

使用 JSFiddle 提升 JavaScript 技能

预计时间：60 分钟

介绍：

本实验旨在通过实际练习提升您使用 JS Fiddle 的 JavaScript 编程技能。重点是解决现实世界中的逻辑问题，鼓励您思考并编写高效代码，并在 JS Fiddle 中执行以查看输出。在本实验结束时，您将提高在不同场景中实施解决方案的能力，并增强您的编码信心。

目标：

- 使用 JavaScript 发展解决问题的能力
- 练习编写和调试逻辑程序
- 理解如何使用循环、函数和条件逻辑实现现实世界的解决方案
- 加强在 JSFiddle 等平台上的编码实践

练习 1：计算总销售额

问题：

您正在为一家在线商店工作。您的任务是编写一个 JavaScript 代码片段，计算一组销售交易的总销售额。

输入细节：

- 一个表示销售交易的对象数组。每个对象具有以下属性：
 - item：产品名称（字符串）
 - quantity：售出单位数（整数）
 - price：单价（浮点数）

输出细节：

- 一个表示总销售额的单一数字

实现步骤：

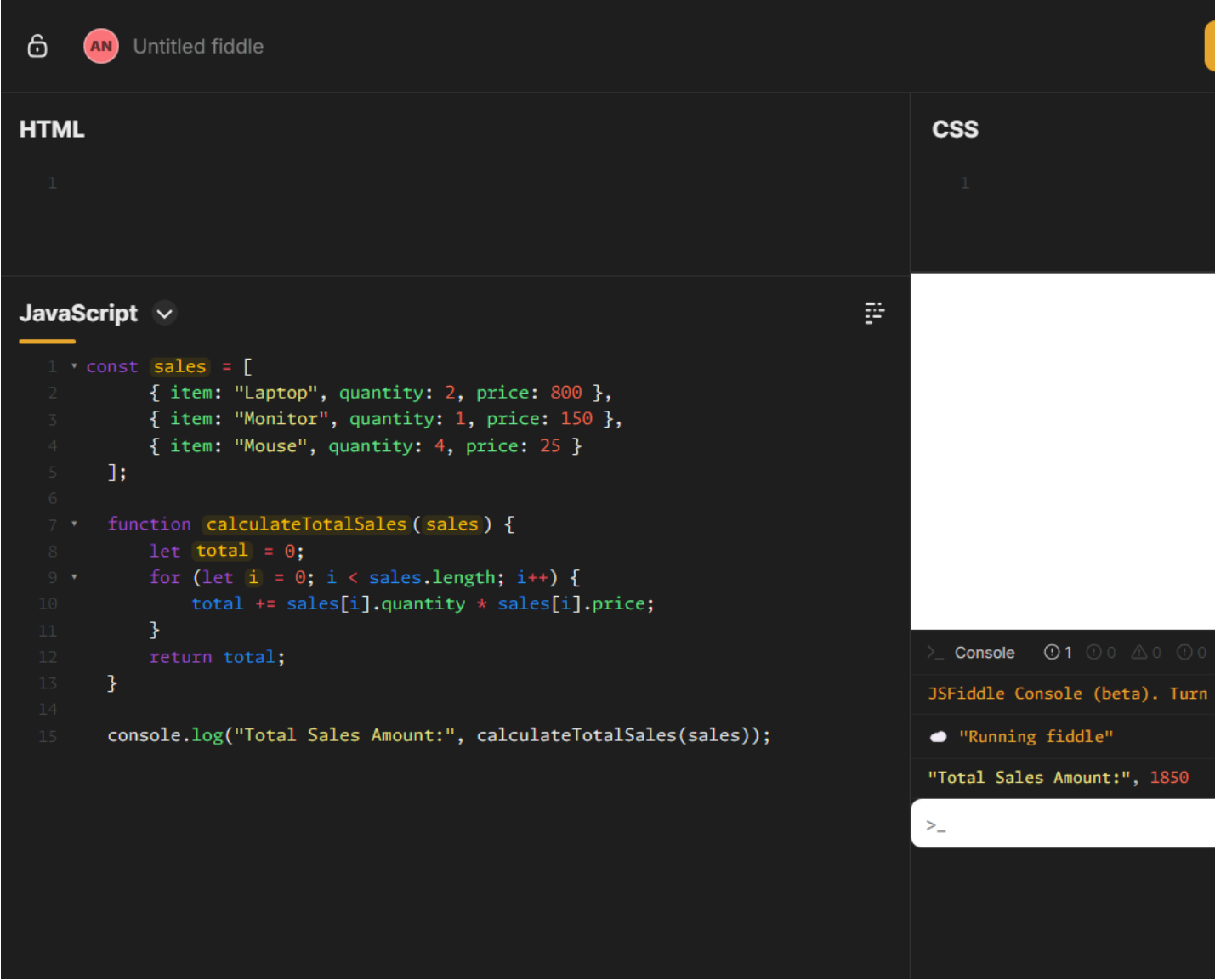
1. 定义一个包含至少 3 个示例对象的销售交易数组
2. 编写一个函数 calculateTotalSales，接受该数组作为输入
3. 使用循环遍历数组并计算总销售额
4. 将总销售额打印到控制台

