

Web APIs 第一天

Dom获取&属性操作







- 1. 掌握DOM属性操作,完成元素内容设置
- 2. 元素属性设置,控制元素样式





- ◆ Web API 基本认知
- ◆ 获取DOM对象
- ◆ 操作元素内容
- ◆ 操作元素属性
- ◆ 定时器-间歇函数
- ◆ 综合案例





- 变量声明有三个 var let 和 const
- 我们应该用那个呢?
- 首先var 先排除,老派写法,问题很多,可以淘汰掉...
- let or const?
- 建议: const 优先,尽量使用const,原因是:
 - ➤ const 语义化更好
 - ▶ 很多变量我们声明的时候就知道他不会被更改了,那为什么不用 const呢?
 - > 实际开发中也是,比如react框架,基本const
- 如果你还在纠结,那么我建议:
 - 有了变量先给const,如果发现它后面是要被修改的,再改为let



● 请问以下的 可不可以把let 改为 const?

可以, 因为他们值没有变过



● 请问以下的 可不可以把let 改为 const?

不可以, 因为变量进行了重新赋值

```
for (let i = 0; i < nums.length; i++) {
   document.write(nums[i])
}</pre>
```



● 请问以下的 可不可以把let 改为 const?

```
let arr = ['red', 'green']
arr.push('pink')
console.log(arr) // ['red', 'green', 'pink']
```

```
let person = {
  uname: 'pink老师',
  age: 18,
  gender: '女'
}
person.address = '武汉黑马'
console.log(person)
```

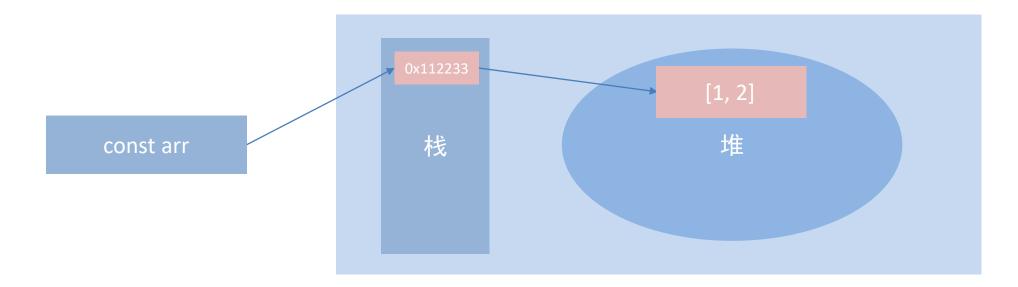
console.log(person)





- const 声明的值不能更改,而且const声明变量的时候需要里面进行初始化
- 但是对于引用数据类型,const声明的变量,里面存的不是值,不是值,不是值,是 地址。

const arr = [1, 2, 3]





● 考考你:可以改为 const 吗?

```
let person = {
  uname: 'pink老师',
  age: 18,
  gender: '女'
}
person.gender = '男'
console.log(person.gender) // 修改了为男
console.log(person)
```

这样改可以了



变量声明

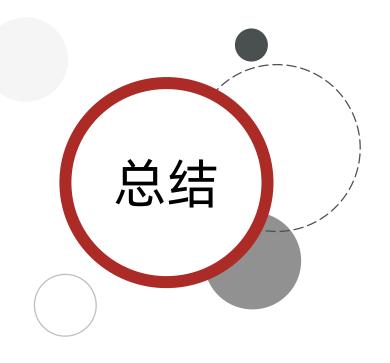
● 考考你:下面写法可以吗?

错误,他们地址不一样

```
const names = []
names = [1, 2, 3]

const obj = {
    obj = {
        uname: 'pink老师'
    }
}
```





- 1. 以后声明变量我们优先使用哪个?
 - > const
 - ➤ 有了变量先给const,如果发现它后面是要被修改的,再改为let
- 2. 为什么const声明的对象可以修改里面的属性?
 - ▶ 因为对象是引用类型,里面存储的是地址,只要地址不变,就不会报错
 - ▶ 建议数组和对象使用 const 来声明
- 3. 什么时候使用let声明变量?
 - ➤ 如果基本数据类型的值或者引用类型的地址发生变化的时候,需要用let
 - ▶ 比如 一个变量进行加减运算, 比如 for循环中的 i++





