发生变化或刷新时执行**）

写样式之前要先把内外边距去掉：margin：0； padding：0

# pc端怎么适配移动端

1. 禁止移动端缩放(meta)
2. Sc3媒体适配 madel

# label

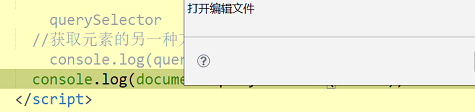
快速获取焦点

# This，分页，上拉加载，下拉刷新，，

# 小知识点

1. **浮动的元素会脱离文档流**
2. **document.documentElement.clientWidth：获取文档宽**
3. **常见的px em rem vh vm**
4. **代码格式回退：shift+tab**
5. **em继承父亲**
6. **ajax会跨域**
7. **Table自带排序**
8. **0表示boolean中的false**
9. **！important：级别更高**（在样式后面加）
10. **text-indent：首行缩进**
11. **letter-spacing：1em：设置字体间距**
12. **中心点：transform-origin：**
13. **empty()方法：从被选元素中移出所有内容，包括所有文本和子节点**
14. **String使用减号时默认自动转换**
15. **localStorage：本地存储**
16. **undefined==null**
17. **数据类型只能比较值，不能比较数据类型**
    1. **空字符串，null，undefined，0，，转换成boolean是false**
    2. **空数组‘[ ]’，数组{ } 可以转换成boolean的true**
    3. **八进制以0开头**
    4. **十六进制以0X开头**
    5. **NaN：（not a number）：非数值**
    6. **NaN不等于NaN**
    7. **ISNaN判断是否是NaN**
18. **parseFloat不认十六进制**
19. **PHP去除警告：**
    1. **error\_reporting(E\_ERROR)**
    2. **Ini\_set(“display\_errors”,”Off”);**
20. **显示所有post里面的内容：print\_r($\_POST)**
    1. 只要有NaN都是NaN
    2. 100-null=0(null转换成0)
    3. 两个 ‘！！’转化之后表示boolean的true
21. **JS中的join()：用于把数组中的所有元素放入一个字符串**

**元素是通过指定的分隔符进行分隔的**

1. **堆是先进先出；栈是先进后出**
2. **获取未来元素的点击事件：.on、.delegate()**
3. **Js不存在块级作用域**（for循环外面也能检测到var里面定义的值，可以用let解决）
4. **delegate()：**
   1. **方法为指定的元素添加一个或多个事件处理程序**
   2. **delegate()方法的事件适用于当前或未来的元素**
   3. **$(selector).delegate(‘ul li’，’click’，function(){alert(111)})**
5. **on()：**在被选元素级子元素上添加一个或多个事件处理程序
   1. **on()方法添加的事件适用于当前及未来元素**
6. **内置方法都是函数对象**
7. **引用数据类型（对象）可以随意添加属性，基本数据类型不能添加属性**
8. **Js没有引用传参，只有值传递**
9. **typeOf只能判断出object还有function，判断不出精确的数据类型**
