

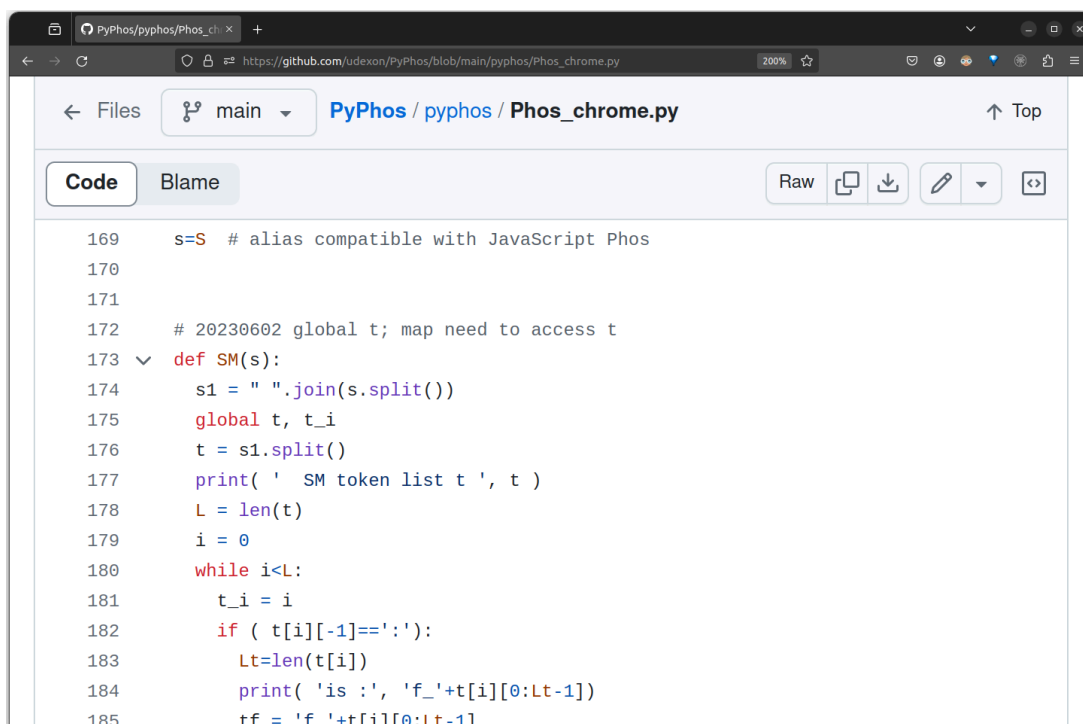
PyPhos

– Phoscript (Lightweight FORTH Shell) in Python

<https://github.com/udexon/PyPhos>

May 2024

1. Phoscript is a lightweight FORTH derived metaprogramming script, based on the following token processing algorithm, as shown in function SM from line 173 in Phos_chrome.py (figures 1A and 1B) below:
 - A. Line 176: split a SPACE delimited string into an array (list) of tokens.
 - B. Line 180: loop through the list of tokens:
 - i. (At the end of while loop) Line 218: push data tokens on to the stack S. (figure 1B)
 - ii. If token t matches any of the function token, evaluate (execute) the target function. (Lines 182 onwards.)

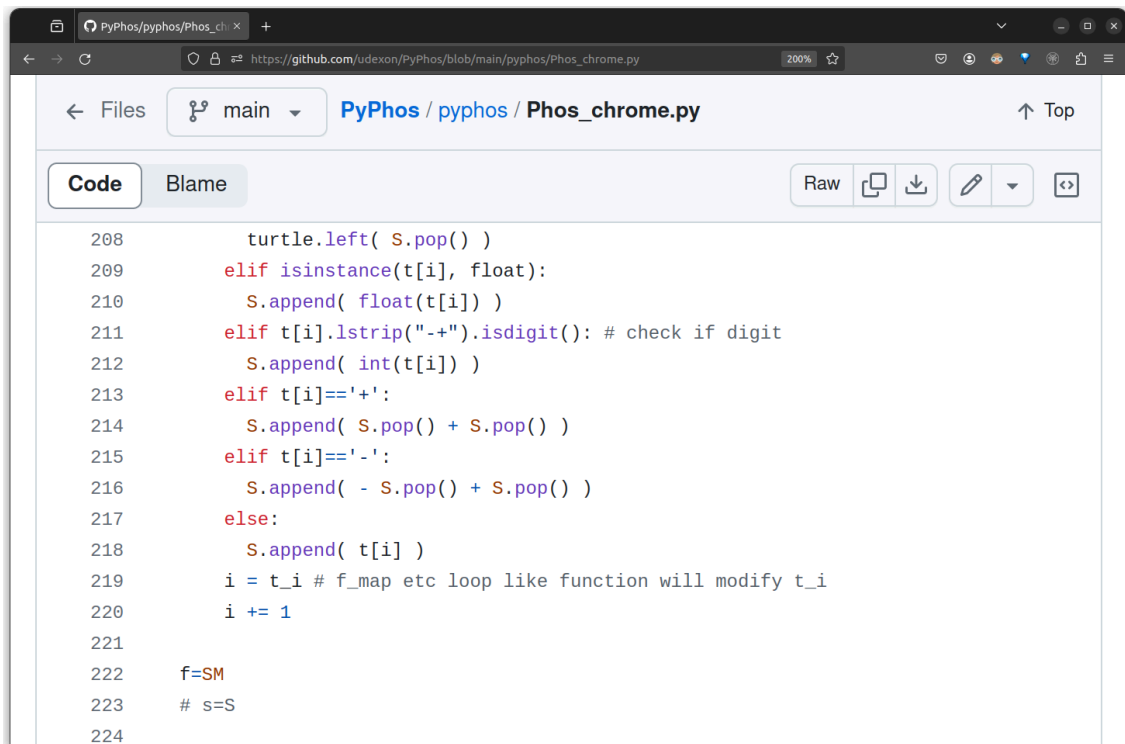


```
169 s=S # alias compatible with JavaScript Phos
170
171
172 # 20230602 global t; map need to access t
173 def SM(s):
174     s1 = " ".join(s.split())
175     global t, t_i
176     t = s1.split()
177     print( ' SM token list t ', t )
178     L = len(t)
179     i = 0
180     while i<L:
181         t_i = i
182         if ( t[i][-1]==':' ):
183             Lt=len(t[i])
184             print( 'is :', 'f_'+t[i][0:Lt-1])
185             tf = 'f_'+t[i][0:Lt-1]
```