Web API 基本认知

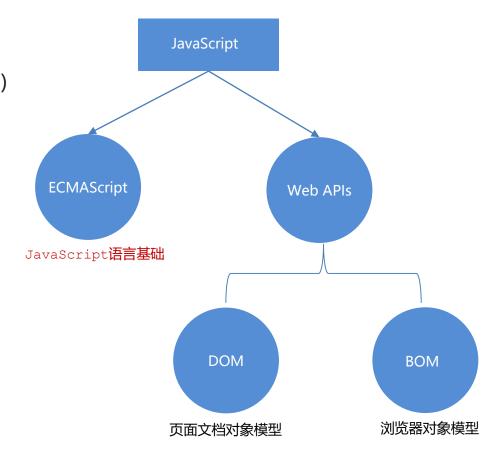
- 作用和分类
- 什么是DOM
- DOM树
- DOM对象



1.1 作用和分类

● 作用: 就是使用 JS 去操作 html 和浏览器

● 分类: DOM (文档对象模型)、BOM (浏览器对象模型)







Web API 基本认知

- 作用和分类
- · 什么是DOM
- DOM树
- DOM对象

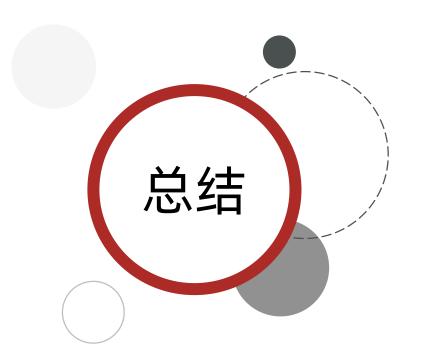


1.2 什么是DOM

- DOM (Document Object Model——文档对象模型)是用来呈现以及与任意 HTML 或 XML文档交互的API
- 白话文: DOM是浏览器提供的一套专门用来 操作网页内容 的功能
- DOM作用
 - 开发网页内容特效和实现用户交互







- 1. Web API阶段我们学习那两部分?
 - > DOM
 - **≻** BOM
- 2. DOM 是什么? 有什么作用?
 - ➤ DOM 是文档对象模型
 - ▶ 操作网页内容,可以开发网页内容特效和实现用户交 互





Web API 基本认知

- 作用和分类
- 什么是DOM
- DOM树
- DOM对象



1.3 DOM树

- DOM树是什么
 - ➤ 将 HTML 文档以树状结构直观的表现出来, 我们称之为文档树或 DOM 树
 - ▶ 描述网页内容关系的名词
 - ▶ 作用: 文档树直观的体现了标签与标签之间的关系

```
document
   DOM
(文档对象模型)
                                                                 元素节点
                                                                 属性节点
                          HTML
                                                                 文本节点
                head
                                       body
                        更多...
                                                            更多...
        meta
                                            链接名
                                                                文本
                                                          class
       content
```





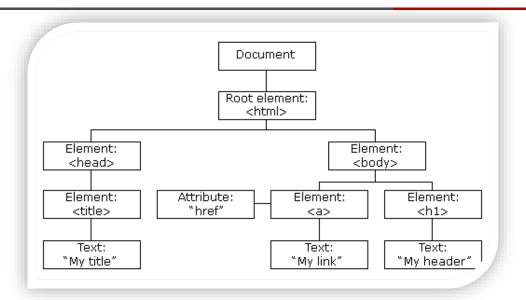
Web API 基本认知

- 作用和分类
- 什么是DOM
- DOM树
- DOM对象



1.4 DOM对象(重要)

- DOM对象:浏览器根据html标签生成的 JS对象
 - ▶ 所有的标签属性都可以在这个对象上面找到
 - ▶ 修改这个对象的属性会自动映射到标签身上
- DOM的核心思想
 - ▶ 把网页内容当做对象来处理
- document 对象
 - ▶ 是 DOM 里提供的一个对象
 - ▶ 所以它提供的属性和方法都是用来访问和操作网页内容的
 - ✓ 例: document.write()
 - ➢ 网页所有内容都在document里面









1. DOM 树是什么?

- ▶ 将 HTML 文档以树状结构直观的表现出来,我们称之为文档树或 DOM 树
- ▶ 作用: 文档树直观的体现了标签与标签之间的关系
- 2. DOM对象怎么创建的?
 - ➤ 浏览器根据html标签生成的 JS对象 (DOM对象)
 - ➤ DOM的核心就是把内容当对象来处理
- 3. document 是什么?
 - ➤ 是 DOM 里提供的一个对象
 - 网页所有内容都在document里面





- ◆ Web API 基本认知
- ◆ 获取DOM对象
- ◆ 操作元素内容
- ◆ 操作元素属性
- ◆ 定时器-间歇函数
- ◆ 综合案例





获取DOM元素

- · 根据CSS选择器来获取DOM元素 (重点)
- 其他获取DOM元素方法(了解)



二、获取DOM对象

目标:能查找/获取DOM对象

提问:我们想要操作某个标签首先做什么?