   1. **可以使用instanceof来判断**
10. **对象里必须是键值对**
11. **for(var i=0;i<5;i++){} 是没有块级作用域**
12. **函数的参数都是局部变量**
13. **{}中必须是键值对**
14. **只有函数对象才有原型**
15. **声明式和命令式**
16. **面向对象与原型**
    1. **使用prototype可以用来查看方法的所有属性：**console.log(Number.prototype)
    2. **创建工厂模式一定要返回对象**
    3. **构造函数创建对象在使用的时候需要new关键字**
    4. **双等号是比较值，若是引用类型比较地址，基本数据类型比较值和类型**
    5. **只有函数对象才有原型**
    6. **原型目的：共享属性和方法**
    7. **每个对象上都有\_\_proto\_\_**
    8. **构造函数都是大写**
    9. **如果实例里面和原型上有相同的属性和方法的时候，就用实例里面的属性和方法（就近原则）**
    10. **判断属性是在原型里面还是在实例里面：hasOwnProperty()**
    11. **构造函数必须使用new运算符来调用，否则就是普通函数**
    12. 字面量创建方式使用constructor属性不会指向实例，而会指向object，构造函数创建相反
17. **Call()和apply():属于继承**
18. **构造函数和普通函数：构造函数开头大写**
19. **闭包**
    1. 闭包可以访问局部变量，并把它驻留在内存里
    2. 局部变量无法累加，使用闭包后可以
    3. let保存作用域，值出来之后才能用
    4. 闭包中的this指向window
    5. 闭包会导致内存泄漏
    6. 闭包+原型+通过实例对象提供特权方法（this.get/this.set）
    7. 作用域链？？？
20. 变量提顶、块级作用域(var let)、、、
21. 函数没返回值会返回undefind
22. **append和prepend：**append：后面插入元素；prepend：从前面插入
23. **静态属性和方法**
    1. 原型上都是静态方法
    2. 静态方法是指不需要声明类的实例就能
24. **有构造函数的地方都有原型**
25. **字面量本身就是单例**
26. **数组深拷贝和浅拷贝**
27. **获取元素的另一种方法**
    1. **querySelectorAll():**
    2. **document.querySelector():**
    3. 
28. 虚拟dom是js
29. Cdata
30. **Js兼容**
    1. **早期的浏览器获取事件对象的时候IE：window.event W3C：e(e值得是事件函数里面的时间对象)**
    2. **获取元素大小，getComputedStyle()**
31. **怪异模式会发生什么事？？？？？？？？？**
32. **Dom1、2、3、级，查看一哈**
33. **outerText不兼容，outerHTML**
34. **通过\_\_proto\_\_实现原型链的，**
35. **this指向：window、匿名函数、对象调用、构造函数、事件对象**
36. **addEventListener(‘事件’，‘函数’，false)：三个参数，默认冒泡(false)；true(捕获)**
    1. 事件默认形式都是冒泡
    2. 阻止冒泡：**e.stopPropagation()**
    3. **事件委托(事件代理)**
    4. **匿名函数+闭包**

