

注：

1. 每题 10 分，答题时间 40 分钟。
2. 不作答的题不给分，对于作答不完整，但思路书写清晰的也可给满分。

1. 分别写出以下输出结果

```
var aAry = [1, 2, 3];
var bAry = aAry;
aAry[0] = 8;
bAry[0] = 9;
console.log(aAry, bAry); //写下输出结果
aAry = [4, 5, 6];
console.log(aAry, bAry); //写下输出结果
```

2. 点击 a、b、c 分别输出

```
<ul>
  <li>a</li>
  <li>b</li>
  <li>c</li>
</ul>
<script>
  //点击 a、b、c 分别输出？
  let li = document.querySelectorAll("li");
  for (var i = 0; i < li.length; i++) {
    li[i].addEventListener("click", () => {
      console.log(i);
    })
  }
</script>
```

3. 反转 container 里面的子元素,前提：不使用 innerHTML

```
<ul class="container">
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4</li>
  <li>5</li>
  ...
  <li>1000</li>
</ul>
```

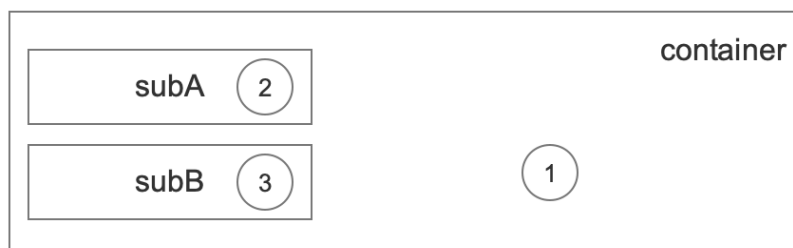
4. 写出以下输出结果

```
function Animal() {
  this.name = "animal";
  this.say = function () {
    console.log(this.name);
  }
}
Animal.prototype.look = function () {
  console.log("Animal look");
};
function Dog() {
  this.name = "dog";
}
Dog.prototype = Object.create(Animal.prototype);
var dog = new Dog();
//写出输出内容
dog.look();

//写出输出内容
dog.say();
```

5. 点击图中 1, 2, 3 处分别输出什么内容?

```
<body>
<div class="container">
  <div class="subA">subA</div>
  <div class="subB">subB</div>
</div>
<script>
  document.querySelector(".subA").addEventListener("click", (e) => {
    console.log("subA click");
    console.log(e.target.getAttribute("class"));
  });
  document.querySelector(".container").addEventListener("click", (e) => {
    console.log("container click");
    console.log(e.target.getAttribute("class"));
  })
</script>
</body>
```

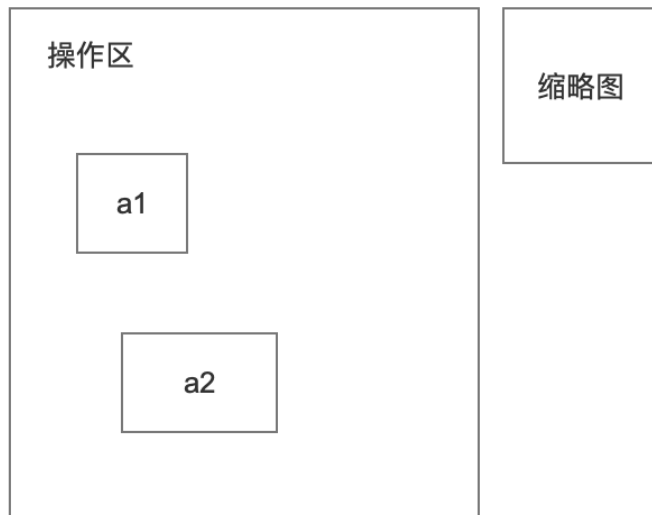


6. 利用数据生成对应的 dom 结构

```
<body>
<div id="container"></div>
<script>
  //假如以下有 1000 个节点，请将以下数据解析生成对应的 dom 结构
  var treeData = [
    {"name": "A-1"},
    {"name": "A-2"},
    {"name": "A-3"},
    {
      "name": "A-4",
      "sub": [
        {"name": "A-4-1"},
        {"name": "A-4-2"},
        {
          "name": "A-4-3",
          "sub": [
            {"name": "A-4-3-1"},
            {"name": "A-4-3-2"}
          ]
        }
      ]
    }
  ]
};
//在下面空白处写下通用方法

</script>
</body>
```

7. 根据需求，给出优化方案



需求：

如图左边为操作区，右边为缩略图。当用户在左边区域操作 dom 时，右边缩略图能实时更新
已知条件：

1. 截图非常耗性能
2. 左侧 dom 中所有元素的信息都有记录，如下。当元素信息改变时，数据也会同步刷新

```
{
  a1:{ "x":100,"y":200,"width":100,"height":100,"color":"#FF0000"},
  a2:{ "x":200,"y":100,"width":50,"height":100,"color":"#000000"},
  ...
}
```

//----- 以下是实现方案（伪代码） -----//

//截图函数，能将 dom 截成一张图片(截图非常耗性能)

```
function domToImage(){
```

```
}
```

```
a1.on("位置改变",function(){
  domToImage();
```

```
})
```

```
a1.on("大小改变",function(){
  domToImage();
```

```
})
```

```
a1.on("颜色改变",function(){
  domToImage();
```

```
})
```

```
...
```

请根据以上条件优化此方案

8. 利用 mousedown mousemove mouseup，实现一个 dom 拖动类，mousedown mousemove mouseup 分别应该注册在哪个 dom 节点上

```
<body>
  <div class="dragDom">
</body>
```

9. 利用自己的掌握的 javascript 知识实现一个能检测页面滚动速度（包括大小与方向）的方法

10. 以下是匀速直线运动，请分别写出匀速圆周运动、自由落体运动

写思路、公式

```
<div class="ball"></div>
<script>
  var ball = document.querySelector(".ball");
  ball.style.position = "absolute";
  var s = 0;
  var v = 1;
  setInterval(() => {
    s += v;
    ball.style.top = s + "px";
  })
  //匀速圆周运动

  //自由落体运动

</script>
</body>
```