链滴

保存网页内容时 自动删除页面头尾广告

作者：mutousoft
原文链接：https://ld246.com/article/1672492273849
来源网站：链滴

网页可以非常方便的为我们展示各种信息，如果遇到重要的资料文献，希望在本地电脑上保存下来该么操作呢？把网址添加到收藏夹，下次直接打开网址查看，但如果资源被网站删除，就再也找不到了还是保存在自己电脑里比较放心，那就使用浏览器的保存网页吧，如果保存为单个文件，则只有文字容，图片丢失了。如果保存所有内容，将产生一个网页文件和一个资源文件夹，包括图片在内的文件保存在这个文件夹中，由于文件较多不容易归类保存和传输。使用保存网页的方式，除正文外，还会存网页标题导航栏、信息侧边栏、底部联系信息等无用的内容。

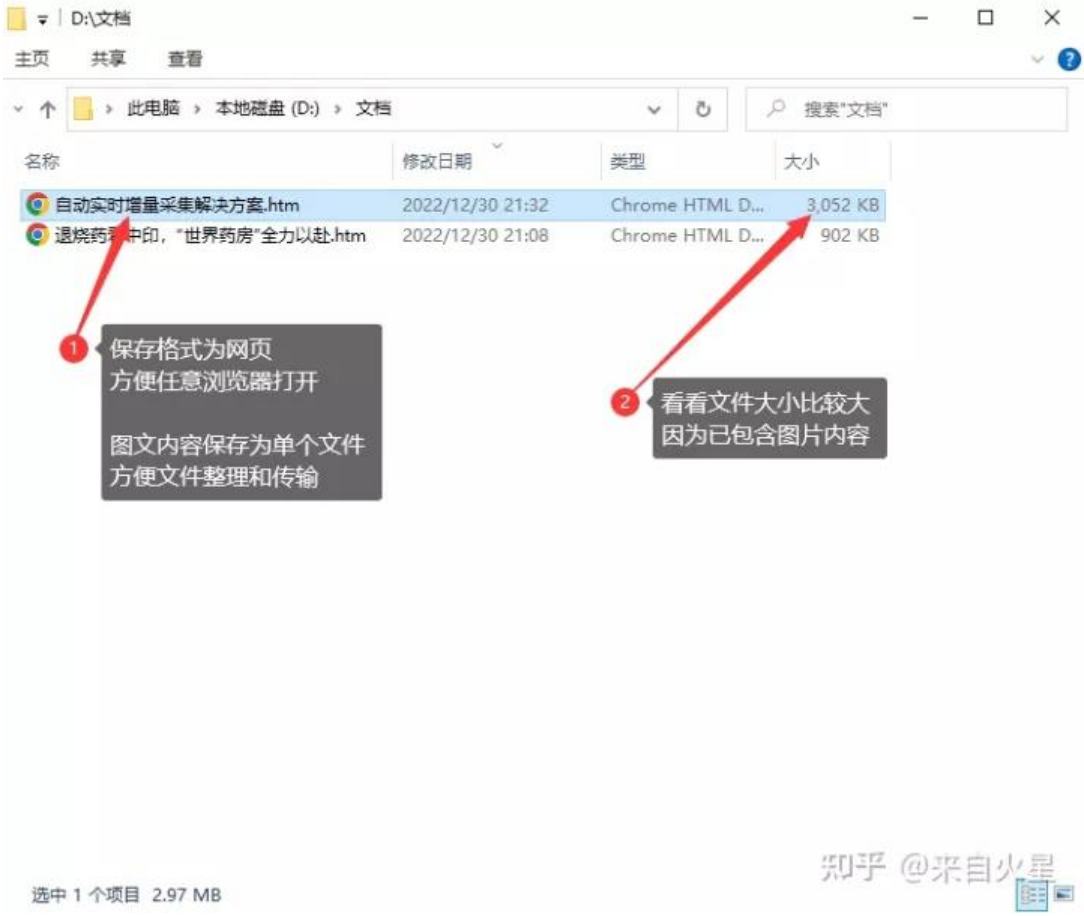
需要保存的网页标题和正文

有没有一种方法，保存网页时，自动智能识别内容标题和正文，且仅保存标题和包括图片在内的正文容，自动删除网页无效的头尾和侧边内容，更要过滤网页上的广告。这就是“AI保存网页“，如下图示，打开任意新闻、公告或文章页面，再点击”AI保存网页“，就可以一键保存网页标题和正文。

