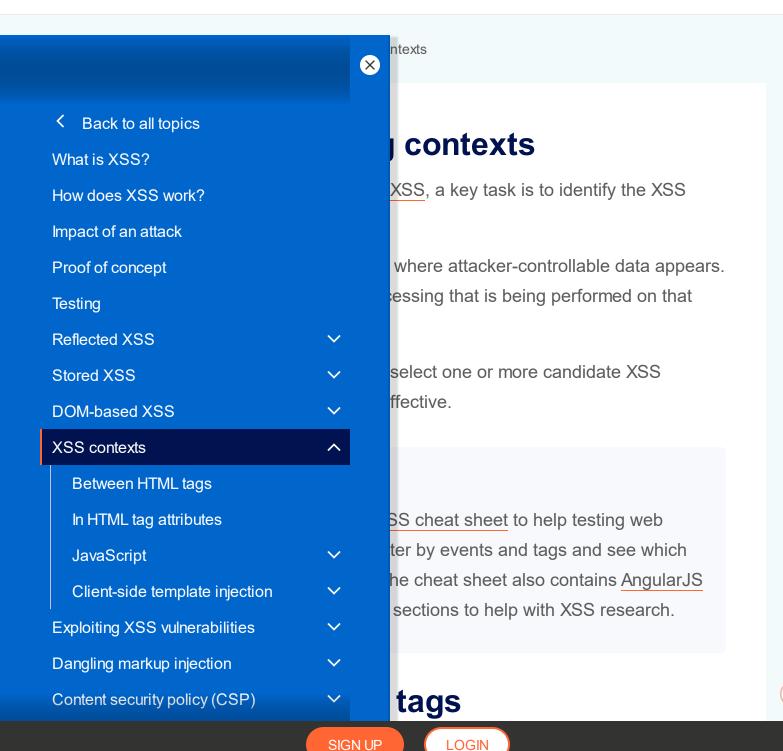
PortSwigger





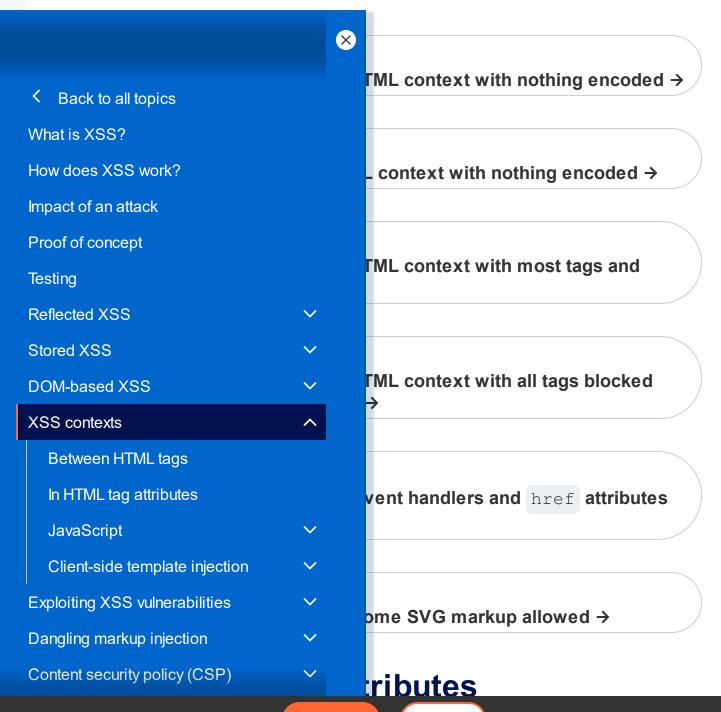
Academy home



When the XSS context is text between HTML tags, you need to introduce some new HTML tags designed to trigger execution of JavaScript.

Some useful ways of executing JavaScript are:

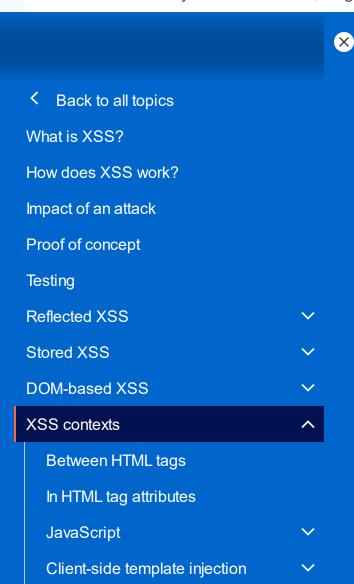
```
<script>alert(document.domain)</script>
<img src=1 onerror=alert(1)>
```