- ▶ [点击这里查看提示](#)
- ▶ [点击这里查看解决方案代码](#)

在 JSFiddle 上编写程序：

- 访问 [JSFiddle](#)
- 在 JavaScript 部分编写代码
- 点击 Run 按钮执行程序，并在控制台部分查看结果

代码的输出应如下图所示。



练习 2：生成订单收据

问题：
编写一个 JavaScript 程序，为客户的订单生成收据。收据应包括每个项目的名称、数量、价格和总费用。

输入细节：

- 一个表示已订购项目的对象数组。每个对象包含：
 - item: 产品名称（字符串）
 - quantity: 订购数量（整数）
 - price: 每单位价格（浮点数）

输出细节：

- 一份详细的收据，显示每个项目的详细信息和总金额

实现步骤：

1. 定义一个包含至少 3 个示例条目的已订购项目数组
2. 编写一个函数 generateReceipt，将该数组作为输入
3. 使用循环遍历项目，计算每个项目的总金额和总金额
4. 以格式化字符串打印收据

- ▶ [点击这里查看提示](#)
- ▶ [点击这里查看解决方案代码](#)

在 JSFiddle 上编写程序：

- 访问 [JSFiddle](#)
- 在 JavaScript 部分编写代码
- 点击 Run 按钮执行程序，并在控制台部分检查结果

代码的输出应如下图所示。

HTML

1

CSS

1

JavaScript

```
1  const orders = [
2    { item: "Espresso", quantity: 2, price: 3.5 },
3    { item: "Latte", quantity: 3, price: 4.0 },
4    { item: "Cappuccino", quantity: 1, price: 4.5 }
5  ];
6
7  function generateReceipt(orders) {
8    let grandTotal = 0;
9    console.log("Receipt:");
10   console.log("-----");
11   for (let i = 0; i < orders.length; i++) {
12     const itemTotal = orders[i].quantity * orders[i].price;
13     grandTotal += itemTotal;
14     console.log(`${orders[i].item} - Quantity: ${orders[i].quantity}, Price: ${orders[i].price}, Total: ${itemTotal}`);
15   }
16   console.log("-----");
17   console.log(`Grand Total: ${grandTotal}`);
18 }
19
20 generateReceipt(orders);
```

Console

7 0 0 0

"Receipt:"

"-----"

"Espresso - Quantity: 2, Price: 3.5, Total: 7.0"

"Latte - Quantity: 3, Price: 4.0, Total: 12.0"

"Cappuccino - Quantity: 1, Price: 4.5, Total: 4.5"

"-----"

"Grand Total: \$23.5"

