

Forum:[Hack Forums](#)

Thread: [Eval](#)

CALL ME FORK - March 7, 2016, 6:31 pm

So I recently looking at some functions and I stumbled up on the Code eval function. Now I have a question. What do I use this for?

You see, I was confused on why I would use this instead of just plain print.Code eval('print("wow")')

wowCouldn't I just do this

Code print("wow")

wowDoesn't this do the same thing, but just more complicated?

Am I misusing this, or are there many other things you can do with this function?

If you guys could also provide an example, that would be good too.?

Goodies - March 7, 2016, 6:43 pm

Yes, those functions do the same thing. Try not to use exec() or eval() if you can afford it.

Sometimes it may be cheeky and useful to do so. For example:

PHP Code

```
<?php
class myClass:
    pass
myObject = myClass()
for i in range(1000):
    exec("myObject.var{} = {}".format(i, i**2))
```

What this would do is create an instance of myClass() and set a bunch of instance variables. You would now be able to reference:

PHP Code

```
<?php
print("{}".format(myObject.var773)) # => 597529
```

However, exec is not the best thing to use. You should be using setattr in that case. It works just the same.

PHP Code

```
<?php
myObject = myClass()
for i in range(1000):
    setattr(myObject, "var{}".format(i), i)
```

But that's just an example of when it could exec() could actually be useful.

But yeah, eval/exec should not be used, but this would be an example of eval():

PHP Code

```
<?php
# create the object with exec() above
sum = 0
for i in range(1000):
    sum += eval("myObject.var{}".format(i))
```

Even then, you should use `hasattr` to verify that it actually has the attribute and `getattr` to grab the value.

?Irs?LI? - March 7, 2016, 6:48 pm

Yeah ive found use for `eval` in PHP before where I read in a dump of an array in a string format, and could easily use it as such as a quick hack since it was 1-time code:

PHP Code

```
<?php
$string = "array ('a' => 'apple', 'b' => 'banana', 'c' => array ('x', 'y', 'z'))";
eval('$newArray = ' . $string);
echo $newArray['a']; // Prints apple
```

Of course normally you'd try and avoid using `eval` since it can be dangerous.

CALL_ME_FORK - March 9, 2016, 5:02 pm

So why is this system saying I tried in SQL injection when I was merely typing python code in the code function?

Anyway I made a calculator in python with the `eval` function. I do have another question, why don't people use this function? It is so much easier with this than with a normal calculator. Also, why should I stay way from the `eval` and `exec` function? Does it create bugs or security issues?

Goodies - March 9, 2016, 5:17 pm

(Mar 9, 2016 06:02 PM)CALL_ME_FORK Wrote: [\[CITE\]](#) So why is this system saying I tried in SQL injection when I was merely typing python code in the code function?

Anyway I made a calculator in python with the `eval` function. I do have another question, why don't people use this function? It is so much easier with this than with a normal calculator. Also, why should I stay way from the `eval` and `exec` function? Does it create bugs or security issues?
Because `eval` is dangerous.

Peeka - March 9, 2016, 6:34 pm

I saw this in the thread recent posts and I quickly thought "uh oh".
Try not to use `eval`, super dangerous.

Stonky - March 10, 2016, 10:28 am

(Mar 9, 2016 07:34 PM)Peekaboi Wrote: [\[CITE\]](#) I saw this in the thread recent posts and I quickly thought "uh oh".

Try not to use `eval`, super dangerous.
I back this. `Eval` isn't something you should use in production code.

CALL_ME_FORK - March 10, 2016, 3:23 pm

Is `eval` dangerous because anyone can input anything in them?
I was just using this for me anyway so I don't consider it too dangerous.

Peeka - March 10, 2016, 3:29 pm

(Mar 10, 2016 04:23 PM)CALL_ME_FORK Wrote: [\[CITE\]](#) Is `eval` dangerous because anyone can input anything in them?

I was just using this for me anyway so I don't consider it too dangerous.
Many hackers attempt to use eval because eval can evaluate any code from a input (base 64)

CALL_ME_FORK - *March 10, 2016, 3:43 pm*

Ah, thank you! Someone can close this thread now.