▶ 肯定首先选中这个标签,跟 CSS选择器类似,选中标签才能操作

● 查找元素DOM元素就是利用 JS 选择页面中标签元素

学习路径:

1. 根据CSS选择器来获取DOM元素 (重点)

2. 其他获取DOM元素方法(了解)



2.1 根据CSS选择器来获取DOM元素 (重点)

1. 选择匹配的第一个元素

语法:

document.querySelector('css选择器')

参数:

包含一个或多个有效的CSS选择器 字符串

返回值:

CSS选择器匹配的第一个元素,一个 HTMLElement对象。

如果没有匹配到,则返回null。

多参看文档: https://developer.mozilla.org/zh-CN/docs/Web/API/Document/querySelector



2.1 根据CSS选择器来获取DOM元素 (重点)

2. 选择匹配的多个元素

语法:

document.querySelectorAll('css选择器')

参数:

包含一个或多个有效的CSS选择器 字符串

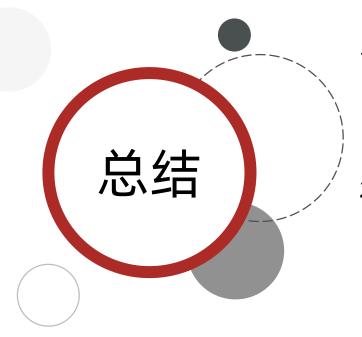
返回值:

CSS选择器匹配的NodeList 对象集合

例如:

document.querySelectorAll('ul li')





- 1. 获取一个DOM元素我们使用谁?能直接操作修改吗?
 - querySelector()
 - ▶ 可以
- 2. 获取多个DOM元素我们使用谁?能直接修改吗?如果不能可以怎么做到修改?
 - > querySelectorAll()
 - ▶ 不可以, 只能通过遍历的方式一次给里面的元素做修改



2.1 根据CSS选择器来获取DOM元素 (重点)

document.querySelectorAll('css选择器')

得到的是一个伪数组:

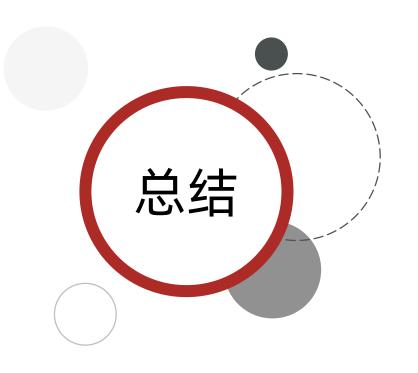
- > 有长度有索引号的数组
- ▶ 但是没有 pop() push() 等数组方法

想要得到里面的每一个对象,则需要遍历(for)的方式获得。

注意事项

哪怕只有一个元素,通过querySelectAll() 获取过来的也是一个伪数组,里面只有一个元素而已





- 1. 获取页面中的标签我们最终常用那两种方式?
 - > querySelectorAll()
 - > querySelector()
- 2. 他们两者的区别是什么?
 - ➤ querySelector() 只能选择一个元素, 可以直接操作
 - querySelectorAll()可以选择多个元素,得到的是伪数组,需要遍历得到每一个元素
- 3. 他们两者小括号里面的参数有神马注意事项?
 - ➤ 里面写css选择器
 - > 必须是字符串,也就是必须加引号





· 请控制台依次输出 3个 li 的 DOM对象

```
    %
    %
    %
```



2.2 其他获取DOM元素方法(了解)

```
|/ 根据id获取一个元素
| document.getElementById('nav')
| 根据 标签获取一类元素 获取页面 所有div
| document.getElementsByTagName('div')
| 根据 类名获取元素 获取页面 所有类名为 w的
| document.getElementsByClassName('w')
```