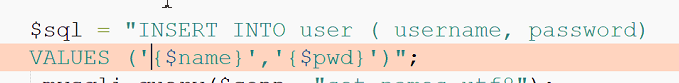
**Ajax中dataType：后端传给前端的数据类型**

**Bootstrap**

Php:查询数据、插入数据、

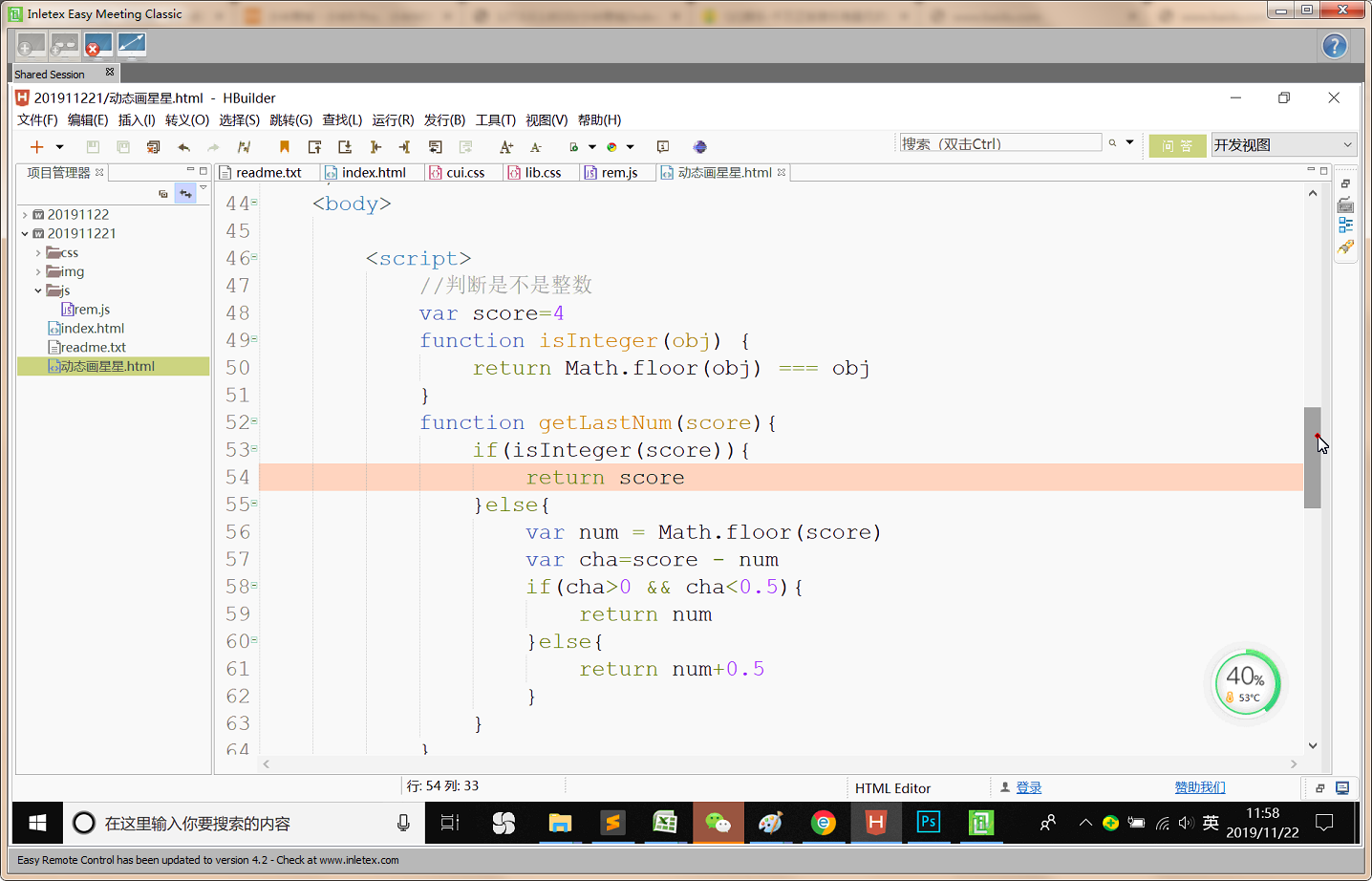
echo：输出

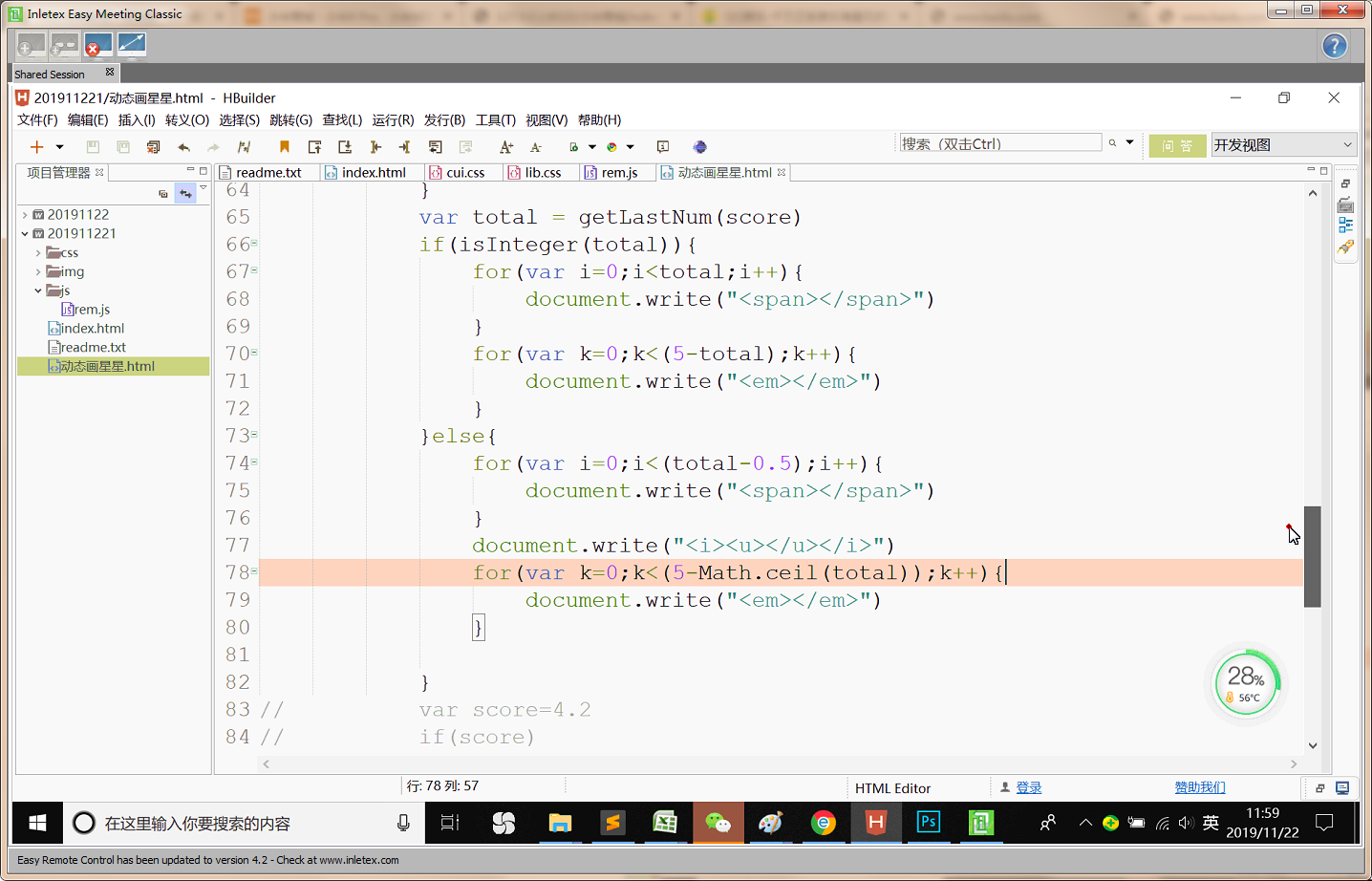
Php中“.”相当于“+”，里面的变量用花括号扩起来



