下面的 JSX 代码中，哪一个无法达到预期的效果？
正确答案: C　你的答案: 空 (错误)

```
<h2>Hello World</h2>
<input type="checkbox"/>
<div class="msg-box">{msg}</div>
<label htmlFor="name">Leo</label>
<div style={{height: 50}}></div>
<img src={imgSrc}/>
```

正则表达式 /a+(bab)?(caac)*/ ，下列选项中是该正则表达式的子集是？
正确答案: C　你的答案: 空 (错误)

```
/(bab)(caca)/
/a(bab){2}(caac)*/
/a{2}/
/a+(bab){0,1}(ca)+(ca)/
/a(^bab)+(caac){1,}/
/a+(babc){2,}(acc){1,}/
```

下列说法错误的是：
正确答案: C　你的答案: 空 (错误)

在 Blink 和 WebKit 的浏览器中，某个元素具有 3D 或透视变换（perspective transform）的 CSS 属性，会让浏览器创建单独的图层。
我们平常会使用 left 和 top 属性来修改元素的位置，但 left 和 top 会触发重布局，取而代之的更好方法是使用 translate，这个不会触发重布局。
移动端要想动画性能流畅，应该使用 3D 硬件加速，因此最好给页面中的元素尽量添加 translate3d 或者 translateZ(0) 来触发 3D 硬件加速。
解决浏览器渲染的性能问题时，首要目标就是要避免层的重绘和重排。

将数组 var a=[1,2,3] 变成数组 [4,3,2,1] 下面的方式正确的是？
正确答案: A C　你的答案: 空 (错误)

```
a.reverse().unshift(4)
a.push(4).reverse()
a.push(4); a.reverse()
a.splice(3,1,4).reverse()
```

目前 HTTP2 协议已经逐渐普及到日常服务器中，以下对于 HTTP2 协议描述正确的是：
正确答案: A B C D　你的答案: 空 (错误)

所有 http 请求都建立在一个 TCP 请求上，实现多路复用
可以给请求添加优先级

服务器主动推送 server push
HTTP2 的头部会减小，从而减少流量传输

请问下面哪种方式可以在不改变原来数组的情况下，拷贝出数组 b ，且满足 b!=a 。例如数组 a 为 [1,2,3] 。
正确答案: B D 你的答案: 空 (错误)

```
let b=a;
let b=a.slice();
let b=a.splice(0,0);
let b=a.concat();
```

以下代码，分别给节点 #box 增加如下样式，问节点 #box 距离 body 的上边距是多少？
<body style="margin:0;padding:0">

</body>
如果设置 position: static ; 则上边距为 1 px
如果设置 position: relative ; 则上边距为 2 px
如果设置 position: absolute ; 则上边距为 3 px
如果设置 position: sticky ; 则滚动起来上边距为 4 px
你的答案 (错误)

参考答案
(1) 20
(2) 30
(3) 30
(4) 10

我们需要实现动态加载一个 JavaScript 资源，但是有几处不知道如何处理，需要您的帮助完成这一项工作
var script = document.createElement("script");
var head = document.getElementsByTagName("head")[0];

script.type = "text/javascript";
script.src = "//i.alicdn.com/resource.js";

