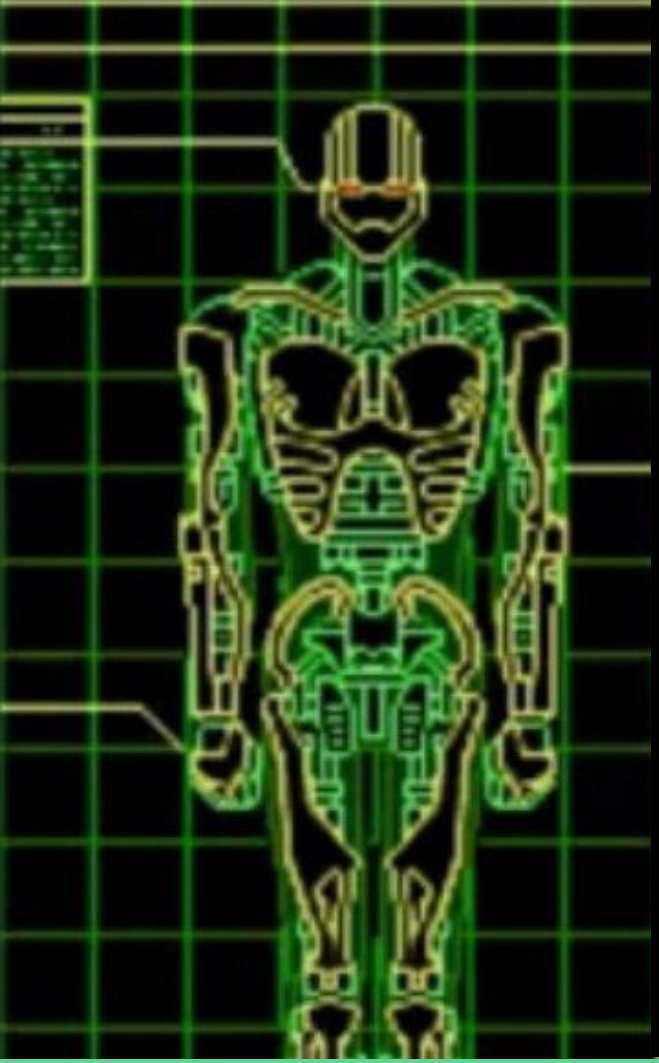
Wrapping C libraries into Python modules



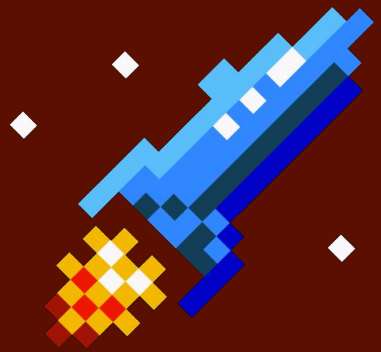
Agenda

- Porque um módulo C
- Como criar um módulo
- Como instalar um módulo
- Propriedades
- Como utilizar um módulo
- Experimento
- Comparação

Por que um módulo em C ?

- Computação muito pesada (numpy)
- Aumentar desempenho (Scipy)
- Não quero reescrever bibliotecas maduras escritas em C (PIL)
- Controle de recursos de baixo nível (memória RAM)
- Embutir o Python em alguma aplicação (camadas)

Como criar um módulo



```c
#include <Python.h>
```

Python.h

- Objetos
- Funções
- Tipos
- Macros

- Inclui *headers* comuns

{stdio,string,errno,limits,assert,stdlib}.h

https://docs.python.org/3/c-api/intro.html

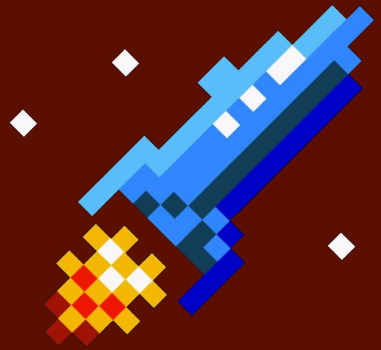
Python.h

- Variáveis e funções possuem prefixos
  - Py - Deve-se usar nos módulos
  - _Py - De uso interno do interpretador

- Não criar variáveis com esses prefixos
  - Confunde o interpretador
  - Legibilidade!
  - Conflitos em futuras versões do Python

https://docs.python.org/3/c-api/intro.html

Como criar um módulo



```c
#include <Python.h>


static PyObject *
hello (PyObject *self)
{
    return Py_BuildValue("s", "Hello Pythonista");
}
```