Figure 1A

As such, Phoscript could well be the simplest introduction into FORTH and related stack machine (metaprogramming) languages, which can easily be picked up by anyone with 3 months of programming experience, and can be (theoretically and practically) ported to any known programming language, including but not limited to:

- C, C++, C#, Java, JavaScript, PHP, Python, Rust, LISP, Haskell, HTML, etc.



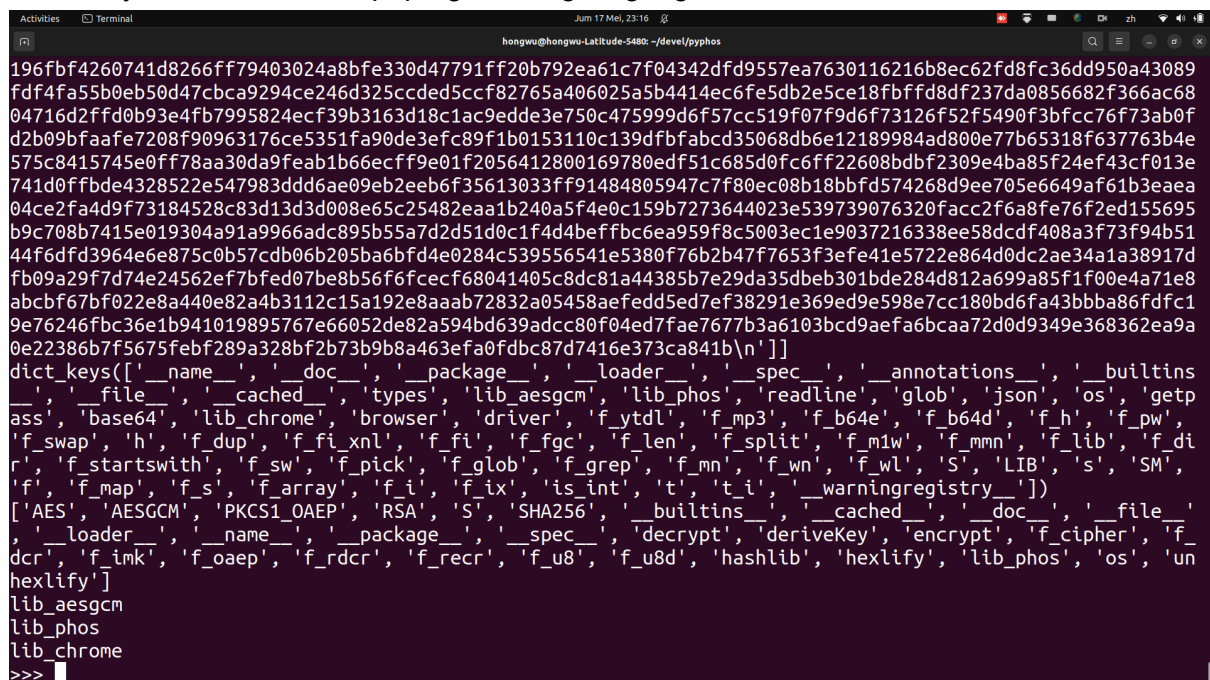
```
208     turtle.left( S.pop() )
209     elif isinstance(t[i], float):
210         S.append( float(t[i]) )
211     elif t[i].lstrip("-").isdigit(): # check if digit
212         S.append( int(t[i]) )
213     elif t[i]=='+' :
214         S.append( S.pop() + S.pop() )
215     elif t[i]=='-' :
216         S.append( - S.pop() + S.pop() )
217     else:
218         S.append( t[i] )
219     i = t_i # f_map etc loop like function will modify t_i
220     i += 1
221
222     f=SM
223     # s=S
224
```