原文链接：保存网页内容时 自动删除页面头尾广告

网页保存后与原页面对比

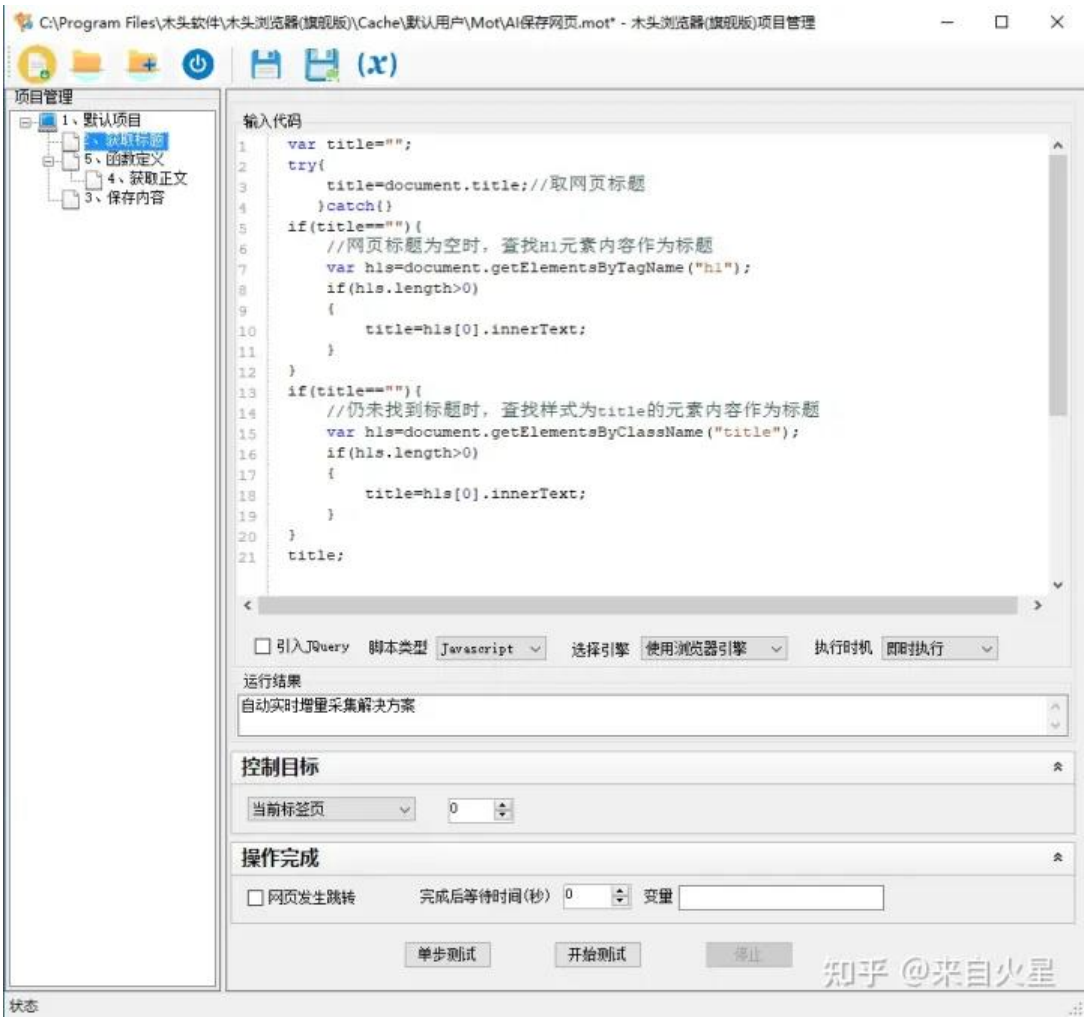
找到文档保存路径，可以看到以文章标题命名的网页文件。这个文档比较大，是因为同时保存和正文的图片，也就是说把文字和图片都保存在单个文档中的。且为htm网页格式，可以使用任意浏览器打。把图片保存在htm网页代码中，是什么原理呢？原来木头浏览器在保存网页时，自动把网页上的图转换成Base64编码，这样就可以在单个文件中保存图片了。