Objetos

```
static PyObject *
```

- Retornar PyObject *
  - Referência para um objeto opaco
- A maioria dos objetos Python estão no *Heap*
  - Pymalloc != *malloc*

https://docs.python.org/3/c-api/module.html

Definindo funções

```c
static PyObject* my_function_with_no_args (PyObject *self);


static PyObject* my_function (PyObject *self, PyObject *args);


static PyObject* my_function_with_keywords (PyObject *self,
                                            PyObject *args, PyObject *kwargs);
```

Objetos

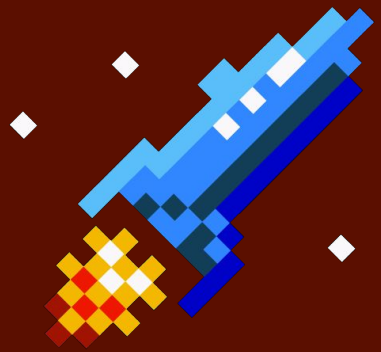
```
return Py_BuildValue("s", "Hello Pythonista");
```

```
PyObject* Py_BuildValue (const char *format, ...)
```

format	C type
c	char
f	float
i	int
d	double

format	C type
u	Py_UNICODE*
O	PyObject*
[...]	...
{...}	...

Como criar um módulo

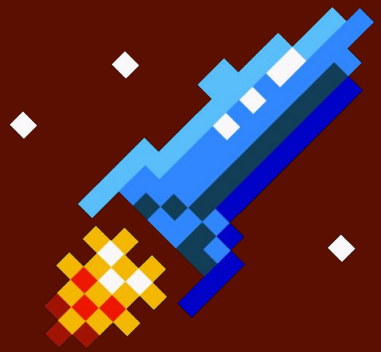


```c
#include <Python.h>

static PyObject *
hello (PyObject *self)
{
    return Py_BuildValue("s", "Hello Pythonista");
}

static char docstring[] = "Hello world module for Python written in C";
```

Como criar um módulo



```c
#include <Python.h>

static PyObject *
hello (PyObject *self)
{
    return Py_BuildValue("s", "Hello Pythonista");
}

static char docstring[] = "Hello world module for Python written in C";
static PyMethodDef module_methods[] = {
    {"hello", (PyCFunction) hello, METH_NOARGS, docstring},
    {NULL, NULL, 0, NULL}
};
```

Lista de funções

```c
struct PyMethodDef {
    char *ml_name;          /* Nome no módulo Python */
    PyCFunction ml_meth;    /* Endereço da função */
    int ml_flags;           /* Opções de assinatura de funções */
    char *ml_doc;           /* Docstring da função */
};
```

https://docs.python.org/3/c-api/structures.html#c.PyMethodDef

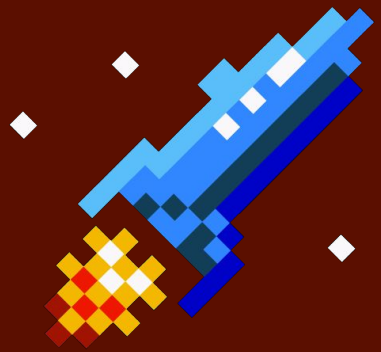
Lista de funções

```
{"hello", (PyCFunction) hello, METH_NOARGS, docstring}
```

- METH_NOARGS
- METH_VARARGS
- METH_KEYWORDS

https://docs.python.org/2/c-api/structures.html

Como criar um módulo



```c
#include <Python.h>

static PyObject *
hello (PyObject *self)
{
    return Py_BuildValue("s", "Hello Pythonista");
}

static char module_docstring[] = "Hello world module for Python written in C";
static PyMethodDef module_methods[] = {
    {"hello", (PyCFunction) hello, METH_NOARGS, module_docstring},
    {NULL, NULL, 0, NULL}
};

PyMODINIT_FUNC
initmodule(void)
{
    Py_InitModule("module", module_methods);
}
```

Inicializando o módulo

```
/* Python 2 */
PyMODINIT_FUNC init<yourmodulename>(void)


/* Python 3 */
PyMODINIT_FUNC PyInit_<yourmodulename>(void)
```