Figure 1B

2. To illustrate, run Phos_chrome.py in python interactive mode:

```
$ python -i Pho_chrome.py
```

This will load all necessary libraries and open a Chrome web browser using Selenium library, which will serve to illustrate the power, simplicity and versatility of Phoscript as an interface to both Python and JavaScript programming languages.



```
196fbf4260741d8266ff79403024a8bfe330d47791ff20b792ea61c7f04342dfd9557ea7630116216b8ec62fd8fc36dd950a43089
fdf4fa55b0eb50d47cbca9294ce246d325ccded5ccf82765a406025a5b4414ec6fe5db2e5ce18fbffd8df237da0856682f366ac68
04716d2fdd0b93e4fb7995824ecf39b3163d18c1ac9edde3e750c475999d6f57cc519f07f9d6f73126f52f5490f3bfcc76f73ab0f
d2b09bf9aafef7208f90963176ce5351fa90de3efc89f1b0153110c139dfbfbcd35068db6e12189984ad800e77b65318f637763b4e
575c8415745e0ff78aa30da9feab1b66ecff9e01f2056412800169780edf51c685d0fc6ff22608bdfb2309e4ba85f24ef43cf013e
741d0fffbde4328522e547983ddd6ae09eb2eeb6f35613033ff91484805947c7f80ec08b18bbfd574268d9ee705e6649af61b3eaea
04ce2fa4d9f73184528c83d13d3d008e65c25482eaa1b240a5f4e0c159b7273644023e539739076320facc2f6a8fe76f2ed155695
b9c708b7415e019304a91a9966adc895b55a7d2d51d0c1f4d4beffbc6ea959f8c5003ec1e9037216338ee58dcdf408a3f73f94b51
44f6dfd3964e6e875c0b57cdeb06b205ba6bfd4e0284c539556541e5380f76b2b47f7653f3efe41e5722e864d0dc2ae34a1a38917d
fb09a29f7d74e24562ef7bfed07be8b56f6fcec68041405c8dc81a44385b7e29da35dbeb301bde284d812a699a85f1f00e4a71e8
abcbf67bf022e8a440e82a4b3112c15a192e8aaab72832a05458aefedd5ed7ef38291e369ed9e598e7cc180bd6fa43bbba86fdcf1
9e76246fbc36e1b941019895767e66052de82a594bd639adcc80f04ed7fae7677b3a6103bcd9aefa6bcaa72d0d9349e368362ea9a
0e22386b7f5675feb289a328bf2b73b9b8a463efa0fdb8c87d7416e373ca841b\n']]
dict_keys(['__name__', '__doc__', '__package__', '__loader__', '__spec__', '__annotations__', '__builtins__
__', '__file__', '__cached__', 'types', 'lib_aesgcm', 'lib_phos', 'readline', 'glob', 'json', 'os', 'getp
ass', 'base64', 'lib_chrome', 'browser', 'driver', 'f_ytdl', 'f_mp3', 'f_b64e', 'f_b64d', 'f_h', 'f_pw',
'f_swap', 'h', 'f_dup', 'f_fi_xnl', 'f_fi', 'f_fg', 'f_len', 'f_split', 'f_m1w', 'f_mmn', 'f_lib', 'f_di
r', 'f_startswith', 'f_sw', 'f_pick', 'f_glob', 'f_grep', 'f_mn', 'f_wn', 'f_wl', 'S', 'LIB', 's', 'SM',
'f', 'f_map', 'f_s', 'f_array', 'f_i', 'f_ix', 'is_int', 't', 't_i', '__warningregistry__'])
['AES', 'AESGCM', 'PKCS1_OAEP', 'RSA', 'S', 'SHA256', '__builtins__', '__cached__', '__doc__', '__file__
', '__loader__', '__name__', '__package__', '__spec__', 'decrypt', 'deriveKey', 'encrypt', 'f_cipher', 'f_
dcr', 'f_imk', 'f_oaep', 'f_rdc', 'f_recr', 'f_u8', 'f_u8d', 'hashlib', 'hexlify', 'lib_phos', 'os', 'un
hexlify']
lib_aesgcm
lib_phos
lib_chrome
>>>
```

Figure 2A

```

264 # must not have 2 line breaks, will stop function def in interpreter mode
265
266 SM("pvk.txt fi:")
267 print(S)
268
269 print( globals().keys() )
270 print( dir( lib_aesgcm ) )
271
272
273 for key in list(globals().keys()): # iter on both key and values
274     if key.startswith('lib_'):
275         print( key )
276
277 tab_phos = ''
278
279 # 11. windows_before = driver.current_window_handle
280 def f_tab_phos():

```

Figure 2B

The initialization code in line 266 loads a private key file and push it on to the stack, and line 267 prints the stack S, and can be seen in the top part of figure 2A. The remaining part of the screen are outputs from lines 269 to 275, which are the defined functions callable using Phoscript (FORTH) syntax.

