

4月2日-4月8日

1.学习

css方面

1.
 - 接触项目列表 (ul/ol) 属性

- 列表样式

```
list-style-type:disc|circle|square|decimal|lower-roman|uperalpha;
```

- 列表位置

```
list-style-position:outside|inside;
```

- 列表图片

```
list-style-image:url();
```

2.
 - transform属性

- translate平移

```
transform:translate(x,y);      //2d平移
transform:translate3D(x,y,z);  //3d转换
transform:translateX();
transform:translateY();
```

- roate(deg)旋转

```
transform:roate(angle);        //正数顺时针旋转
transform:roate(x,y,z);        //3d旋转 (未尝试)
transform:roateX(angle);       //绕x轴3d旋转 (未尝试)
```

- scale(浮点数)缩放

```
transform:scale(x,y);          //2d缩放
transform:scale3d(x,y,z);
transform:scaleX(x);
```

3.
 - transition属性

- property:none|all|定义应用过渡效果
- duration:time (过渡时间)
- timing-function:linear|ease|ease-in|ease-out|ease-in-out|cubic-bezier(n,n,n,n)(指定转速曲线)
- delay:time(延迟开始)
- 例子

```
div{
  width:100px;height:100px;
  transition:width 2s;
  transition-timing-function:ease-in;
  transition-delay:2ms;
}
div:hover{
  width:200px;
}
```

4. ◦ animation属性

- @keyframes{ from{} to{} }

```
@keyframes{
  0%{top:0px}
  50%{top:200px;}
  100%{top:0px;}
}
```

- animation-name(定义动画名)
- animation-duration(动画时长)
- animation-timing-function(转速曲线)
- animation-delay(延迟)
- animation-iteration-count: number|infinite(播放次数)
- animation-direction:normal|reverse|alternate|alternate-reverse|initial(是否轮流/反向播放)
- animation-fill-mode:none|forwards|backwards|both(播放后保留的样式)
- play-state:paused|running(指定动画是否在运行/暂停) //未尝试

5. ◦ flex布局

- 语法

```
box.parent{
  display:flex;
  flex-direction:row|row-reverse|column|column-reverse;
  //主轴方向
  flex-wrap:wrap|nowrap|wrap-reverse;
  //主轴是否换行
  flex-flow:                //综合上面两项
  justify-content:flex-start|flex-end|center|space-between|space-around;
  //项目在主轴的对齐方式
  align-items:flex-start|flex-end|center|baseline|stretch;
  //交叉轴对其方式
```

```

    align-content:flex-start|flex-end|center|space-between|space-around;
    //多主轴对齐
}
box.child{
    order:integer;
    //项目排序
    flex-grow:integer;
    //放大比例
    flex-shrink:(1~0);
    //缩小比例
    flex-basis:<length>|auto;
    //分配前占据空间
    flex:                //从何上面三项
    align-self:auto|flex-start|flex-end|center|baseline|stretch;
    //单个项目交叉轴对其方式与其他不同
}

```

js方面

1. js基础语法
2. js DOM树

- 常用属性

```

nodeName    //节点名称
             //文档#document 文本#text
nodeType    //节点类型
             //1元素 2属性 3文本
nodeValue   //文本节点的值 或 属性节点的值
childNodes  //元素节点所有子元素
firstChild
lastChild
nextSibling

```

- 常用DOM方法

```

getElementById("")           //获得一个元素节点
getElementsByName("")        //获得一个数组
getElementsByClassName("")    //获得一个数组
getAttribute("attribute")    //获得属性
setAttribute("attribute",value) //更改属性

```

- 常用DOM事件

```

onload
onclick
onmouseover
onmouseout
onfocus
onblur

```

3. 制作图片库
4. 分离HTML和js，渐进增强，平稳退化，向后兼容性
5. 常用脚本

- addLoadEvent()添加页面加载完成时执行函数

```
function addLoadEvent(func){
    var ondonload=window.onload;
    if(typeof window.onload!='function')
    {
        window.onload=func;
    }
    else
    {
        window.onload=function(){
            oldonload();
            func();
        }
    }
}
```

- insertAfter()再在目标节点后插入节点

```
function insertAfter(newElement,targetElement){
    var parent=targetElement.parentNode;
    if(parent.lastChild==targetElement)
    {
        parent.appendChild(newElement);
    }
    else
    {
        parent.insertBefore(newElement,targetElement.nextSibling);
    }
}
```

css问题

1. 把animation动画放在上，无法实现展开图片的效果，因为动画是直接放大图片。
 2. 使用flex布局能轻易实现有规则的定位（如水平垂直居中），但当flex的项目类型不同时，难以准确定位，此时用position定位比较便捷。
 3. 多个div页面排版时，可否先用绝对定位确定div的位置，再对div相对定位，方便div内的元素使用绝对定位。
 4. 有没有布局方法能让页面的内容（包括div，img，文本）在用户调整窗口大小时位置不发生改变？
- 待实践：将每个最外部的div的width和height设定为固定px，并且设定绝对定位
5. 把内容放在一个div中，外面再套一个不设置宽高的div

js问题

1. <textarea>实现鼠标点击时清空原有内容，鼠标移开时内容重置。

- 已解决：使用onclick/onmouseout事件，分别定义两个函数更改标签的firstChild的nodeValue值
2. 一个HTML文档中可以有几个不同的id属性？使用多个id会有什么后果？
 3. 用document.getElementsByClassName()时函数总是无法被执行。
- 已解决：此函数获得的时一个数组，和getElementsByTagName一样，需要再变量后加下标[0]。
4. 模仿图片库，将点击链接更换图片改为更换div，第一次点击是更换成功，第二次点击时网页跳转了？
(突然懵逼)