>_

练习 3：验证密码

问题：
编写一个 JavaScript 程序来验证密码列表。密码有效的条件是：

- 仅包含字母和数字（字母和数字）
- 长度必须至少为 8 个字符，但不超过 20 个字符

输入细节：

- 一个密码数组（字符串）

输出细节：

- 一条消息，指示每个密码是有效还是无效

实现步骤：

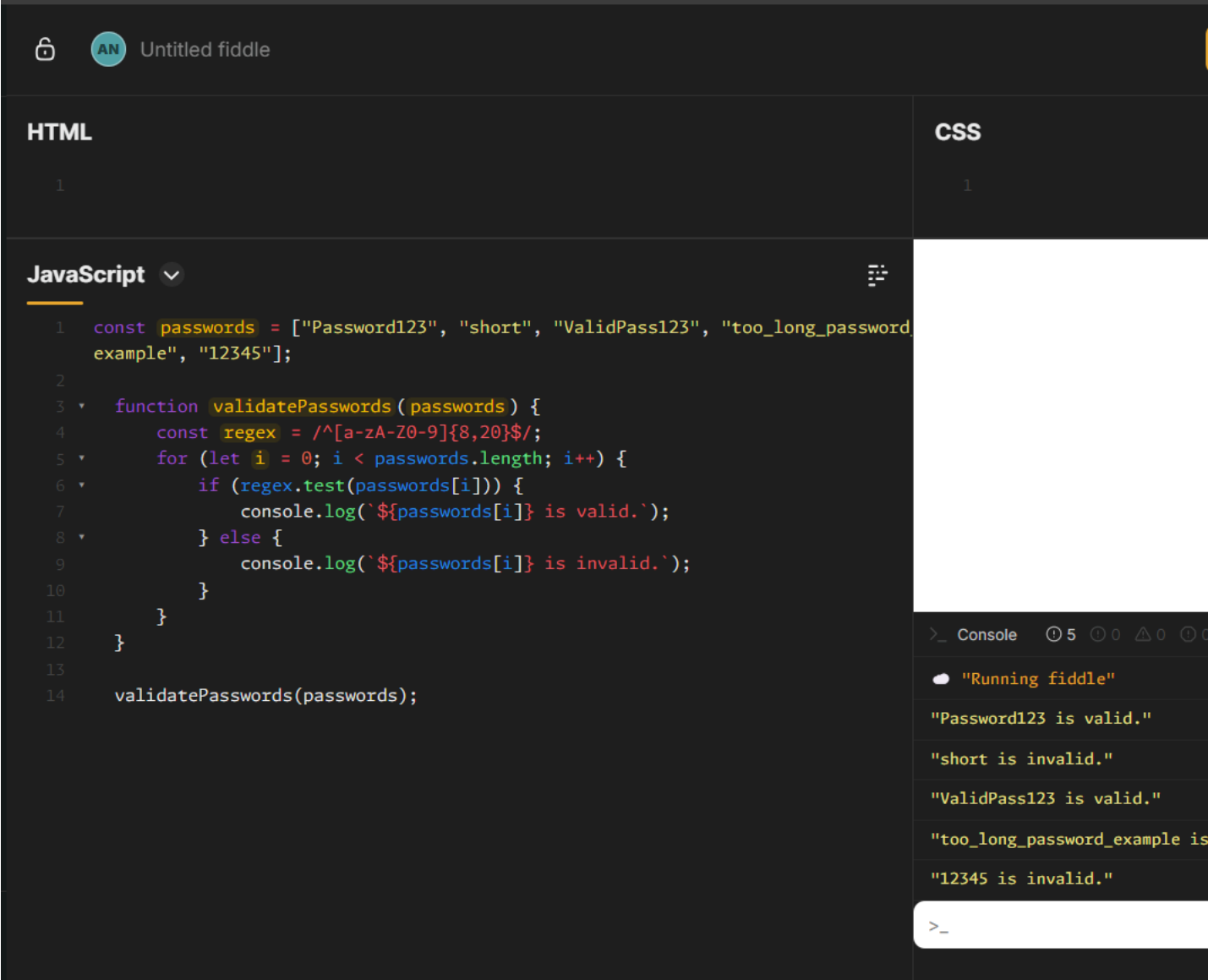
1. 定义一个示例密码数组
2. 编写一个函数 validatePasswords，将该数组作为输入
3. 使用循环遍历密码并根据验证标准检查每个密码
4. 记录每个密码是有效还是无效

- ▶ [点击这里查看提示](#)
- ▶ [点击这里查看解决方案代码](#)