Let us illustrate how Phoscript works by decrypting the encrypted private key file.

```

>>> S
[[ 'be0c6943b402575b-6723892835e9fc7d79b00268-97f7b13bf1e6e64c6c470e9fb0064aca339293a492786407d0d648dd2d98
66d2d3f31ab308b5c6ffc978c81e4e2ec0fca4202e9193252d93ce57290d6171cab0d85509536c1cd557765d0493d1317af4678ea
01210950f56fa198e8d82f0b7e3add5eb71a00ce5a4d65d9f4bc828631c7bb7eb8b8b7592f51e0dd28ea4099cd86e2d5f9f8b2229
95158ff3076d321aa1620d921440ef8e1f98a524a7745f214ae862e8b3a714b38c0872050584db00554b4984dbe13306081e95a3f
f1da10cf41108ce6641fbac87c56e1d74f56cc41ccd11c32a43528f5736999da65a9394d770a024b3ac2ef276afe05d4697c5e68f
196fbf4260741d8266ff79403024a8bfe330d47791ff20b792ea61c7f04342dfd9557ea7630116216b8ec62fd8fc36dd950a43089
fdf4fa55b0eb50d47cbca9294ce246d325ccded5ccf82765a406025a5b4414ec6fe5db2e5ce18fbffd8df237da0856682f366ac68
04716d2ffdd0b93e4fb7995824ecf39b3163d18c1ac9edde3e750c475999d6f57cc519f07f9d6f73126f52f5490f3bfcc76f73ab0f
d2b09bfaafe7208f90963176ce5351fa90de3efc89f1b0153110c139dfbfabacd35068db6e12189984ad800e77b65318f637763b4e
575c8415745e0ff78aa30da9feab1b66ecff9e01f2056412800169780edf51c685d0fc6ff22608bdf2309e4ba85f24ef43cf013e
741d0fffbde4328522e547983ddd6ae09eb2eeb6f35613033ff91484805947c7f80ec08b18bbfd574268d9ee705e6649af61b3eaea
04ce2fa4d9f73184528c83d13d3d008e65c25482eaa1b240a5f4e0c159b7273644023e539739076320facc2f6a8fe76f2ed155695
b9c708b7415e019304a91a9966adc895b55a7d2d51d0c1f4d4beffbc6ea959f8c5003ec1e9037216338ee58dcdf408a3f73f94b51
44f6dfd3964e6e875c0b57cd06b205ba6bfd4e0284c539556541e5380f76b2b47f7653f3efe41e5722e864d0dc2ae34a1a38917d
fb09a29f7d74e24562ef7bfed07be8b56f6fcecfc68041405c8dc81a44385b7e29da35dbeb301bde284d812a699a85f1f00e4a71e8
abcbf67bf022e8a440e82a4b3112c15a192e8aaab72832a05458aefedd5ed7ef38291e369ed9e598e7cc180bd6fa43bbba86dfc1
9e76246fbc36e1b941019895767e66052de82a594bd639adcc80f04ed7fae7677b3a6103bcd9aefa6bcaa72d0d9349e368362ea9a
0e22386b7f5675febfb289a328bf2b73b9b8a463efa0fdbc87d7416e373ca841b\n' ]]
>>> s.append(s[0][0][: -1])

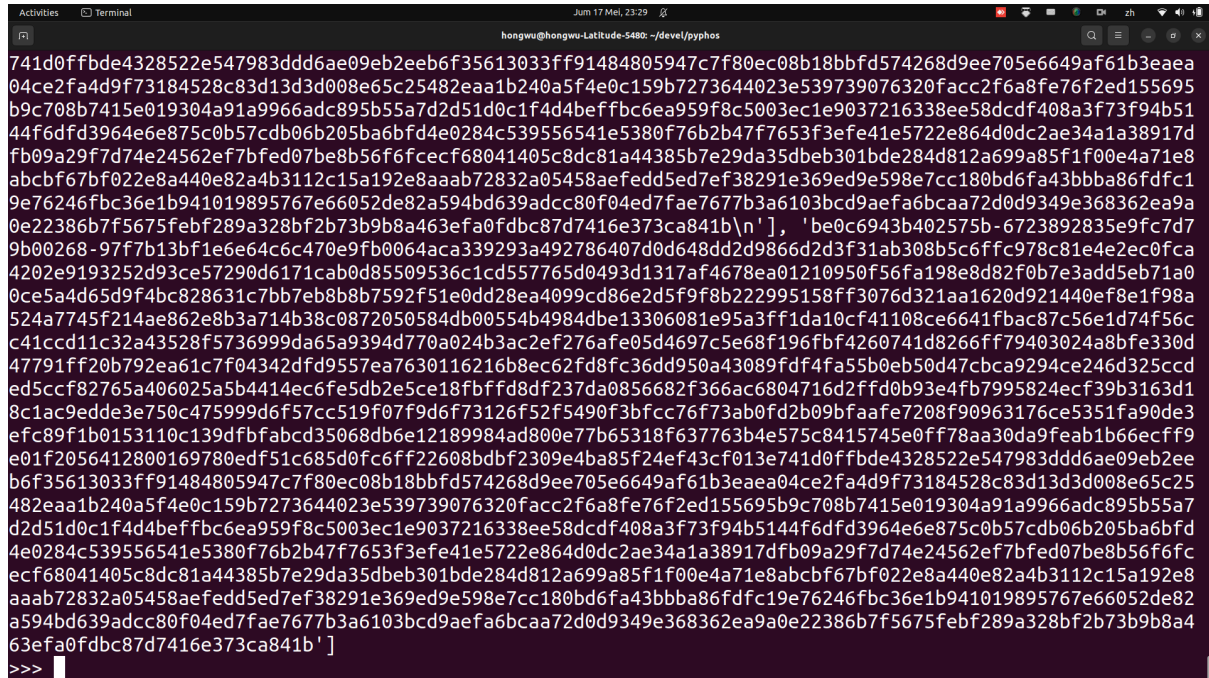
```