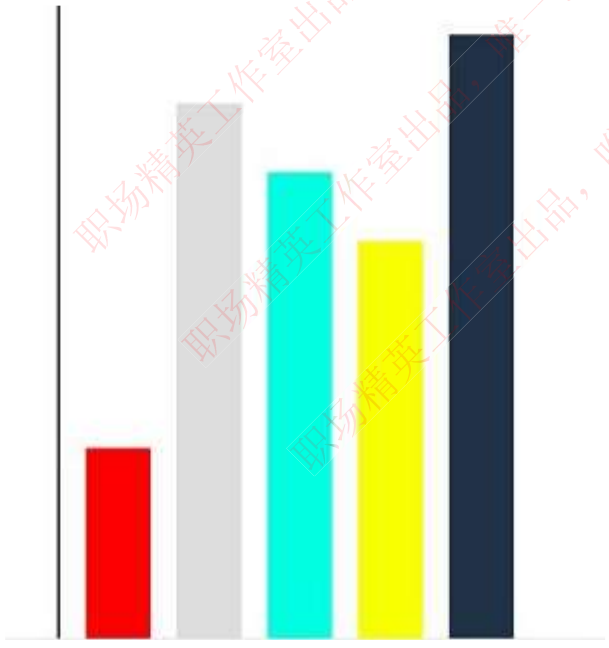
// 绑定资源加载成功事件
script. 1 = function( ){
// 判断资源加载状态是否为加载成功或加载完成
if(2 . test (script. 3)) {
script.onreadystatechange = null;

```
. . . .
      }
};

// 绑定资源加载失败事件
script. 4 = function( ) {
. . . .
};

head.insertBefore (script , head.firstChild)
```

你的答案 (错误)

参考答案
(1) onreadystatechange
(2) /^(loaded|complete)$/
(3) readyState
(4) onerror

请使用两种不同的 CSS 方法（要求 dom 结构不同）实现下图所示的条形图。从左到右的条形分别记为 A,B,C,D,E。A 的高度为 30%，颜色为#f00；B 的高度为 80%，颜色为#ddd；C 的高度为 70%，颜色为#0fd；D 的高度为 60%，颜色为#ff0；E 的高度为 90%，颜色为#234，每个条形之间的距离可以任意设置（可以考虑使用 CSS3 新属性来实现）。



```
* {
   margin: 0;
   padding: 0;
}
#context {
```

```
    width: 500px;
    height: 300px;
    margin: 0 auto;
    display: flex;
    justify-content: space-around;
    align-items: flex-end;
}
.flex-box {
    width: 15%;
}
.flex-box:nth-child(1) {
    height: 30%;
    background-color: #f00;
}
.flex-box:nth-child(2) {
    height: 80%;
    background-color: #ddd;
}
.flex-box:nth-child(3) {
    height: 70%;
    background-color: #0fd;
}
.flex-box:nth-child(4) {
    height: 60%;
    background-color: #ff0;
}
.flex-box:nth-child(5) {
    height: 90%;
    background-color: #234;
}
1
2
3
4
5
6
7
<div id="context">
        <div class="flex-box"></div>
        <div class="flex-box"></div>
        <div class="flex-box"></div>
        <div class="flex-box"></div>
        <div class="flex-box"></div>
</div>
```

请实现方法 parse ，作用如下：

```
var object = {
  b: { c: 4 }, d: [{ e: 5 }, { e: 6 }]
};
console.log( parse(object, 'b.c') == 4 ) //true
console.log( parse(object, 'd[0].e') == 5 ) //true
console.log( parse(object, 'd.0.e') == 5 ) //true
console.log( parse(object, 'd[1].e') == 6 ) //true
console.log( parse(object, 'd.1.e') == 6 ) //true
console.log( parse(object, 'f') == 'undefined' ) //true

function parse(obj,str){

    var arr=str.replace(/\[(\w)\]/g,'.$1').split('.'),i=0,tmp=obj;//把[]都替换成.的形式

    while(i<arr.length&&tmp){

        tmp=tmp[arr[i++]];

    }

    //console.log(tmp)

    return tmp!=undefined?tmp:'undefined';//注意 undefined 要返回字符串

}
```

请问何为混合应用 (Hybrid APP)，与原生 Native 应用相比它的优劣势。

Hybrid APP 翻译过来就是混合 APP。概念：半原生班 web 的混合类 APP，需要下载安装，看上去类似原生 APP，但只有少数 UI Web view ，访问的内容是 web，例如新闻类 APP，视频类 APP 普遍采用原生框架，web 的内容。混合 APP 追求原生 APP 的体验，但仍受限于技术，网速等。

与原生 APP 比优势：成本低、更新快

劣势：操作速度慢，影响用户体验。