Daily_Learning(2019-11-13)

输出什么?

```
function nums(a, b) {
  if
  (a > b)
  console.log('a is bigger')
  else
  console.log('b is bigger')
  return
  a + b
}

console.log(nums(4, 2))
console.log(nums(1, 2))
```

- A: a is bigger, 6 and b is bigger, 3
- B: a is bigger, undefined and b is bigger, undefined
- C: undefined and undefined
- D: SyntaxError

输出什么?

```
const person = {
  name: "Lydia",
  age: 21
}

for (const [x, y] of Object.entries(person)) {
  console.log(x, y)
}
```

- A: name Lydia and age 21
- B: [”name”, ”Lydia”] and [”age”, 21]
- C: [”name”, ”age”] and undefined
- D: Error

解析;

1： B

在JavaScript中，我们不必显式地编写分号(;)，但是JavaScript引擎仍然在语句之后自动添加分号。这称为自动分号插入。例如，一个语句可以是变量，或者像throw、return、break这样的关键字。

在这里，我们在新的一行上写了一个return语句和另一个值a + b 。然而，由于它是一个新行，引擎并不知道它实际上是我们想要返回的值。相反，它会在return后面自动添加分号。你可以这样看:

```
  return;
  a + b
```

这意味着永远不会到达a + b，因为函数在return关键字之后停止运行。如果没有返回值，就像这里，函数返回undefined。注意，在if/else语句之后没有自动插入!

2: A

Object.entries()方法返回一个给定对象自身可枚举属性的键值对数组，上述情况返回一个二维数组，数组每个元素是一个包含键和值的数组：

```
[['name', 'Lydia'], ['age', 21]]
```

使用for-of循环，我们可以迭代数组中的每个元素，上述情况是子数组。 我们可以使用const [x，y]在for-of循环中解构子数组。 x等于子数组中的第一个元素，y等于子数组中的第二个元素。

第一个子阵列是[“name”，“Lydia”]，其中x等于name，而y等于Lydia。 第二个子阵列是[“age”，21]，其中x等于age，而y等于21。