图片和文字内容保存在一个网页文件中

那么是怎样智能识别文章标题和正文的呢？有js基础的小伙伴可以继续往下看。

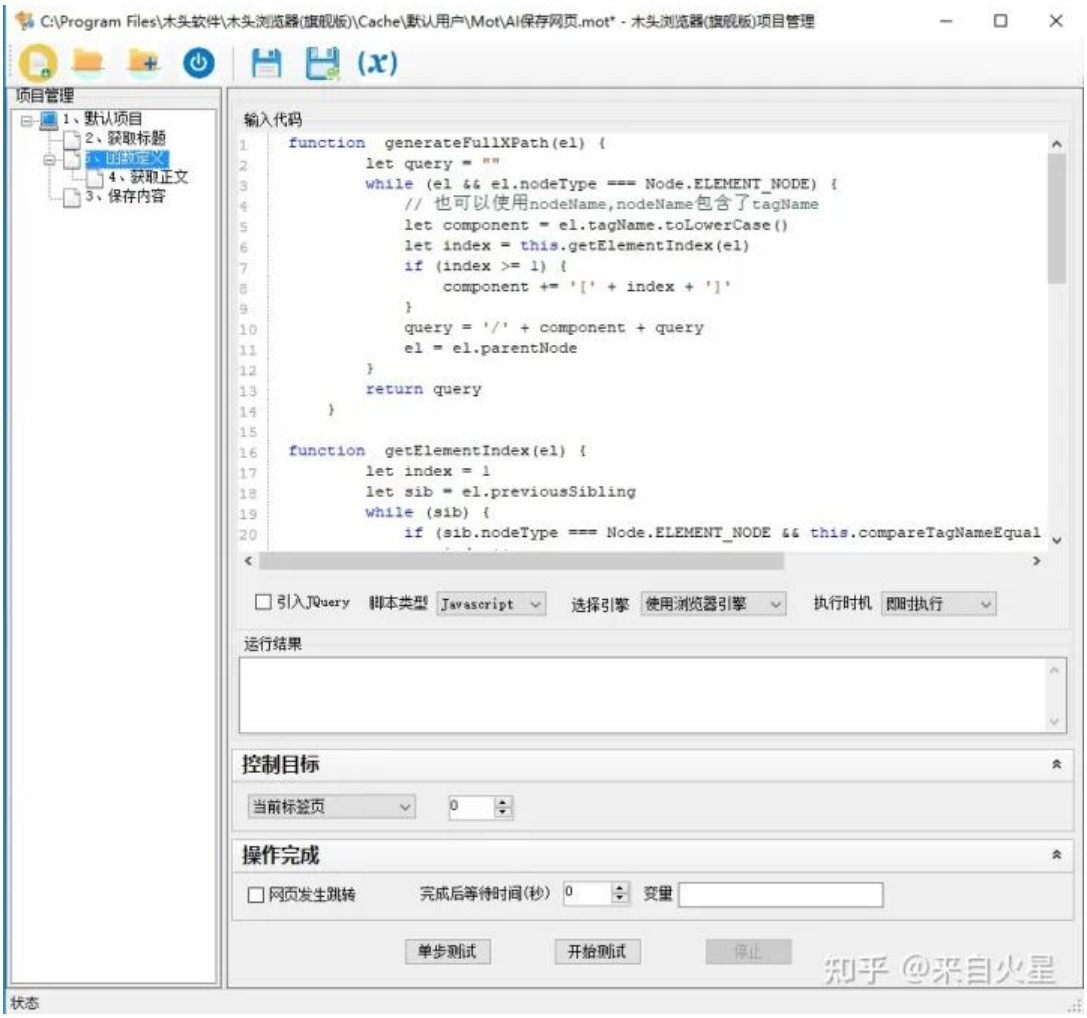
首先在项目管理器中，创建一个脚本代码步骤，通过执行一段js脚本代码找出文章标题。首先获取网头部的title标签作为标题，其次是查找H1元素内容作为标题，如果还是找不到，则查找样式为title的素内容作为标题。