Figure 3A

Lowercase s (alias to uppercase S) will show the stack, as shown in figure 3A.

```
>>> s.append(s[0][0][:-1])
```

The command above drop the last newline character from the contents of the private key file, and push the result on to the stack, as s[1], as shown in figure 3B.



```
741d0ffbfde4328522e547983ddd6ae09eb2eeb6f35613033ff91484805947c7f80ec08b18bbfd574268d9ee705e6649af61b3eaea
04ce2fa4d9f73184528c83d13d3d008e65c25482eaa1b240a5f4e0c159b7273644023e539739076320facc2f6a8fe76f2ed155695
b9c708b7415e019304a91a9966adc895b55a7d2d51d0c1f4d4beffbc6ea959f8c5003ec1e9037216338ee58dcdf408a3f73f94b51
44f6dfd3964e6e875c0b57cd06b205ba6bfd4e0284c539556541e5380f76b2b47f7653f3efe41e5722e864d0dc2ae34a1a38917d
fb09a29f7d74e24562ef7bfed07be8b56f6fcec68041405c8dc81a44385b7e29da35dbeb301bde284d812a699a85f1f00e4a71e8
abcbf67bf022e8a440e82a4b3112c15a192e8aaab72832a05458aefedd5ed7ef38291e369ed9e598e7cc180bd6fa43bbba86fdcf1
9e76246fbc36e1b941019895767e66052de82a594bd639adcc80f04ed7fae7677b3a6103bcd9aefa6bcaa72d0d9349e368362ea9a
0e22386b7f5675febf289a328bf2b73b9b8a463efa0fdb87d7416e373ca841b\n'], 'be0c6943b402575b-6723892835e9fc7d7
9b00268-97f7b13bf1e6e64c6c470e9fb0064aca339293a492786407d0d648dd2d9866d2d3f31ab308b5c6ffcc978c81e4e2ec0fca
4202e9193252d93ce57290d6171cab0d85509536c1cd557765d0493d1317af4678ea01210950f56fa198e8d82f0b7e3add5eb71a0
0ce5a4d65d9f4bc828631c7bb7eb8b8b7592f51e0dd28ea4099cd86e2d5f9f8b222995158ff3076d321aa1620d921440ef8e1f98a
524a7745f214ae862e8b3a714b38c0872050584db00554b4984dbe13306081e95a3ff1da10cf41108ce6641fbac87c56e1d74f56c
c41ccd11c32a43528f5736999da65a9394d770a024b3ac2ef276afe05d4697c5e68f196fbf4260741d8266ff79403024a8bfe330d
47791ff20b792ea61c7f04342dfd9557ea7630116216b8ec62fd8fc36dd950a43089fdf4fa55b0eb50d47cbca9294ce246d325ccd
ed5ccf82765a406025a5b4414ec6fe5db2e5ce18fbffdf8df237da0856682f366ac6804716d2ffd0b93e4fb7995824ecf39b3163d1
8c1ac9edde3e750c475999d6f57cc519f07f9d6f73126f52f5490f3bfcc76f73ab0fd2b09bfaafe7208f90963176ce5351fa90de3
efc89f1b0153110c139dfbfabacd35068db6e12189984ad800e77b65318f637763b4e575c8415745e0ff78aa30da9feab1b66ecff9
e01f2056412800169780edf51c685d0fc6ff22608dbdf2309e4ba85f24ef43cf013e741d0ffbfde4328522e547983ddd6ae09eb2ee
b6f35613033ff91484805947c7f80ec08b18bbfd574268d9ee705e6649af61b3eaea04ce2fa4d9f73184528c83d13d3d008e65c25
482eaa1b240a5f4e0c159b7273644023e539739076320facc2f6a8fe76f2ed155695b9c708b7415e019304a91a9966adc895b55a7
d2d51d0c1f4d4beffbc6ea959f8c5003ec1e9037216338ee58dcdf408a3f73f94b5144f6dfd3964e6e875c0b57cd06b205ba6bfd
4e0284c539556541e5380f76b2b47f7653f3efe41e5722e864d0dc2ae34a1a38917dfb09a29f7d74e24562ef7bfed07be8b56f6fc
ecf68041405c8dc81a44385b7e29da35dbeb301bde284d812a699a85f1f00e4a71e8abcbf67bf022e8a440e82a4b3112c15a192e8
aaab72832a05458aefedd5ed7ef38291e369ed9e598e7cc180bd6fa43bbba86fdcf19e76246fbc36e1b941019895767e66052de82
a594bd639adcc80f04ed7fae7677b3a6103bcd9aefa6bcaa72d0d9349e368362ea9a0e22386b7f5675febf289a328bf2b73b9b8a4
63efa0fdb87d7416e373ca841b']
>>>
```