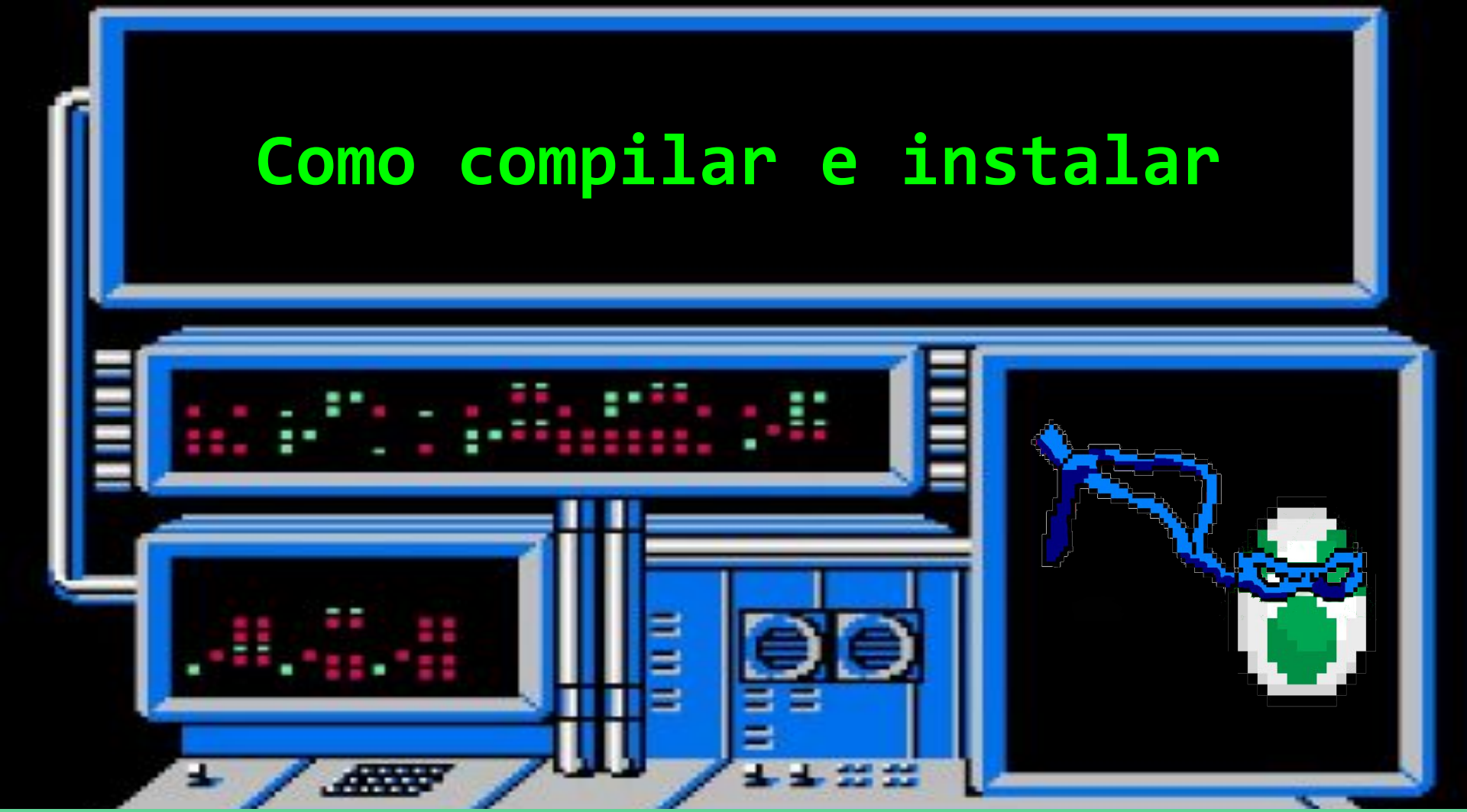
Inicializando o módulo

```
PyObject* Py_InitModule(char *name, PyMethodDef *methods)

PyObject* Py_InitModule3(
            char *name, PyMethodDef *methods, char *doc)

PyObject* Py_InitModule4(char *name, PyMethodDef *methods,
                    char *doc, PyObject *self, int apiver)
```

Como compilar e instalar



Instalação

setup.py

```
from distutils.core import setup
from distutils.core import Extension


setup(
    name='module',
    version='1.0',
    ext_modules=[Extension('module', ['hello.c'])]
)
```