- ◆ Web API 基本认知
- ◆ 获取DOM对象
- ◆ 操作元素内容
- ◆ 操作元素属性
- ◆ 定时器-间歇函数
- ◆ 综合案例





操作元素内容

- 对象.innerText 属性
- 对象.innerHTML 属性



三、操作元素内容

目标: 能够修改元素的文本更换内容

- DOM对象都是根据标签生成的,所以操作标签,本质上就是操作DOM对象。
- 就是操作对象使用的点语法。
- 如果想要修改标签元素的里面的内容,则可以使用如下几种方式:
- 学习路径:
- 1. 对象.innerText 属性
- 2. 对象.innerHTML 属性





三、操作元素内容

1. 元素innerText 属性

- ▶ 将文本内容添加/更新到任意标签位置
- ▶ 显示纯文本,不解析标签

举例说明

```
const info = document.querySelector('.info')

// 获取标签内部的文字

// console.log(info.innerText)

// 添加/修改标签内部文字内容

info.innerText = '嗨喽,俺是刘德华~'
```



三、操作元素内容

2. 元素.innerHTML 属性

- 将文本内容添加/更新到任意标签位置
- > 会解析标签,多标签建议使用模板字符

举例说明

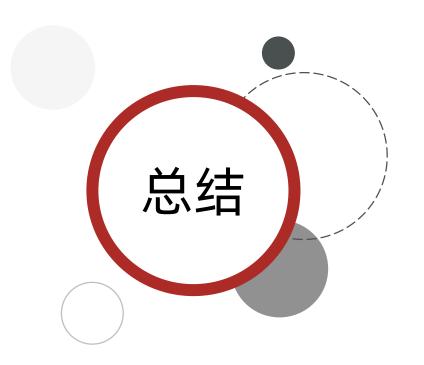
```
const info = document.querySelector('.info')

// 获取标签内部的文字

// console.log(info.innerHTML)

// 添加/修改标签内部文字内容
info.innerHTML = `嗨喽,俺是<strong>刘德华</strong>~ `
```





- 1. 设置/修改DOM元素内容有哪2钟方式?
 - ➤ 元素.innerText 属性
 - ▶ 元素.innerHTML 属性
- 2. 三者的区别是什么?
 - ➤ 元素.innerText 属性 只识别文本,不能解析标签
 - ➤ 元素.innerHTML 属性 能识别文本,能够解析标签
 - ▶ 如果还在纠结到底用谁,你可以选择innerHTML



富案例

年会抽奖案例

需求: 从数组随机抽取一等奖、二等奖和三等奖, 显示到对应的标签里面。

素材:

➤ html文件结构

▶ 数组名单 '周杰伦', '刘德华', '周星驰', 'Pink老师', '张学友'





富案例

年会抽奖案例

需求: 从数组随机抽取一等奖、二等奖和三等奖, 显示到对应的标签里面。

分析:

①: 声明数组: const personArr = ['周杰伦', '刘德华', '周星驰', 'Pink老师', '张学友']

②:一等奖:随机生成一个数字(0~数组长度),找到对应数组的名字

③:通过innerText 或者 innerHTML 将名字写入span元素内部

④: 二等奖依次类推







- ◆ Web API 基本认知
- ◆ 获取DOM对象
- ◆ 操作元素内容
- ◆ 操作元素属性
- ◆ 定时器-间歇函数
- ◆ 综合案例





操作元素属性

- 操作元素**常用**属性
- 操作元素**样式**属性
- 操作 **表单元素** 属性
- 自定义属性



4.1 操作元素常用属性

- 还可以通过 JS 设置/修改标签元素属性, 比如通过 src更换 图片
- 最常见的属性比如: href、title、src 等
- 语法:

```
对象.属性 = 值
```

举例说明

```
// 1. 获取元素
const pic = document.querySelector('img')
// 2. 操作元素
pic.src = './images/b02.jpg'
pic.title = '刘德华黑马演唱会'
```





· 页面刷新,图片随机更换

需求: 当我们刷新页面, 页面中的图片随机显示不同的图片





1 练习

• 页面刷新,图片随机更换

需求: 当我们刷新页面, 页面中的图片随机显示不同的图片

分析:

①: 随机显示,则需要用到随机函数

②: 更换图片需要用到图片的 src 属性, 进行修改

③:核心思路:

- 1. 获取图片元素
- 2. 随机得到图片序号
- 3. 图片.src = 图片随机路径





操作元素属性

- 操作元素**常用**属性
- 操作元素**样式**属性
- 操作 **表单元素** 属性
- 自定义属性



4.2 操作元素样式属性

- 还可以通过 JS 设置/修改标签元素的样式属性。
- 比如通过 轮播图小圆点自动更换颜色样式
- ▶ 点击按钮可以滚动图片,这是移动的图片的位置 left 等等

学习路径:

- 1. 通过 style 属性操作CSS
- 2. 操作类名(className) 操作CSS
- 3. 通过 classList 操作类控制CSS





4.2 操作元素样式属性

- 1.通过 style 属性操作CSS
- 语法:

对象.style.样式属性 = 值

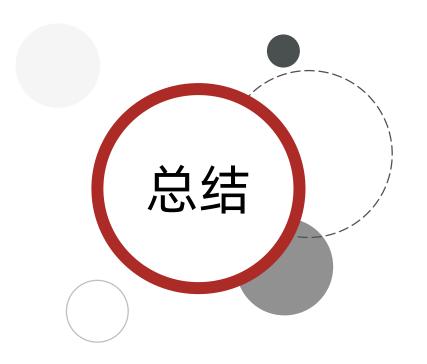
举例说明

```
const box = document.querySelector('.box')
// 修改元素样式
box.style.width = '200px'
box.style.marginTop = '15px'
box.style.backgroundColor = 'pink'
```

注意:

- 1. 修改样式通过style属性引出
- 2. 如果属性有-连接符,需要转换为小驼峰命名法
- 3. 赋值的时候,需要的时候不要忘记加CSS单位





- 1. 设置/修改元素样式属性通过 style 属性引出来?
- 2. 如果需要修改一个div盒子的样式,比如 padding-left, 如何写?
 - element.style.paddingLeft = '300px'
 - > 小驼峰命名法
- 3. 因为我们是样式属性,一定别忘记,大部分数字后面都需要加单位

```
const box = document.querySelector('.box')
// 修改元素样式
box.style.width = '200px'
box.style.marginTop = '15px'
box.style.backgroundColor = 'pink'
```





• 页面刷新,页面随机更换背景图片

需求: 当我们刷新页面, 页面中的背景图片随机显示不同的图片

分析:

①: 随机函数

②: css页面背景图片 background-image

③:标签选择body, 因为body是唯一的标签,可以直接写 document.body.style



4.2 操作元素**样式**属性

- 2. 操作类名(className) 操作CSS
- 如果修改的样式比较多,直接通过style属性修改比较繁琐,我们可以通过借助于css类名的形式。
- 语法:

// active 是一个css类名 元素.className = 'active'

- 注意:
- 1. 由于class是关键字, 所以使用className去代替
- 2. className是使用新值换旧值,如果需要添加一个类,需要保留之前的类名





- 1. 使用 className 有什么好处?
 - ▶ 可以同时修改多个样式
- 2. 使用 className 有什么注意事项?
 - ▶ 直接使用 className 赋值会覆盖以前的类名



4.2 操作元素样式属性

- 3. 通过 classList 操作类控制CSS
- 为了解决className 容易覆盖以前的类名,我们可以通过classList方式追加和删除类名
- 语法:

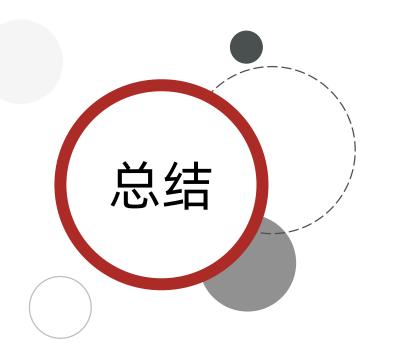
```
工業'ClassList.toggle('菜名')

// 删除一个类

元素.classList.remove('类名')

// 切换一个类

元素.classList.toggle('类名')
```





- ▶ 修改大量样式的更方便
- ▶ 修改不多样式的时候方便
- > classList 是追加和删除不影响以前类名







• 轮播图随机版

需求: 当我们刷新页面, 页面中的轮播图会显示不同图片以及样式

模块:

①:图片会随机变换

②:底部盒子背景颜色和文字内容会变换

③: 小圆点随机一个高亮显示





1 练习

• 轮播图随机版

需求: 当我们刷新页面, 页面中的轮播图会显示不同图片以及样式

分析:

①: 准备一个数组对象,里面包含详细信息(素材包含)

②: 随机选择一个数字,选出数组对应的对象,更换图片,底部盒子背景颜色,以及文字内容

③:利用这个随机数字,让小圆点添加高亮的类(addClass) 利用css 结构伪类选择器

```
// 1. 初始数据
const sliderData = [
    { url: './images/slider01. jpg', title: '对人类来说会不会太超前了?', color: 'rgb(100, 67, 68)' },
    { url: './images/slider02. jpg', title: '开启剑与雪的黑暗传说!', color: 'rgb(43, 35, 26)' },
    { url: './images/slider03. jpg', title: '真正的jo厨出现了!', color: 'rgb(36, 31, 33)' },
    { url: './images/slider04. jpg', title: '李玉刚: 让世界通过B站看到东方大国文化', color: 'rgb(139, 98, 66)' },
    { url: './images/slider05. jpg', title: '快来分享你的寒假日常吧~', color: 'rgb(67, 90, 92)' },
    { url: './images/slider06. jpg', title: '哔哩哔哩小年YEAH', color: 'rgb(166, 131, 143)' },
    { url: './images/slider07. jpg', title: '一站式解决你的电脑配置问题!!!', color: 'rgb(53, 29, 25)' },
    { url: './images/slider08. jpg', title: '谁不想和小猫咪贴贴呢!', color: 'rgb(99, 72, 114)' },
```





操作元素属性

- 操作元素**常用**属性
- 操作元素**样式**属性
- 操作 **表单元素** 属性
- 自定义属性



4.3 操作表单元素 属性

● 表单很多情况,也需要修改属性,比如点击眼睛,可以看到密码,本质是把表单类型转换为文本框

● 正常的有属性有取值的 跟其他的标签属性没有任何区别

▶ 获取: DOM对象.属性名

▶ 设置: DOM对象.属性名 = 新值

表单.value = '用户名' 表单.type = 'password'



■ 全选	商品	商家	价格
	小米手机	小米	¥ 1999
	小米净水器	小米	¥4999
	小米电视	小米	¥ 5999



4.3 操作表单元素 属性

- 表单属性中添加就有效果,移除就没有效果,一律使用布尔值表示 如果为true 代表添加了该属性 如果是false 代表移 除了该属性
- 比如: disabled、checked、selected

■ 全选	商品	商家	价格
	小米手机	小米	¥ 1999
	小米净水器	小米	¥4999
	小米电视	小米	¥ 5999





操作元素属性

- 操作元素**常用**属性
- 操作元素**样式**属性
- 操作 表单元素 属性
- 自定义属性



4.4 自定义属性

- **标准属性:** 标签天生自带的属性 比如class id title等, 可以直接使用点语法操作比如: disabled、checked、selected
- 自定义属性:
 - ➤ 在html5中推出来了专门的data-自定义属性
 - ➤ 在标签上一律以data-开头
 - ➤ 在DOM对象上一律以dataset对象方式获取





- ◆ Web API 基本认知
- ◆ 获取DOM对象
- ◆ 操作元素内容
- ◆ 操作元素属性
- ◆ 定时器-间歇函数
- ◆ 综合案例





定时器-间歇函数

- 定时器函数介绍
- 定时器函数基本使用



五、定时器-间歇函数

目标: 能够说出定时器函数在开发中的使用场景

■ 网页中经常会需要一种功能:每隔一段时间需要自动执行一段代码,不需要我们手动去触发

● 例如:网页中的倒计时

● 要实现这种需求,需要定时器函数

● 定时器函数有两种,今天我先讲间歇函数

用户协议

欢迎申请使用传智公司及其他合作运营主体(下列简称为"传智公司"或"传智")提供的服务。请您(下列统称为"用户")仔细阅读以下全部内容(特别是粗体下划线标注的内容)。

如用户不同意本服务条款任意内容,请勿注册或使用传智服务。如用户通过进入注册程序并勾选"我同意传智邮箱帐号服务条款",即表示用户与传智公司已达成协议,自愿接受本服务条款的所有内容。此后,用户不得

已阅读用户协议(60)







定时器-间歇函数

- 定时器函数介绍
- 定时器函数基本使用



五、定时器-间歇函数

目标: 能够使用定时器函数重复执行代码

定时器函数可以开启和关闭定时器

1. 开启定时器

setInterval(函数,间隔时间)

