**JavaScript 中的 setTimeout()**

setTimeout() 是一种在定时器运行完毕后执行一段代码的方法。

这是 setTimeout() 方法的语法。

let timeoutID = setTimeout(function, delay in milliseconds, argument1, argument2,...);

我们来解释一下这个语法。

**函数**

setTimeout() 将设置一个计时器，一旦计时结束，该函数将运行。

**以毫秒为单位的延迟**

在此方法中，你可以指定希望函数延迟多少毫秒。1,000 毫秒等于 1 秒。

在本例中，消息将在 3 秒延迟后出现在屏幕上（3,000 毫秒）。

const para = document.getElementById("para");

function myMessage() {

para.innerHTML = "I just appeared";

console.log("message appeared");

}

setTimeout(myMessage, 3000);

如果 setTimeout() 方法中不存在延迟，则将其设置为零，消息将立即出现。

const para = document.getElementById("para");

function myMessage() {

para.innerHTML = "No delay in this message";

console.log("message appeared immediately");

}

setTimeout(myMessage);

**参数**

你还可以将可选参数传递给函数。

在此示例对话中，Britney 将提出问题，Ashley 的回复将延迟 3 秒。它将包括来自lunchMenu 函数的两个可选参数。

const ashley = document.getElementById("ashley");

function lunchMenu(food1, food2) {

ashley.innerHTML = `<strong>Ashley: </strong>I had ${food1} and ${food2}.`;

}

setTimeout(lunchMenu, 3000, "pizza", "salad");

**timeoutID**

setTimeout() 将返回 timeoutID，它是一个正整数，是计时器的唯一 ID。

**clearTimeout()**

此方法用于取消 setTimeout()。在该方法中，你必须引用 timeoutID。

这是基本语法。

clearTimeout(timeoutID)

在此示例中，消息将在 10 秒（10,000 毫秒）延迟后出现。但是如果用户点击 Stop Timer 按钮，则 setTimeout() 将被取消。

const timerMsg = document.getElementById("message1");

const stopBtn = document.getElementById("stop");

function timerMessage() {

timerMsg.innerHTML = "Thanks for waiting!";

}

let timeoutID = setTimeout(timerMessage, 10000);

stopBtn.addEventListener("click", () => {

clearTimeout(timeoutID);

timerMsg.innerHTML = "Timer was stopped";

});

**你应该为 setTimeout() 传入一个字符串而不是一个函数吗**

传递字符串而不是函数被认为是不好的做法，有安全风险。

避免像这样编写 setTimeout() ：

setTimeout("console.log('Do not do this');", 1000);

一些代码编辑器会警告你，并建议你改用函数