When the XSS context is into an HTML tag attribute value, you might sometimes be able to terminate the attribute value, close the tag, and introduce a new one. For example:

"><script>alert(document.domain)</script>

More commonly in this situation, angle brackets are blocked or encoded, so



Exploiting XSS vulnerabilities

Content security policy (CSP)

Dangling markup injection

in which it appears. Provided you can normally introduce a new attribute that an event handler. For example:

(document.domain) x="

and also adds the autofocus attribute utomatically without any user interaction.

tribute with angle brackets HTML-

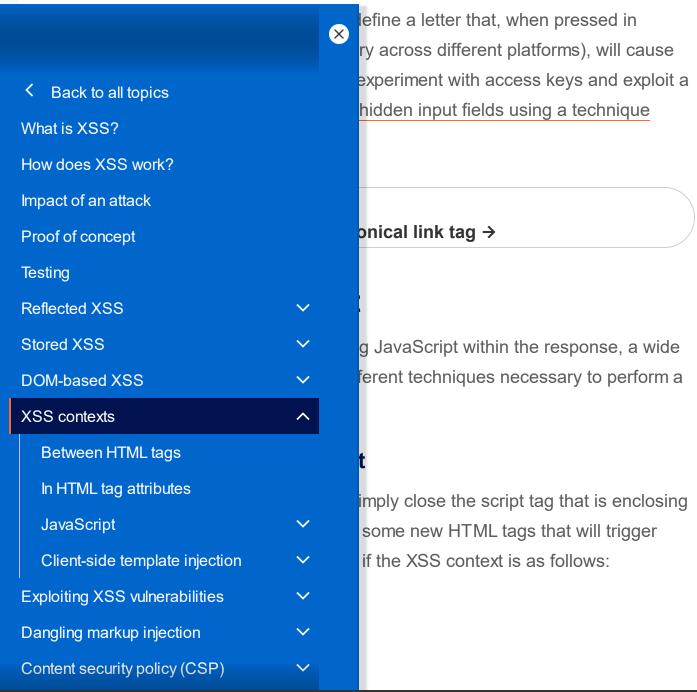
ype of HTML tag attribute that itself can can execute JavaScript without needing ample, if the XSS context is into the can use the javascript pseudo-

(document.domain)">



SIGN UP

You might encounter websites that encode angle brackets but still allow you to inject attributes. Sometimes, these injections are possible even within tags that don't usually fire events automatically, such as a canonical tag. You can exploit this behavior using access keys and user interaction on Chrome. Access keys allow you to provide keyboard shortcuts that reference a specific element. The

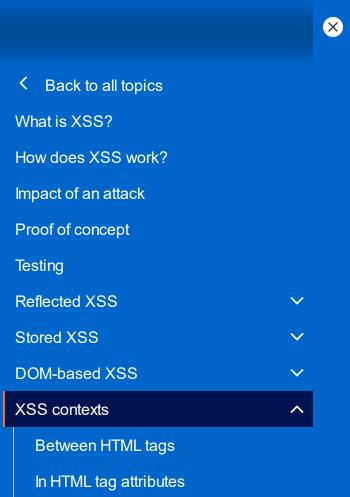


LOGIN

SIGN UP

```
<script>
...
var input = 'controllable data here';
...
</script>
```

then you can use the following payload to break out of the existing JavaScript



JavaScript

Client-side template injection

Exploiting XSS vulnerabilities

Content security policy (CSP)

Dangling markup injection

or=alert(document.domain)>

rser first performs HTML parsing to locks of script, and only later performs execute the embedded scripts. The pt broken, with an unterminated string psequent script being parsed and

JavaScript string with single quote ∍d →

string

ide a quoted string literal, it is often
execute JavaScript directly. It is essential
context, because any syntax errors there
cuting.

string literal are:



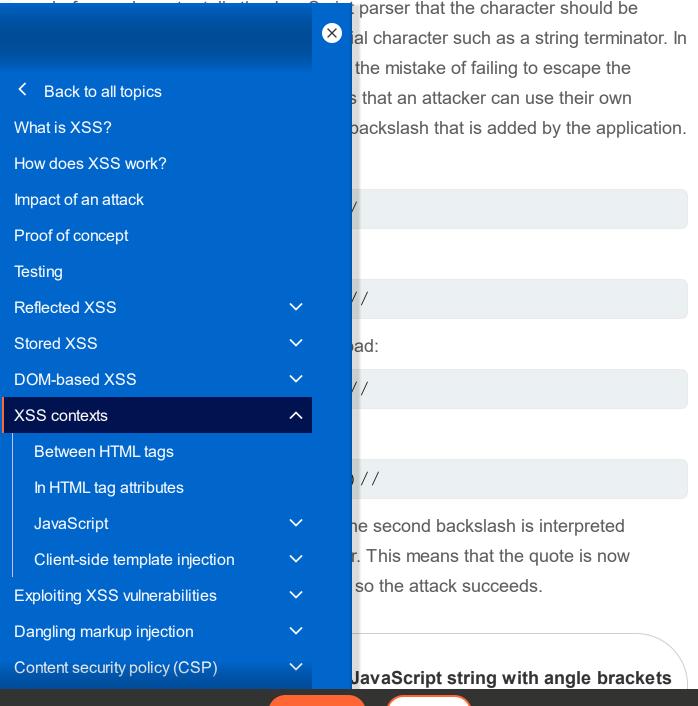
SIGN UP

∐ LAB

APPRENTICE

Reflected XSS into a JavaScript string with angle brackets HTML encoded →

Some applications attempt to prevent input from breaking out of the JavaScript string by escaping any single quote characters with a backslash. A backslash



SIGN UP

Some websites make XSS more difficult by restricting which characters you are allowed to use. This can be on the website level or by deploying a WAF that prevents your requests from ever reaching the website. In these situations, you need to experiment with other ways of calling functions which bypass these security measures. One way of doing this is to use the throw statement with

X Back to all topics What is XSS? How does XSS work? Impact of an attack Proof of concept **Testing** Reflected XSS Stored XSS DOM-based XSS XSS contexts Between HTML tags In HTML tag attributes **JavaScript** Client-side template injection Exploiting XSS vulnerabilities Dangling markup injection Content security policy (CSP)

u to pass arguments to a function without e assigns the alert() function to the ow statement passes the 1 to the c). The end result is that the alert() ent.

echnique to call functions without

that filters certain characters. You'll have cribed above in order to solve it.

vaScript URL with some characters

g JavaScript within a quoted tag attribute, e to make use of HTML-encoding to work

HTML tags and attributes within a ing of tag attribute values before they are

values before they are

SIGN UP

characters that are needed for a successful XSS exploit, you can often bypass the input validation by HTML-encoding those characters.

For example, if the XSS context is as follows:

and the application blocks or escapes single quote characters, you can use the

 (X)

avaScript string and execute your own



What is XSS?

How does XSS work?

Impact of an attack

Proof of concept

Testing

Reflected XSS

Stored XSS

DOM-based XSS

XSS contexts

Between HTML tags

In HTML tag attributes

JavaScript ~

Client-side template injection

Exploiting XSS vulnerabilities

Dangling markup injection

Content security policy (CSP)

ain)-'

entity representing an apostrophe or

ML-decodes the value of the onclick

preted, the entities are decoded as

s, and so the attack succeeds.

lick event with angle brackets and encoded and single quotes and

erals

literals that allow embedded JavaScript

ons are evaluated and are normally

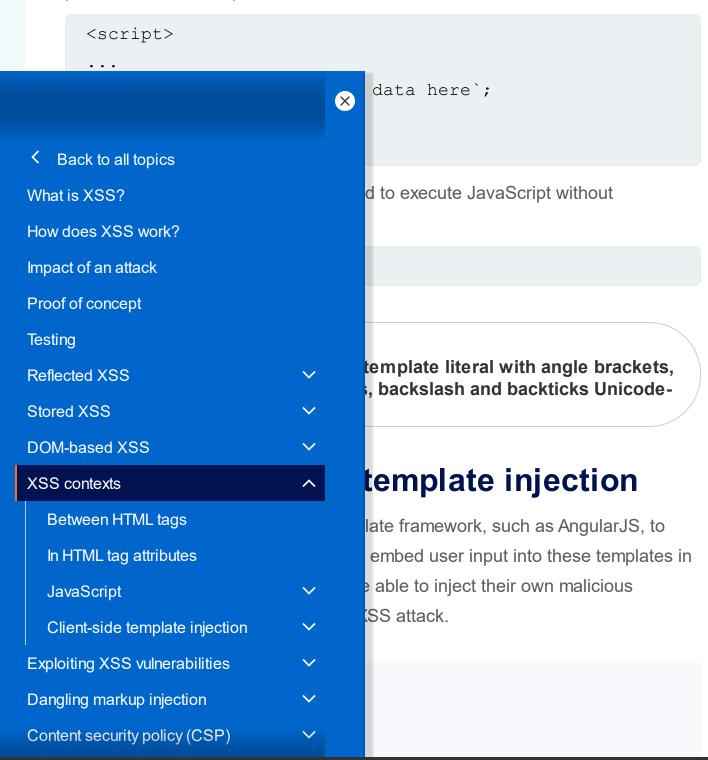
t. Template literals are encapsulated in marks, and embedded expressions are