智能识别网页标题

```
var title="";
try{
   title=document.title;//取网页标题
  }catch{}
if(title==""){
   //网页标题为空时，查找H1元素内容作为标题
   var h1s=document.getElementsByTagName("h1");
   if(h1s.length>0)
   {
      title=h1s[0].innerText;
   }
}
if(title==""){
   //仍未找到标题时，查找样式为title的元素内容作为标题
   var h1s=document.getElementsByClassName("title");
   if(h1s.length>0)
   {
      title=h1s[0].innerText;
   }
}
title;
```

再创建一个脚本代码步骤，定义几个重复使用的函数。

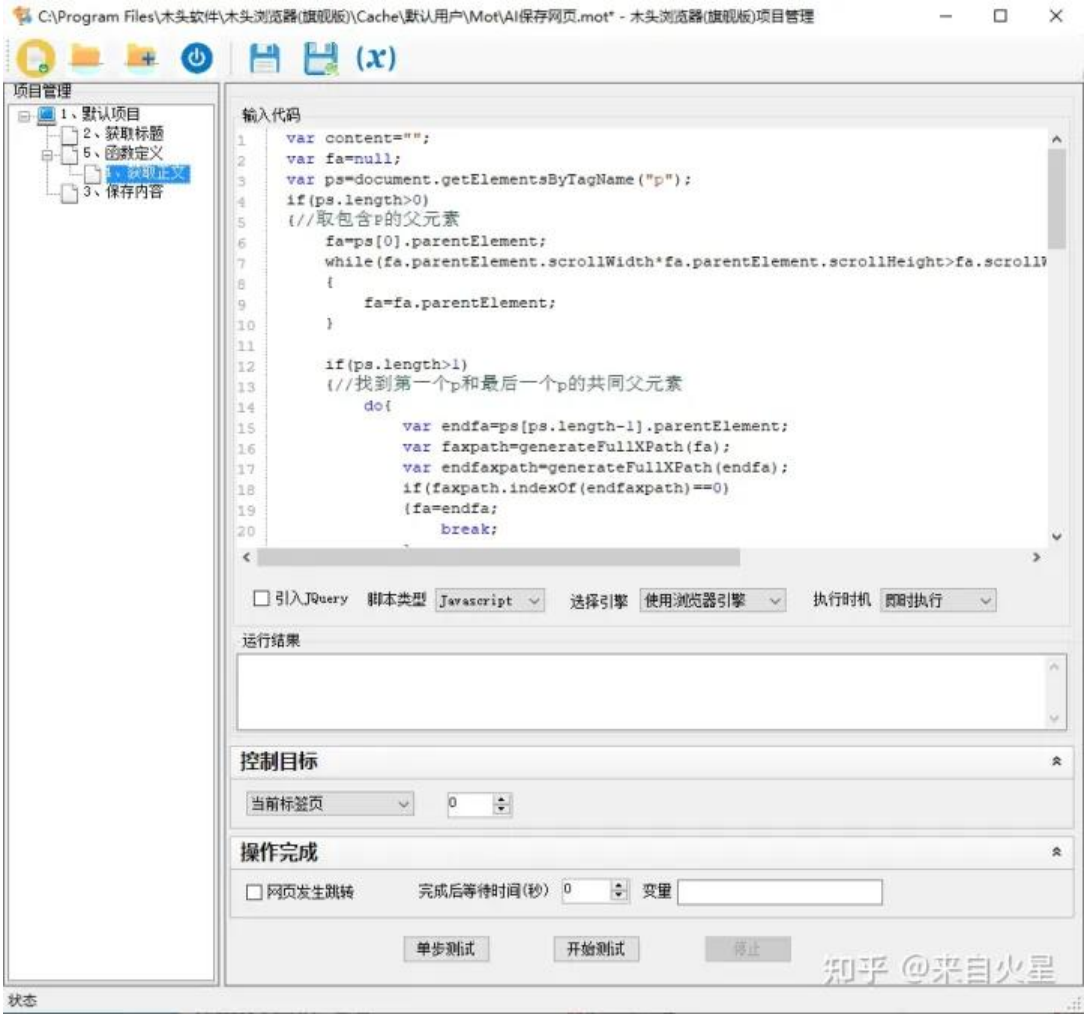



```
function  generateFullXPath(el) {
        let query = ""
        while (el && el.nodeType === Node.ELEMENT_NODE) {
            // 也可以使用nodeName,nodeName包含了tagName
            let component = el.tagName.toLowerCase()
            let index = this.getElementIndex(el)
            if (index >= 1) {
                component += '[' + index + ']'
            }
            query = '/' + component + query
            el = el.parentNode
        }
        return query
    }

function  getElementIndex(el) {
        let index = 1
        let sib = el.previousSibling
        while (sib) {
            if (sib.nodeType === Node.ELEMENT_NODE && this.compareTagNameEqual(el, sib)) {
                index++
            }
            sib = sib.previousSibling
        }
```