在 JSFiddle 上编写程序：

- 转到 [JSFiddle](#)
- 在 JavaScript 部分编写代码
- 通过点击 Run 按钮执行程序，并在控制台部分检查结果

代码的输出应如下面的截图所示。



练习 4：跟踪产品库存水平

问题：
您正在为一家在线零售公司工作。您的任务是编写一个 JavaScript 程序，跟踪库存中各种产品的库存水平。程序应检查产品是否有库存，并记录适当的消息。

输入细节：

- 一个表示产品的对象数组。每个对象包含：
 - product：产品名称（字符串）
 - stock：可用库存数量（整数）

输出细节：

- 每个产品的消息，指示该产品是“有库存”还是“缺货”。

实现步骤：

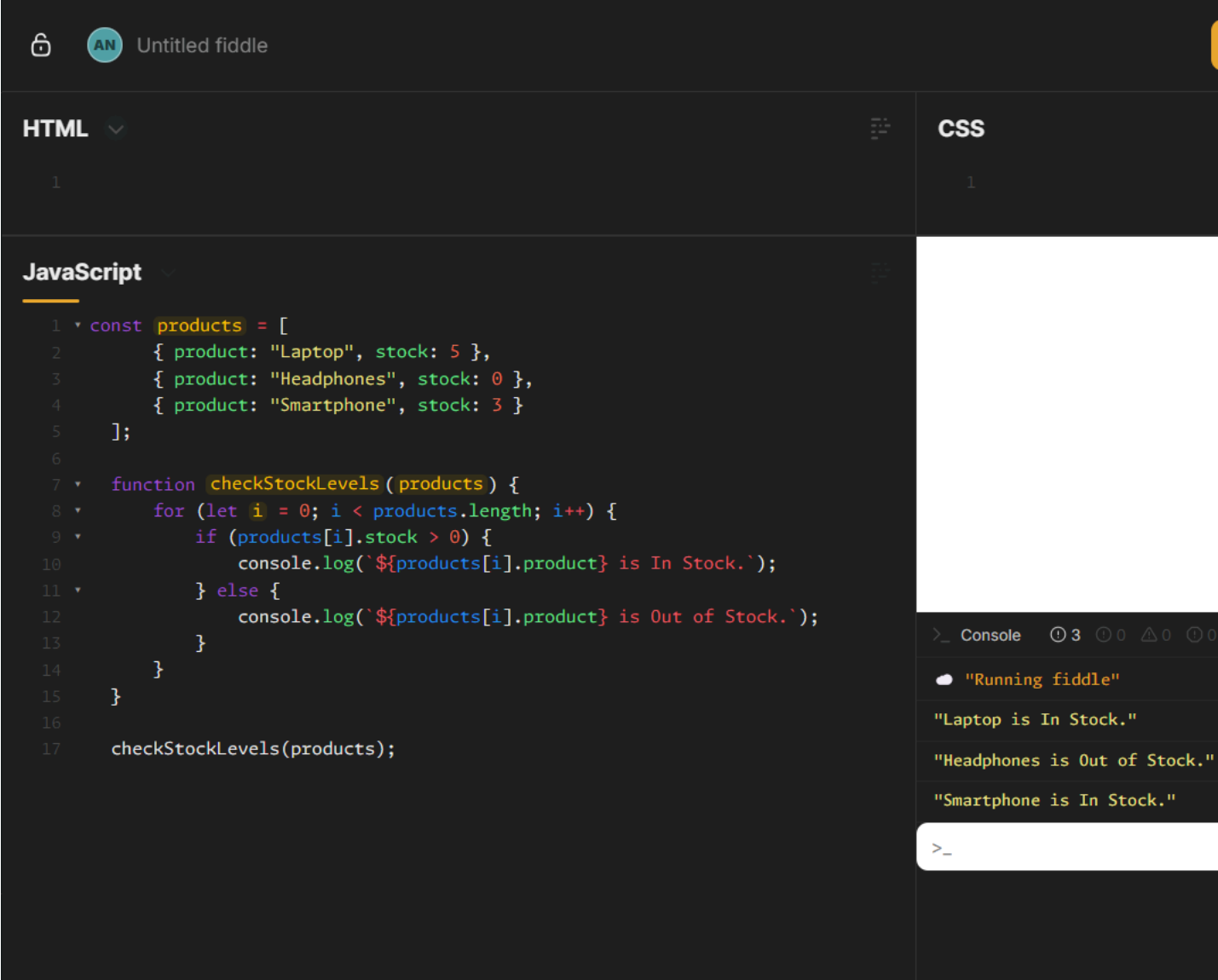
1. 定义一个包含至少 3 个示例产品的产品对象数组
2. 编写一个函数 checkStockLevels，接受该数组作为输入
3. 使用循环遍历数组并检查每个产品的库存水平
4. 打印一条消息，指示产品是“有库存”还是“缺货”

- ▶ [点击此处查看提示](#)
- ▶ [点击此处查看解决代码](#)

在 JSFiddle 上编写程序：

- 访问 [JSFiddle](#)
- 在 JavaScript 部分编写代码
- 点击“运行”按钮执行程序，并在控制台部分检查结果

代码的输出应如下面的截图所示。



结论：

通过这些练习，您已经练习了解决中级难度的问题，使用 JavaScript。每个任务侧重于逻辑思维的不同方面，从验证到字符串操作。继续练习类似的挑战，以进一步提升您的编程技能和信心。

作者

[Rajashree Patil](#)



Skills Network

更新日志

日期	版本	更改者	更改描述
2024-27-12	———	Prashant Juyal	QA 编辑
2024-13-12	1.0	Rajashree Patil	创建初始版本