Figure 3B



```
fb09a29f7d74e24562ef7bfed07be8b56f6fcec68041405c8dc81a44385b7e29da35dbeb301bde284d812a699a85f1f00e4a71e8
abcbf67bf022e8a440e82a4b3112c15a192e8aaab72832a05458aefedd5ed7ef38291e369ed9e598e7cc180bd6fa43bbba86fdcf1
9e76246fbc36e1b941019895767e66052de82a594bd639adcc80f04ed7fae7677b3a6103bcd9aefa6bcaa72d0d9349e368362ea9a
0e22386b7f5675febf289a328bf2b73b9b8a463efa0fdb87d7416e373ca841b\n'], 'be0c6943b402575b-6723892835e9fc7d7
9b00268-97f7b13bf1e6e64c6c470e9fb0064aca339293a492786407d0d648dd2d9866d2d3f31ab308b5c6ffcc978c81e4e2ec0fca
4202e9193252d93ce57290d6171cab0d85509536c1cd557765d0493d1317af4678ea01210950f56fa198e8d82f0b7e3add5eb71a0
0ce5a4d65d9f4bc828631c7bb7eb8b8b7592f51e0dd28ea4099cd86e2d5f9f8b222995158ff3076d321aa1620d921440ef8e1f98a
524a7745f214ae862e8b3a714b38c0872050584db00554b4984dbe13306081e95a3ff1da10cf41108ce6641fbac87c56e1d74f56c
c41ccd11c32a43528f5736999da65a9394d770a024b3ac2ef276afe05d4697c5e68f196fbf4260741d8266ff79403024a8bfe330d
47791ff20b792ea61c7f04342dfd9557ea7630116216b8ec62fd8fc36dd950a43089fdf4fa55b0eb50d47cbca9294ce246d325ccd
ed5ccf82765a406025a5b4414ec6fe5db2e5ce18fbffdf8df237da0856682f366ac6804716d2ffd0b93e4fb7995824ecf39b3163d1
8c1ac9edde3e750c475999d6f57cc519f07f9d6f73126f52f5490f3bfcc76f73ab0fd2b09bfaafe7208f90963176ce5351fa90de3
efc89f1b0153110c139dfbfabacd35068db6e12189984ad800e77b65318f637763b4e575c8415745e0ff78aa30da9feab1b66ecff9
e01f2056412800169780edf51c685d0fc6ff22608dbdf2309e4ba85f24ef43cf013e741d0ffbfde4328522e547983ddd6ae09eb2ee
b6f35613033ff91484805947c7f80ec08b18bbfd574268d9ee705e6649af61b3eaea04ce2fa4d9f73184528c83d13d3d008e65c25
482eaa1b240a5f4e0c159b7273644023e539739076320facc2f6a8fe76f2ed155695b9c708b7415e019304a91a9966adc895b55a7
d2d51d0c1f4d4beffbc6ea959f8c5003ec1e9037216338ee58dcdf408a3f73f94b5144f6dfd3964e6e875c0b57cd06b205ba6bfd
4e0284c539556541e5380f76b2b47f7653f3efe41e5722e864d0dc2ae34a1a38917dfb09a29f7d74e24562ef7bfed07be8b56f6fc
ecf68041405c8dc81a44385b7e29da35dbeb301bde284d812a699a85f1f00e4a71e8abcbf67bf022e8a440e82a4b3112c15a192e8
aaab72832a05458aefedd5ed7ef38291e369ed9e598e7cc180bd6fa43bbba86fdcf19e76246fbc36e1b941019895767e66052de82
a594bd639adcc80f04ed7fae7677b3a6103bcd9aefa6bcaa72d0d9349e368362ea9a0e22386b7f5675febf289a328bf2b73b9b8a4
63efa0fdb87d7416e373ca841b']
>>> f('pw: dcr:')
SM token list t ['pw:', 'dcr:']
is : f_pw
f_pw is defined
Password: 
```