原文链接：保存网页内容时 自动删除页面头尾广告

```
    if (index > 1) return index
    sib = el.nextSibling
    while (sib) {
        if (sib.nodeType === Node.ELEMENT_NODE && this.compareTagNameEqual(el, sib)) {
            return 1
        }
        sib = sib.nextSibling
    }
    return 0;
};


    /**
     * 查看两个元素节点名称是否相同
     */
function compareTagNameEqual(primaryEl, siblingEl) {
    let p = primaryEl, s = siblingEl
    // return (p.tagName === s.tagName && (!p.id || p.id === s.id));
    return (p.tagName === s.tagName)
};
```

同样使用js代码智能识别查找正文元素，一般文章正文部分由多个段落组成，所以我们从P元素入手，到子元素包含大量P元素的元素，就是正文元素了。如果没有P元素，则获取页面中间位置，面积较大元素作为正文元素，并给正文元素设置一个id值”mutoubrowser"作为标记。方便后续步骤调用。

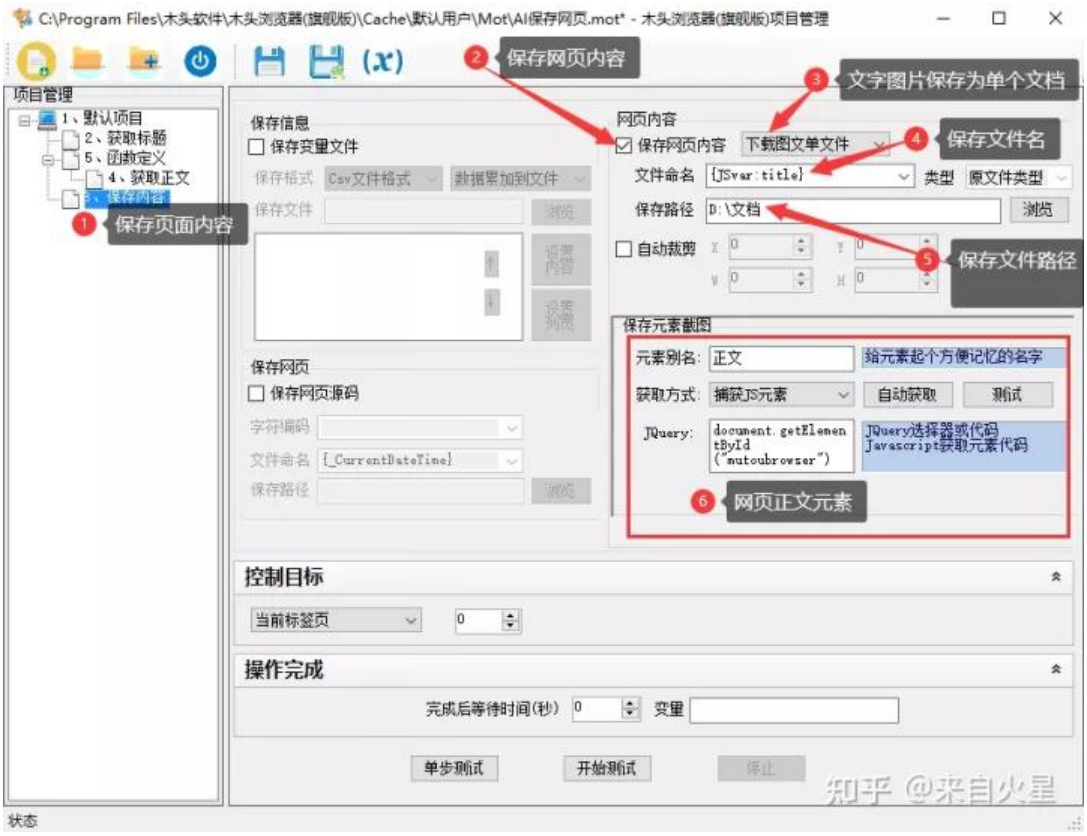


智能识别网页正文

```
var content="";
var fa=null;
var ps=document.getElementsByTagName("p");
if(ps.length>0)
{//取包含P的父元素
    fa=ps[0].parentElement;
    while(fa.parentElement.scrollWidth*fa.parentElement.scrollHeight>fa.scrollWidth*fa.scrollH
ight)
    {
        fa=fa.parentElement;
    }

    if(ps.length>1)
    {//找到第一个p和最后一个p的共同父元素
        do{
            var endfa=ps[ps.length-1].parentElement;
            var faxpath=generateFullXPath(fa);
            var endfaxpath=generateFullXPath(endfa);
