

Daily_Learnning(2019-11-21)

输出什么？

```
1 const spookyItems = ["👻", "🎃", "🕷️"];
2 ({ item: spookyItems[3] } = { item: "💀" });
3
4 console.log(spookyItems);
```

- A: ["👻", "🎃", "🕷️"]
- B: ["👻", "🎃", "🕷️", "💀"]
- C: ["👻", "🎃", "🕷️", { item: "💀" }]
- D: ["👻", "🎃", "🕷️", "[object Object]"]

通过解构对象，我们可以从右侧对象中解压缩值，并将解压缩后的值分配给左侧对象上具有相同属性名称的值。在这种情况下，我们将分配值“💀”到spookyItems[3]。这意味着正在修改spookyItems数组，我们添加了“💀”。打印spookyItems，["👻", "🎃", "🕷️", "💀"]会记录。

B

输出啥？

```
1 const myFunc = ({ x, y, z }) => {
2   console.log(x, y, z);
3 };
4
5 myFunc(1, 2, 3);
```

- A: 1 2 3
- B: {1: 1} {2: 2} {3: 3}
- C: { 1: undefined } undefined undefined
- D: undefined undefined undefined

函数有一个形参，是个对象。但是函数执行时只传了三个数字，无法与对象对应。无效参数。但x,y,z属于形参。有默认值undefined。

D

打印啥?

```
1  const add = x => y => z => {  
2      console.log(x, y, z);  
3      return x + y + z;  
4  };  
5  
6  add(4)(5)(6);
```

- A: 4 5 6
- B: 6 5 4
- C: 4 function function
- D: undefined undefined 6

回想箭头函数可以省略大括号，只有一句话的话。可以理解为

```
1  const add = function(x){  
2      return function(y){  
3          return function(z){  
4              console.log(x, y, z);  
5              return x + y + z  
6          }  
7      }  
8  }  
9  最后的函数拥有访问上级函数作用域的权限。so  
10 A
```