Extension

```
Extension('module', ['hello.c'])
```

- Usado para descrever extensões C/C++ no *setup*
- Apenas um conjunto de atributos
- Quando existem extensões aciona-se o *build_ext*

`cpython/Lib/distutils/extension.py`

Extension

Opções do Extension

name	library_dirs
sources	extra_compile_args
include_dirs	extra_link_args
define_macros	depends

https://docs.python.org/2/distutils/apiref.html#distutils.core.Extension

Extension

```
class build_ext(Command):
```

```
objects = self.compiler.compile(
    sources,
    output_dir=self.build_temp,
    macros=macros,
    include_dirs=ext.include_dirs,
    debug=self.debug,
    extra_postargs=extra_args,
    depends=ext.depends
)
```

cpython/Lib/distutils/command/build_ext.py

Instalação

```
$> python setup.py install
```

```
running install
running build
running build_ext
building 'module' extension
creating build
creating build/temp.linux-x86_64-2.7
{gcc compila o módulo}
running install_lib
copying build/lib.linux-x86_64-2.7/module.so -> /path/site-packages
running install_egg_info
Removing path/module-1.0-py2.7.egg-info
Writing /path/site-packages/module-1.0-py2.7.egg-info
```

```
$> python setup.py --help
```

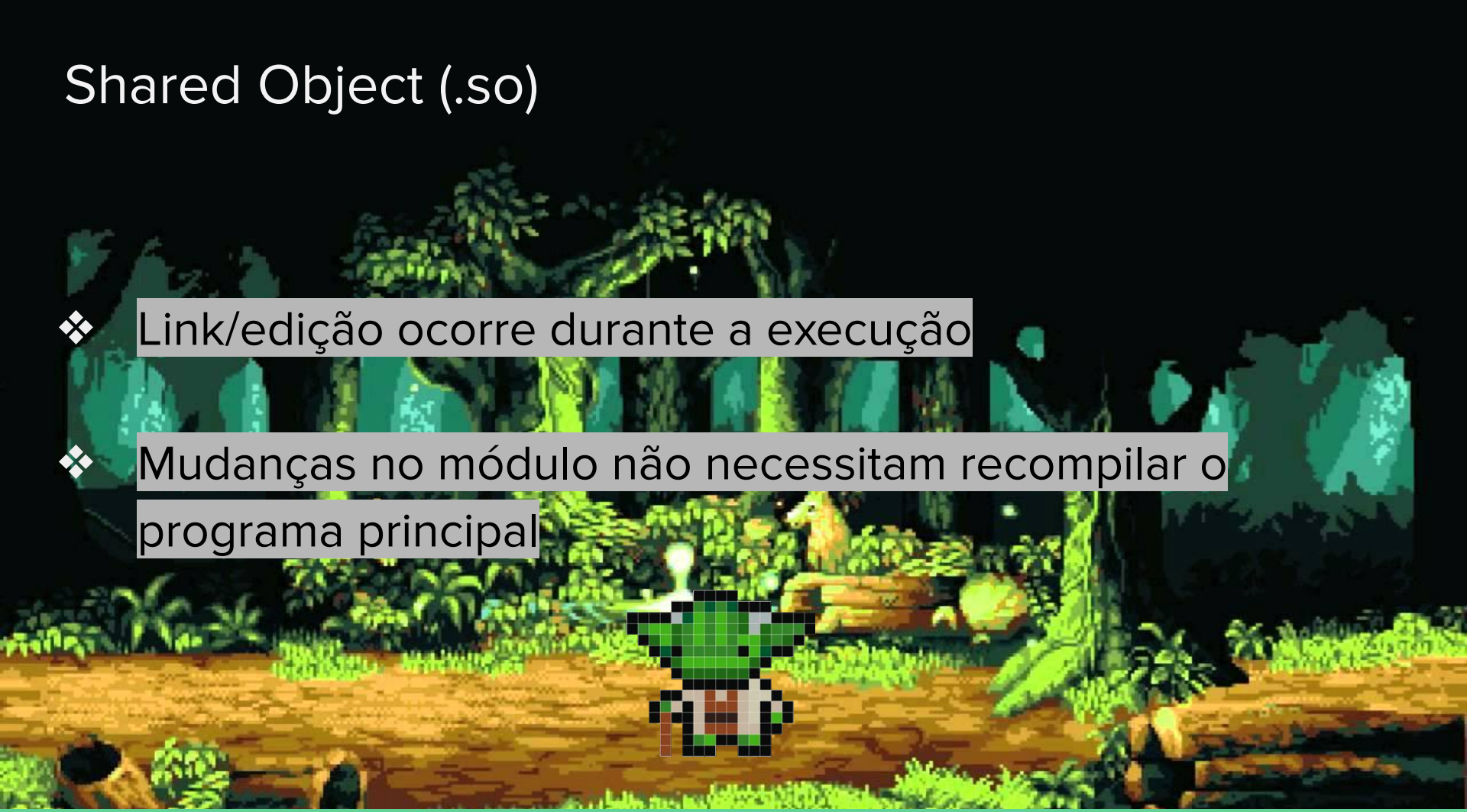
Instalação

{gcc compila o módulo}

```
gcc -pthread -fno-strict-aliasing -fmessage-length=0 -grecord-gcc-switches -O2
-Wall -D_FORTIFY_SOURCE=2 -fstack-protector-strong -funwind-tables
-fasynchronous-unwind-tables -g -DNDEBUG -fmessage-length=0
-grecord-gcc-switches -O2 -Wall -D_FORTIFY_SOURCE=2 -fstack-protector-strong
-funwind-tables -fasynchronous-unwind-tables -g -DOPENSSL_LOAD_CONF -fwrapv
-fPIC -I/usr/include/python2.7 -c hello.c -o
build/temp.linux-x86_64-2.7/hello.o
creating build/lib.linux-x86_64-2.7
gcc -pthread -shared
build/temp.linux-x86_64-2.7/hello.o -L/usr/lib64
-lpython2.7 -o
build/lib.linux-x86_64-2.7/module.so
```

```
$> man gcc
```

Shared Object (.so)



- Link/edição ocorre durante a execução