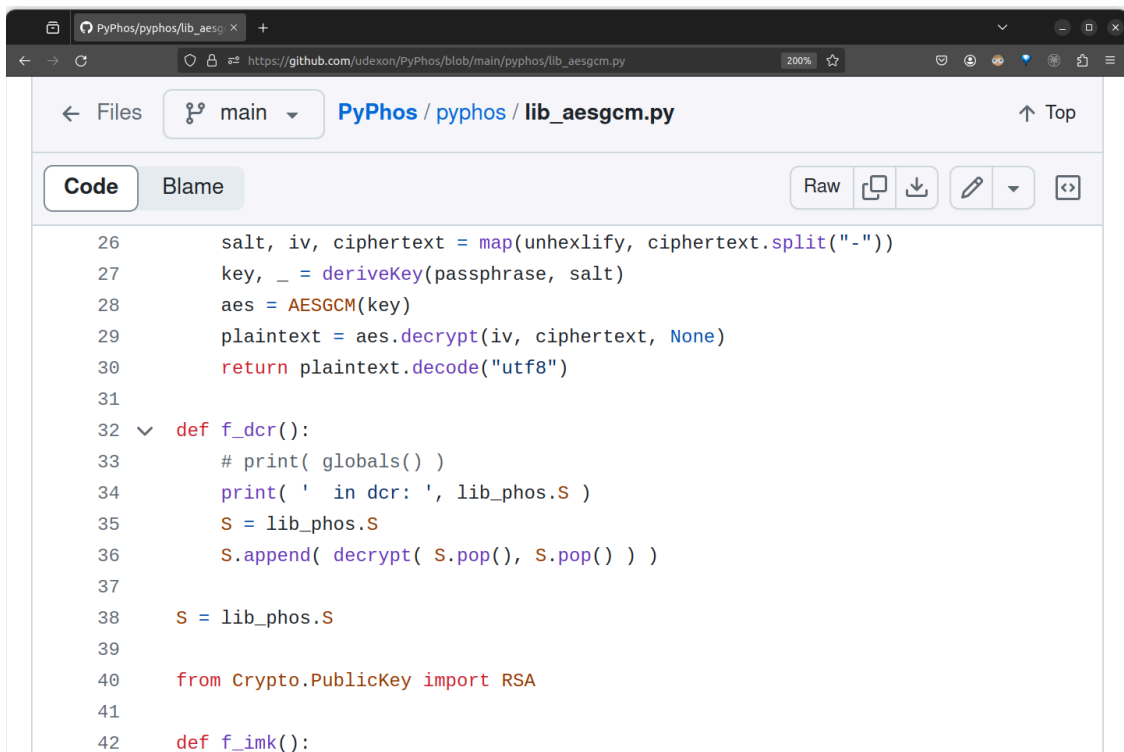
Figure 3C

```
>>> f('pw: dcr:')
```

Enter the command above as shown in figure 3C.

pw: will prompt the user for a decryption password, which should be '12345' (without the quotation mark).

dcr: will map to function `f_dcr` in line 32 `lib_aesgcm.py`, as shown in figure 3D.



```
26     salt, iv, ciphertext = map(unhexlify, ciphertext.split("-"))
27     key, _ = deriveKey(passphrase, salt)
28     aes = AESGCM(key)
29     plaintext = aes.decrypt(iv, ciphertext, None)
30     return plaintext.decode("utf8")
31
32     def f_dcr():
33         # print( globals() )
34         print( ' in dcr: ', lib_phos.S )
35         S = lib_phos.S
36         S.append( decrypt( S.pop(), S.pop() ) )
37
38     S = lib_phos.S
39
40     from Crypto.PublicKey import RSA
41
42     def f_imk():
```

Figure 3D

Finally, entering 's' will show the decrypted private key (figure 3E).

Users are encourage to read the source code and try out other functions, and also add own customised functions, as Phoscript is designed to be extensible by users themselves!!

And finally welcome to the brave new world of Metaprogramming, the best kept secret in Silicon Valley and computer science faculties around the world, the gateway into mathematics and metaphysics!!

fiat lux, Genesis 1:3 (Hebrew: "יהי אור")

Latin: "dixitque Deus fiat lux et facta est lux"

("And said God let there be light, and there was light"),

Greek

"και εἶπεν ὁ Θεός γενηθήτω φως καὶ ἐγένετο φως"
(kai eipen ho Theos genēthēto phōs kai egeneto phōs)