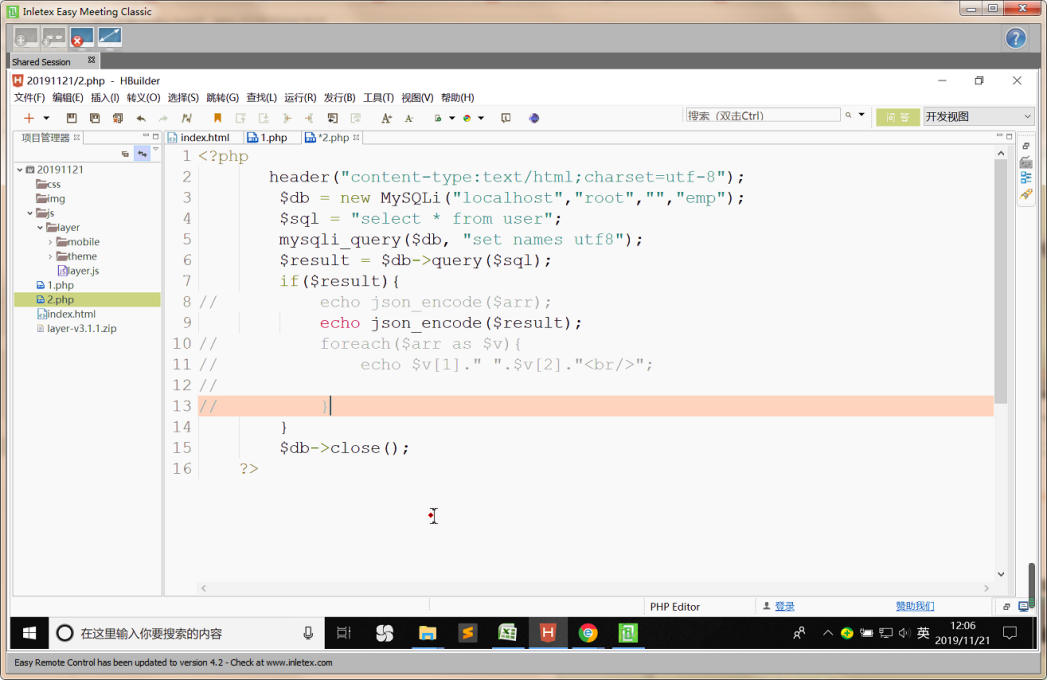
## 跑马灯



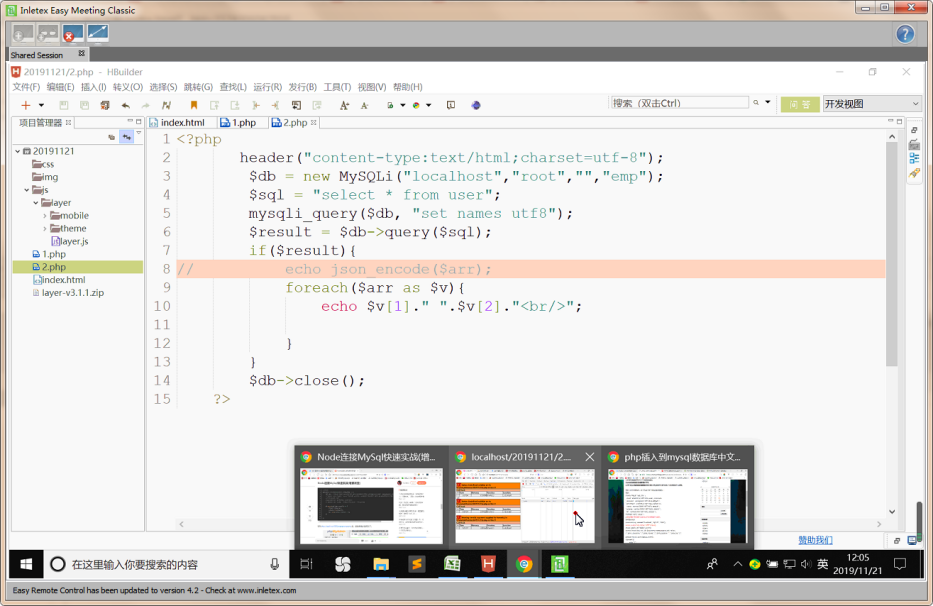
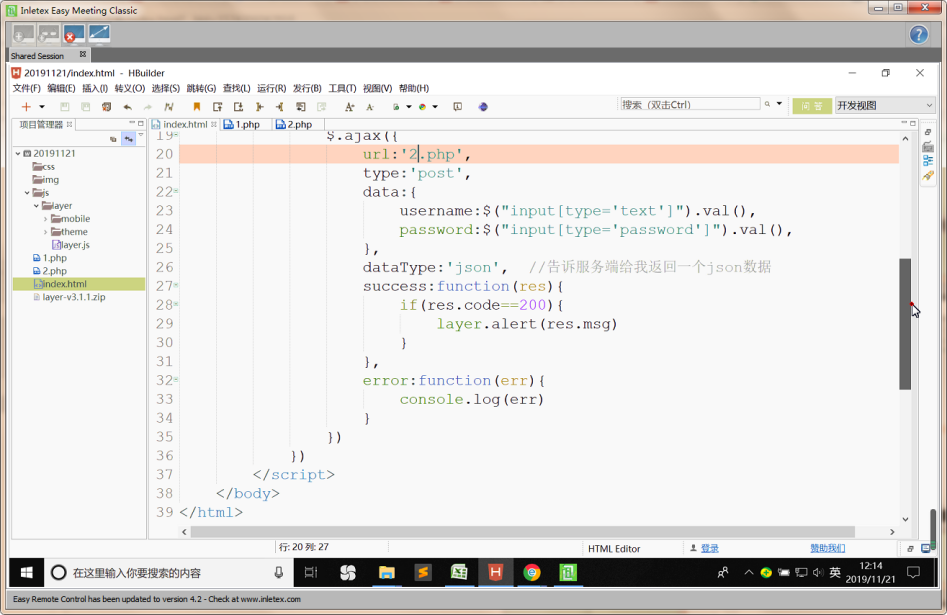