- Mudanças no módulo não necessitam recompilar o programa principal

Hands On

Usando o módulo


```
$> ipython

In [1]: from module import hello

In [2]: hello()
Out[2]: 'Hello Pythonista'

In [3]: help(hello)
```

Carregamento dinâmico


```
In [1]: from sys import modules
In [2]: modules['module']
---------------------------------------------------------------------
KeyError                          Traceback (most recent call last)
<ipython-input-2-f0b257567ce0> in <module>()
----> 1 modules['module']
KeyError: 'module'

In [3]: from module import hello

In [4]: modules['module']
Out[4]: <module 'module' from
'/path/to/lib/python2.7/site-packages/module.so'>
```

Link / edição dinâmica

```
$> ldd /path/to/lib/python2.7/site-packages/module.so
linux-vdso.so.1 (0x00007ffe0bd94000)
libpython2.7.so.1.0 =>/usr/lib64/libpython2.7.so.1.0(0x00007f72ca71b000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f72ca4fe000)
libc.so.6 => /lib64/libc.so.6 (0x00007f72ca15f000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f72c9f5b000)
libutil.so.1 => /lib64/libutil.so.1 (0x00007f72c9d58000)
libm.so.6 => /lib64/libm.so.6 (0x00007f72c9a52000)
/lib64/ld-linux-x86-64.so.2 (0x000055df9d3ae000)
```

```
$> man ldd
```

Tabela de símbolos

```
$> objdump -t /path/to/lib/python2.7/site-packages/module.so | grep text
0000000000000690 l    d  .text  0000000000000000          .text
0000000000000690 l     F .text  0000000000000000          deregister_tm_clones
00000000000006d0 l     F .text  0000000000000000          register_tm_clones
0000000000000720 l     F .text  0000000000000000          __do_global_dtors_aux
0000000000000760 l     F .text  0000000000000000          frame_dummy
0000000000000790 l     F .text  0000000000000015          hello
00000000000007b0 g     F .text  000000000000001d          initmodule
```

```
$> man objdump
```

Experimentos

Um módulo em C para calcular **média**, **moda** e **mediana** de uma lista de **inteiros**