Hebrew

"וַיֹּאמֶר אֱלֹהִים, יְהִי אוֹר; וַיְהִי אוֹר"
(vayo'mer 'Elohim, yehi 'or vayehi 'or)

```
Activities Terminal Jun 17 Mei, 23:33 hongwu@hongwu-Latitude-5480: ~/dev/yyphos
66d2d3f31ab308b5c6ffc978c81e4e2ec0fca4202e9193252d93ce57290d6171cab0d85509536c1cd557765d0493d1317af4678ea
01210950f56fa198e8d82f0b7e3add5eb71a00ce5a4d65d9f4bc828631c7bb7eb8b8b7592f51e0dd28ea4099cd86e2d5f9f8b2229
95158ff3076d321aa1620d921440ef8e1f98a524a7745f214ae862e8b3a714b38c0872050584db00554b4984dbe13306081e95a3f
f1da10cf41108ce6641fbac87c56e1d74f56cc41ccd11c32a43528f5736999da65a9394d770a024b3ac2ef276afe05d4697c5e68f
196fbf4260741d8266ff79403024a8bfe330d47791ff20b792ea61c7f04342dfd9557ea7630116216b8ec62fd8fc36dd950a43089
fd4fa55b0eb50d47cbca9294ce246d325ccded5ccf82765a406025a5b441ec6fe5db2e5ce18fbfbd8df237da0856682f366ac68
04716d2ffd0b93e4fb7995824ecf39b3163d18c1ac9edde3e750c475999d6f57cc519f07f9d6f73126f52f5490f3bfcc76f73ab0f
d2b09bfaafe7208f90963176ce5351fa90de3efc89f1b0153110c139dfbfabacd35068db6e12189984ad800e77b65318f637763b4e
575c8415745e0fff78aa30da9feab1b66ecff9e01f2056412800169780edf51c685d0fc6ff22608bdf2309e4ba85f24ef43cf013e
741d0fffbde4328522e547983ddd6ae09eb2eeb6f35613033ff91484805947c7f80ec08b18bbfd574268d9ee705e6649af61b3eaea
04ce2fa4d9f73184528c83d13d3d008e65c25482eaa1b240a5f4e0c159b7273644023e539739076320facc2f6a8fe76f2ed155695
b9c708b7415e019304a91a9966adc895b55a7d2d51d0c1f4d4beffbc6ea959f8c5003ec1e9037216338ee58dcd408a3f73f94b51
44f6dfd3964e6e875c0b57cdbc06b205ba6bfd4e0284c539556541e5380f76b2b47f7653f3efe41e5722e864d0dc2ae34a1a38917d
fb09a29f7d74e24562ef7bfed07be8b56f6fc6cf68041405c8dc81a44385b7e29da35dbeb301bde284d812a699a85f1f00e4a71e8
abcbf67bf022e8a440e82a4b3112c15a192e8aaab72832a05458aefedd5ed7ef38291e369ed9e598e7cc180bd6fa43bbba86fdcf1
9e76246fbc36e1b941019895767e66052de82a594bd639adcc80f04ed7fae7677b3a6103bcd9aefa6bcaa72d0d9349e368362ea9a
0e22386b7f5675feb289a328bf2b73b9b8a463efa0fdbc87d7416e373ca841b\n'], '-----BEGIN RSA PRIVATE KEY-----\nMI
IICWwIBAAKBggGSK0htbffdcbdpGHEzhW9zk5GFQQ2yVZz+fnOx3HVMQd5jFUNWg/ntJCmw/uotHQFhvrfeHiPomrGxRxDq/Qx39vLzwL3
GKGxdujrWbCoNtk9PE+cxeRh\nwSV0mX/9drbKJx37TRAoo9ScaA1/CLznATOfnb7AnE15sgWMzsiF1bpTAgMBAAC\nngYARclAK13mx
jDK1Y9EfHwPAlA70JWJ1KxrffQUE/UavQBibGwNqeskPZMH1yJ8Z\nZ1Jt1vHYqGf52g20JgpKa6VMjaHp8Yks4+TDBcWmZF8SK+E4ydli
9yaBq4bVndwm\nnpq2bACw21t020smjtfzaFLSxgqHkRINjpBDyggS7GD+ckQJBANTu9/Afwdihf/wy\nnn6iLKYNJESZHLG/PX/cs7EqSB
PitNguJbJB6HjPfbBMaxJC2kln0yukqM1sw6yAz\nnE/VEPlkCQQCAq+OC/LiMXgdL0LGU4iWJARfQpQzgcD3x5U0VDoNfobhxxRh+WCU4
\nc+LcZIr9GnJ4UmZWihDZBrmAjETZyuCLAKeAL9X9MeaHl4JAun29BD7nCzRZ7Zln\nnyuZPnRU0m5TWY8xw7Ty2JnKZm4Liyr/m68upE
QOKXDSbPKpox+sQEFKyQJAEe3h\nnqeMfg5A0n8AinohVcnvAgcY94RhCyw4WK5T2IWE7fAQmwZQuu4sKVYN9Cn+g67sv\nnn1jn2BXjy1
HAVki6uQJAMC71Cqooi0808rz87ESg3FjxEzgGCuOr1gc4m7LuCgUM\nU9APGHZHAf64J3M+hRX/21qjLYIEiPQ0a3Rfwr49Pw==\n---
--END RSA PRIVATE KEY-----']
>>>
```

Figure 3E