            if(faxpath.indexOf(endfaxpath)==0)
            {fa=endfa;
                break;
            }
            else if(endfaxpath.indexOf(faxpath)==0)
            {

                break;
            }
            else
            {
                fa=fa.parentElement;
                endfa=endfa.parentElement;
            }
        }while(true);
    }
}
else
{//取页面中间最大的元素
    var w=document.body.clientWidth;
    var h= document.body.clientHeight;
    var el=document.elementsFromPoint(Math.round(w/2),Math.round(h/3*2));
    if(el!=null){
        var d=0;
        for(var i=0;i<el.length;i++){
        var e=el[i];
        var dd=e.scrollWidth*e.scrollHeight;
        if(dd>d*1.8)
        {
            fa=e;
        }
        d=dd;
        }
    }
```

```
    else
    {
        fa=el.document.body;
    }
}
    if(fa!=null)
    {
        fa.setAttribute("id","mutoubrowser");
        content=fa.innerHTML;
      }
content;
```

再创建一个保存内容步骤，勾选“保存网页内容”，选择“下载图文单个文件”。设置文件名为js变量itle，即标题做为文件名，并指定保存文件路径为“D:\文档”。在窗口下方设置正文的元素，通过js码获得。



保存图文到文件

```
document.getElementById("mutoubrowser");
```

最后保存项目文件为“AI保存网页”，通过点击书签按钮运行这个项目，就能智能识别网页标题和正，保存网页有效内容和图片了。