Ctrl+ shirt +alt +s：ps保存

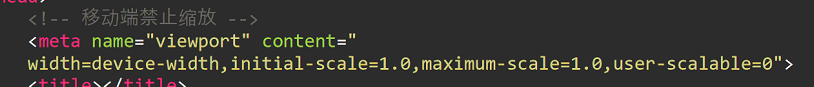


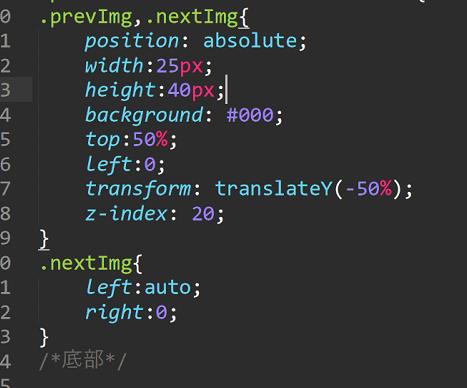
Fetch\_all

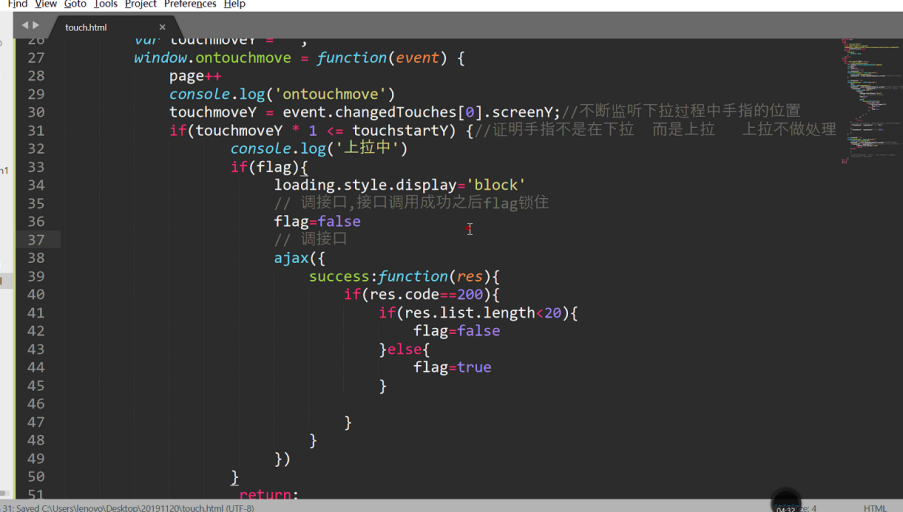


**layer**

移动端禁止缩放

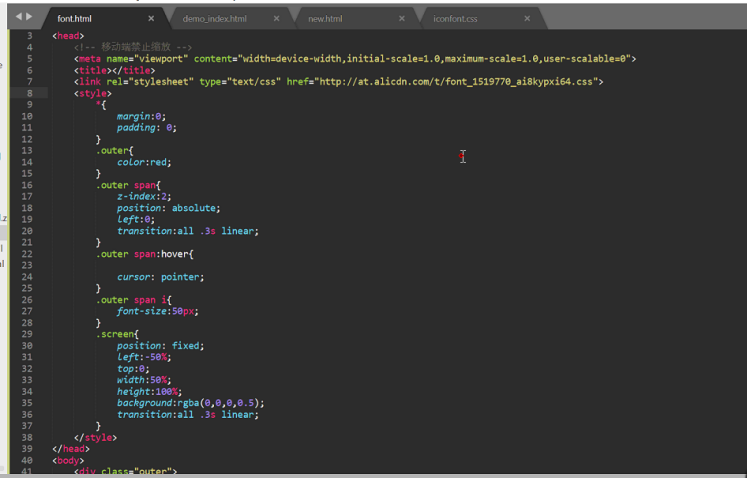


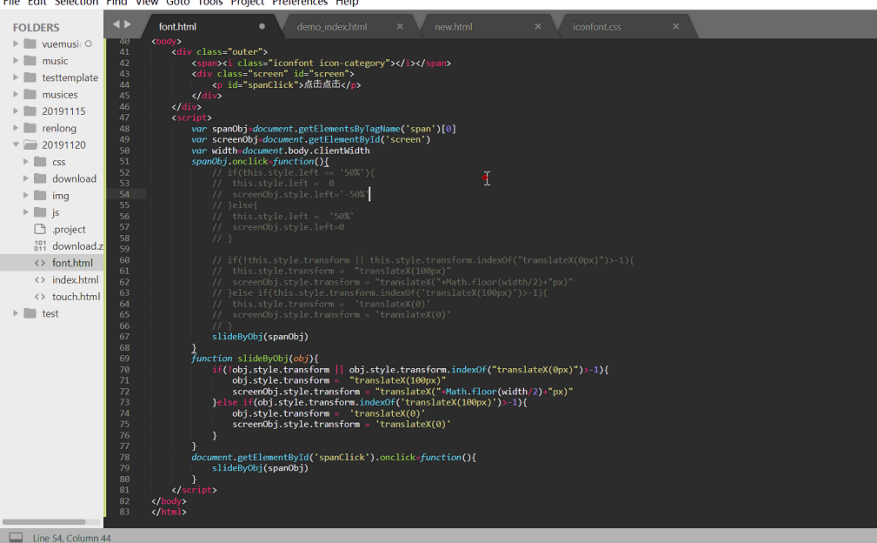




## 手机端左右移动

手机端模仿QQ侧边栏划入划出





京东