- ▶ 作用:每隔一段时间调用这个函数
- ▶ 间隔时间单位是毫秒

举例说明

```
function repeat() {
    console.log('前端程序员,就是头发多咋滴~~')
}
// 每隔一秒调用repeat函数
setInterval(repeat, 1000)
```

注意:

- 1. 函数名字**不需要加括号**
- 2. 定时器返回的是一个id数字



五、定时器-间歇函数

定时器函数可以开启和关闭定时器

2. 关闭定时器

```
let 变量名 = setInterval(函数, 间隔时间) clearInterval(变量名)
```

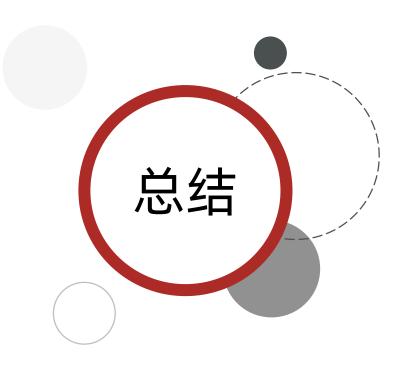
一般不会刚创建就停止,而是满足一定条件再停止

```
let timer = setInterval(function() {
   console.log('hi~~')
}, 1000)
clearInterval(timer)
```

注意:

- 1. 函数名字不需要加括号
- 2. 定时器返回的是一个id数字





- 1. 定时器函数有什么作用?
 - ▶可以根据时间自动重复执行某些代码
- 2. 定时器函数如何开启?
 - ➤ setInterval(函数名, 时间)
- 3. 定时器函数如何关闭?

let 变量名 = setInterval(函数,间隔时间) clearInterval(变量名)



a 案例

阅读注册协议

需求:按钮60秒之后才可以使用

分析:

①:开始先把按钮禁用(disabled 属性)

②:一定要获取元素

③:函数内处理逻辑

秒数开始减减

按钮里面的文字跟着一起变化

如果秒数等于0 停止定时器 里面文字变为 同意 最后 按钮可以点击

用户协议

欢迎申请使用传智公司及其他合作运营主体(下列简称为"传智公司"或"传智")提供的服务。请您(下列统称为"用户")仔细阅读以下全部内容(特别是粗体下划线标注的内容)。

如用户不同意本服务条款任意内容,请勿注册或使用传智服务。如用户通过进入注册程序并勾选"我同意传智邮箱帐号服务条款",即表示用户与传智公司已达成协议,自愿接受本服务条款的所有内容。此后,用户不得

已阅读用户协议(60)





- ◆ Web API 基本认知
- ◆ 获取DOM对象
- ◆ 操作元素内容
- ◆ 操作元素属性
- ◆ 定时器-间歇函数
- ◆ 综合案例



1 案例

轮播图定时器版

需求:每隔一秒钟切换一个图片

分析:

①:准备一个数组对象,里面包含详细信息(素材包含)

②: 获取元素

③:设置定时器函数

设置一个变量++

找到变量对应的对象

更改图片、文字信息

激活小圆点:移除上一个高亮的类名,当前变量对应的小圆点添加类

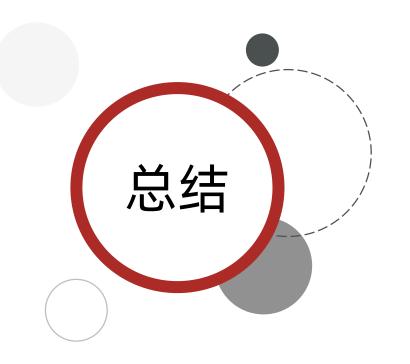
④:处理图片自动复原从头播放(放到变量++后面,紧挨)

如果图片播放到最后一张,就是大于等于数组的长度

则把变量重置为0







- 1. 整理今天笔记, Xmind或者md笔记都可以
- 2. 练习用户倒计时协议案例
- 3. 重点练习自动播放轮播图案例,至少写三遍
- 4. 开始做测试题: PC端: https://ks.wjx.top/vj/hwyC7n1.aspx
- 5. 最后作业
- 6. 预习



你生活的起点并不是那么重要,重要的是最后你能到达哪里



传智教育旗下高端IT教育品牌