# 01 - JavaScript Drum Kit

1. Dom的Classlist属性可以用来增删html的class属性
2. ${}可以拼接字符串 es6特性 如doucument.querySelector(“div[data-key =${}]”)
3. Event.propertyName 可以判断当前transitionend监听事件的样式是否是自己想要的样式，如果不是则不进行处理

# 02 - JS + CSS Clock

1. 时钟效果主要靠Date()对象和rotate()样式来实现
2. 记得用transform-origin来保持中心在表盘中心。
3. 可以用transition-timing-function来控制秒针移动效果
4. box-shadow有5个属性分别为
   1. Inset (可选) Box里面
   2. X轴偏移量 px
   3. Y轴偏移量 px
   4. 发散度 px
   5. 发散距离 px
   6. 颜色

# 03 - CSS Variables

1. Css中fliter的用法：
   1. // filter: blur(5px); 模糊度
   2. filter: brightness(0.4); 亮度
   3. filter: contrast(200%); 抽象都
   4. filter: drop-shadow(16px 16px 20px blue); 边框阴影 : x ，y ，阴影范围，颜色
   5. filter: grayscale(50%); 灰度
   6. filter: hue-rotate(90deg);
   7. filter: invert(75%); 色调反转
   8. filter: opacity(25%); 透明度
   9. filter: saturate(30%); 饱和度
   10. filter: sepia(60%); 褐色度
   11. :root伪类，匹配html标签，常用于声明全局的CSS变量
   12. 可以用oninput事件来实时获取input为range类型的拖动值
   13. elment.style.setProperty(css样式，值) 可以用来修改全局css变量

# 04 - Array Cardio Day 1

1. 数组的用法 map(callback(item){}) ：返回一个处理过的新数组

2. filter(callback(item){}) ：返回一个过滤过的新数组

3. sort([callback(a,b)]) ：返回一个排序过的新数组

4. reduce(callback(prev,cur,index,array),[init]) ： 返回一个累加值 ，init可以是对象，可以用来统计数值

5. str.includes("xx") 相当于contains()的用法，查找字符串中是否有某个字符

6.arr.includes(“xx”) 是查找数组中是否有某个元素

# 05 - Flex Panel Gallery

1. 可以给\* ，\*:after ,\*:before { box-sizing ：border-box}
2. Flex的收缩也可以用transition来增加动画效果
3. Cubic-bezier可以制作弹跳效果的动画
4. 100vh就是视窗的高度

# 06 – Type Ahead

1. Promise对象的用法
   1. Promise.prototype.then
   2. Promise.prototype.catch
   3. Promise.prototype.
   4. Promise.all([promise1,promise2]) 全部成功才执行
   5. Promise.race([promise1,promise2]) 一个成功就执行
2. 全局方法fetch（）的用法，
   1. Fetch(url) 去请求，然后返回一个promise对象，对象中参数带有response对象，然后想什么操作再用response去操作
3. 数组的fliter map forEach的用法

# 07 - Array Cardio Day 2

1. Array.prototype.some(function(){})
   1. 返回一个布尔值，只要找到一个满足条件的元素，就返回true
2. Array.prototype.every()
   1. 返回一个布尔值，所有的元素都满足条件，才返回true
3. Array.prototype,find()
   1. 找第一个满足条件的元素，返回此元素
4. Array.prototype.findIndex()
   1. 找第一个满足条件的元素，返回次元素的下标
5. Array.prototype.slice(start,end)
   1. 找返回内的元素，返回新数组
6. Array.prototype.splice(start,range,replace)
   1. 找元素，然后删除，或者替换，返回删除或者替换的元素，原数组被改变。

# 08 - Fun with HTML5 Canvas

1. 简单的获取滚动条到顶部(左部)的距离：
   1. let top = document.documentElement.scrollTop || document.body.scrollTop;
   2. let left = document.documentElement.scrollLeft || document.body.scrollLeft;
2. HSL可以用来方便的制造彩虹色
   1. H代表色调，取值为0-360
   2. S为饱和度，取值为0-1
   3. L为亮度，取值为0-1
3. Js中的角度都得用弧度表示，所以180度得用Math.PI
4. 数字先慢慢变大再慢慢变小的逻辑，可以用一个代表方向的flag来帮忙
5. 画布的大小可以在Js中设置等于window.innerWidth 和 window.innerHeight

# 09 - Dev Tools Domination

1. 在按 F12 出现的 Chrome 开发工具中，在 Elements 选项卡之中，选择页面的某个标签（以 <p>为例），右键 → Break on → Attributes modifications。即可为该元素添加断点，当它的属性发生改变时，会自动定位到页面代码中的对应行。
2. 关于console调试的一大堆用法
   1. Console.log
      1. 替换字符 %s
      2. 替换整数 %d
      3. 替换小数 %f
      4. 替换样式%c
   2. Console.dir
      1. 打印dom元素的属性
   3. Console.warn
   4. Console.error
   5. Console.info
   6. Console.table
   7. Console.assert()
      1. 如果第一个返回false，就以错误的形式打印第二个参数，
   8. Console.clear() 也可以用ctrl + L直接清空
   9. Console.group /console.groupCollapsed()
      1. 配合console.groupEnd 进行分组
   10. Console.time
       1. 配合console.timeEnd 进行计时，参数必须保持一样

# 10 - Hold Shift and Check Checkboxes

1. 做一个flag来保存上一个被checked的元素，然后被监听的元素如果是上一个被点击的或者第二次被点击的元素时把默认为false的inbetween的flag弄为! inbetween,如果是true的话，表示此元素在第一个点击和第二个点击的元素中间。

# 11 - Custom Video Player

1. Element.getBoundingClientRect() 可以获取当前元素相对于窗口的相对位置和长宽
2. clientWitdh 和 clientHeight 获取元素**可视**的不包括边框的宽度和高度
3. offsetWitdh 和offsetHeight 获取元素**可视**的包括边框的宽度和高度
4. offsetLeft 和offsetTop 获取元素到父元素的距离
5. scrollWidth 和 scrllHeight 获取元素的**实际的**不包括边框的宽度和高度（没有滚动条时等于clientXXX）
6. timeupdate可以用来监听video的currentTime是否被改变了

# 12- Key Sequence Detection

1. event.key可以监听到物理键盘真实的输出结果
2. event,.keyCode 已经不推荐使用了
3. event.code 可以监听到按键的编码

# 13 - Slide in on Scroll

1. 获取滚动条滚动过的距离
   1. Document.documentElement.scrollTop || document.body.scrollTop
2. 解构赋值获取getBoundClientRect方法返回的对象很方便
3. Lodash提供了可以降低监视频率的方法，具体源码很牛逼
4. Undefined 和null都是false ,所以 !undefined 和 !null 都是true

# 14 - JavaScript References VS Copying

1. 对数组和对象的复制都是复制引用。