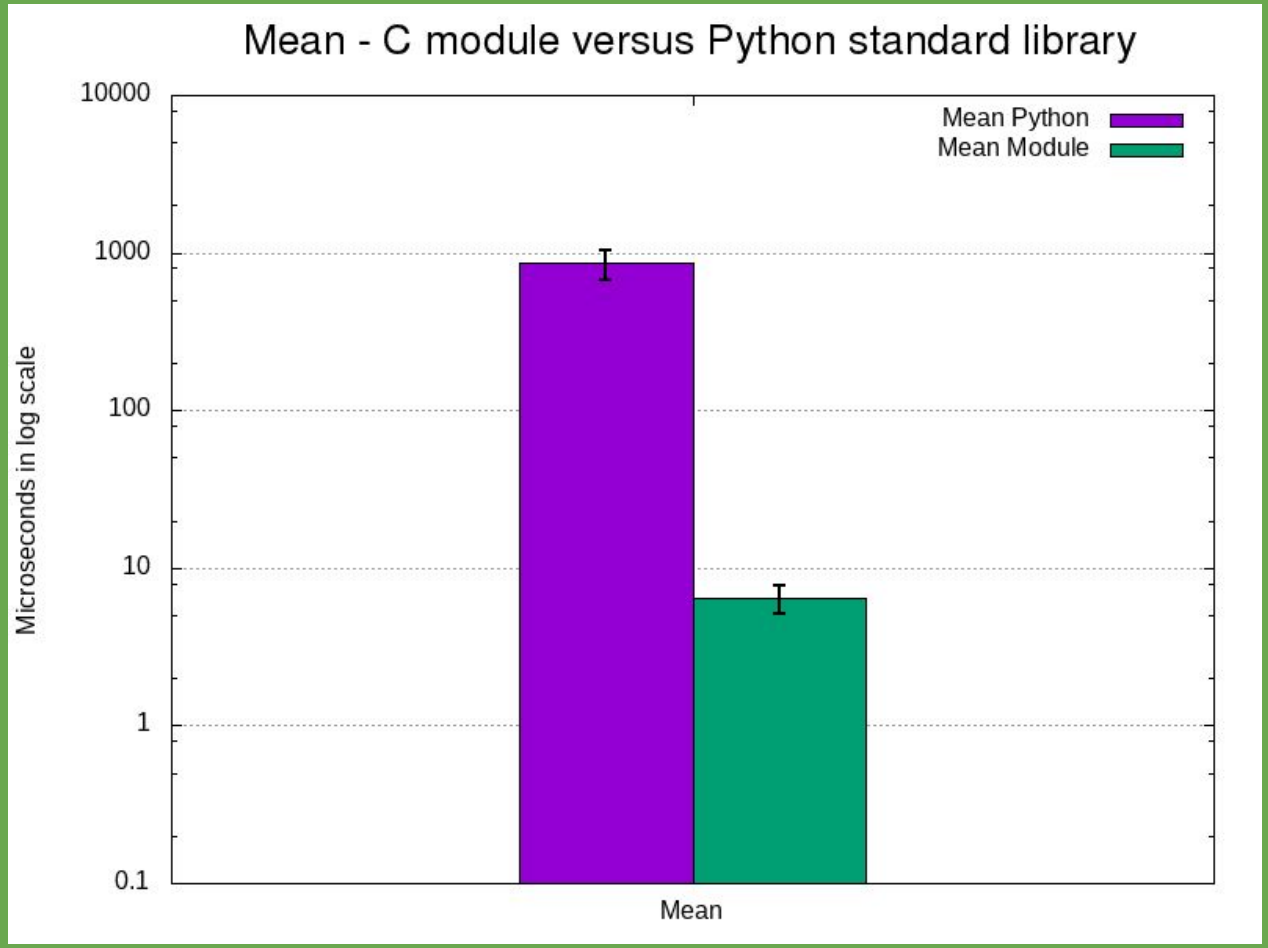
Experimentos

- **1000** listas com **1000** inteiros aleatórios variando entre 1 e 1000
- **Computar o tempo** de execução do **módulo** e da **biblioteca padrão**
- Gerar **histograma** com tempos **médios**