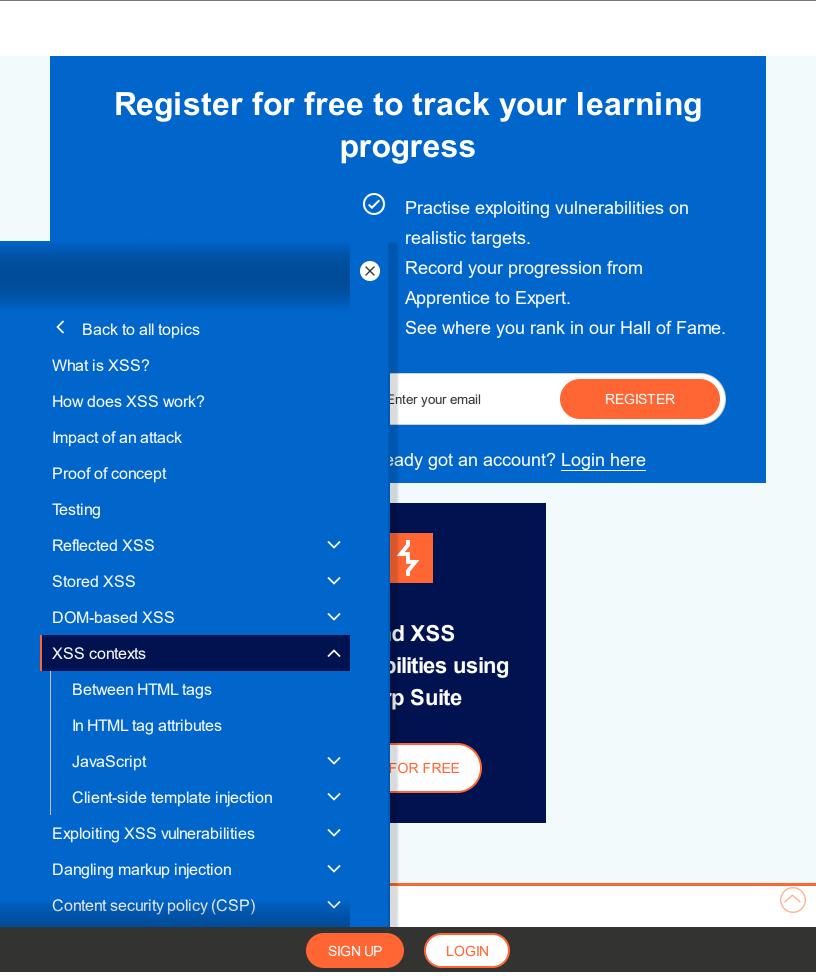
rint a welcome message that includes the



SIGN UP

When the XSS context is into a JavaScript template literal, there is no need to terminate the literal. Instead, you simply need to use the $\{\{\ldots\}\}$ syntax to embed a JavaScript expression that will be executed when the literal is processed. For example, if the XSS context is as follows:





Burp Suite

Web wilnerability scanner Burp Suite Editions Release Notes

Vulnerabilities

Cross-site scripting (XSS)

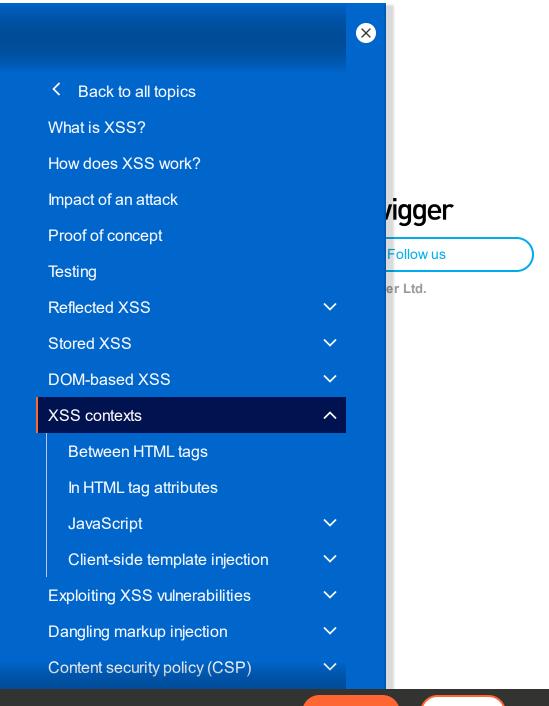
SQL injection

Cross-site request forgery

XML external entity injection

Directory traversal

Server-side request forgery





SIGN UP