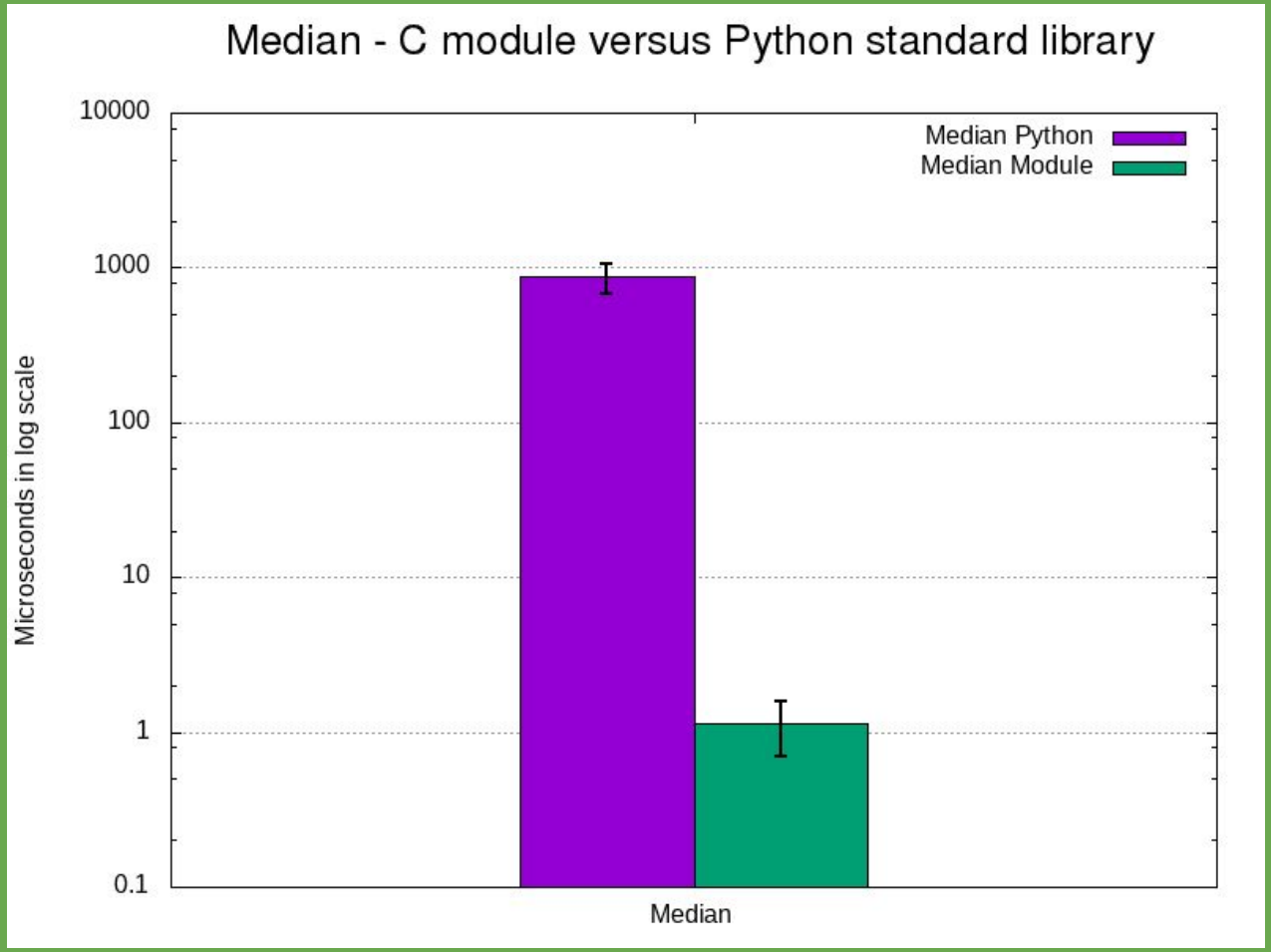
Experimentos

```
for i, input_list in enumerate(inputs):
    # ...
    collect_mean(input_list)
    collect_mode(input_list)
    collect_median(input_list)
```

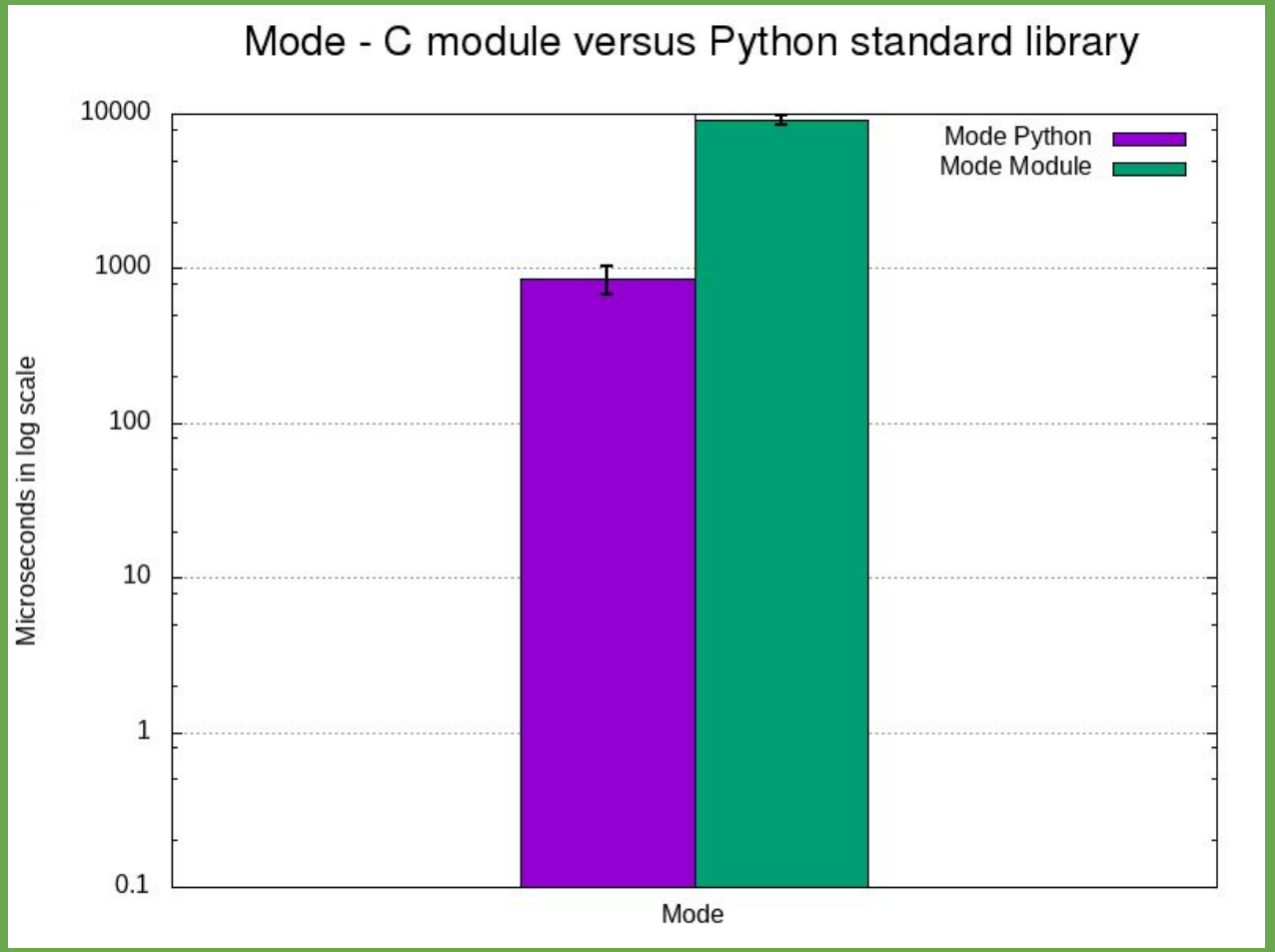
Experimentos



Experimentos



Experimentos



Algoritmo

Porque o módulo em C foi mais lento do que a biblioteca padrão para o cálculo de moda?

Algoritmo

Módulo

```
for(int i = 0; i < seq_size; i++) {
    ...
    for(int j = 0; j < seq_size; j++) {
        ...
        if(_PyLong_AsInt(item_i) == _PyLong_AsInt(item_j)) {
            count++;
        }
    }
    ...
}
```

Algoritmo

Módulo

```
for(int i = 0; i < seq_size; i++) {
    ...
    for(int j = 0; j < seq_size; j++) {
        ...
        if(_PyLong_AsInt                                    j)) {
            count++;
        }
    }
    ...
}
```

O(n²)

Algoritmo

Biblioteca padrão

Utiliza uma *Heap Queue* ou *Priority Queue*

```
                    0
          1                   2
      3       4           5       6
   7     8   9    10    11    12  13   14
 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
```

https://github.com/python/cpython/blob/master/Lib/heapq.py#L33

Algoritmo

Biblioteca padrão

Utiliza uma *Heap Queue* ou *Priority Queue*

O(n log n)

1
3 6
7 8 9 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

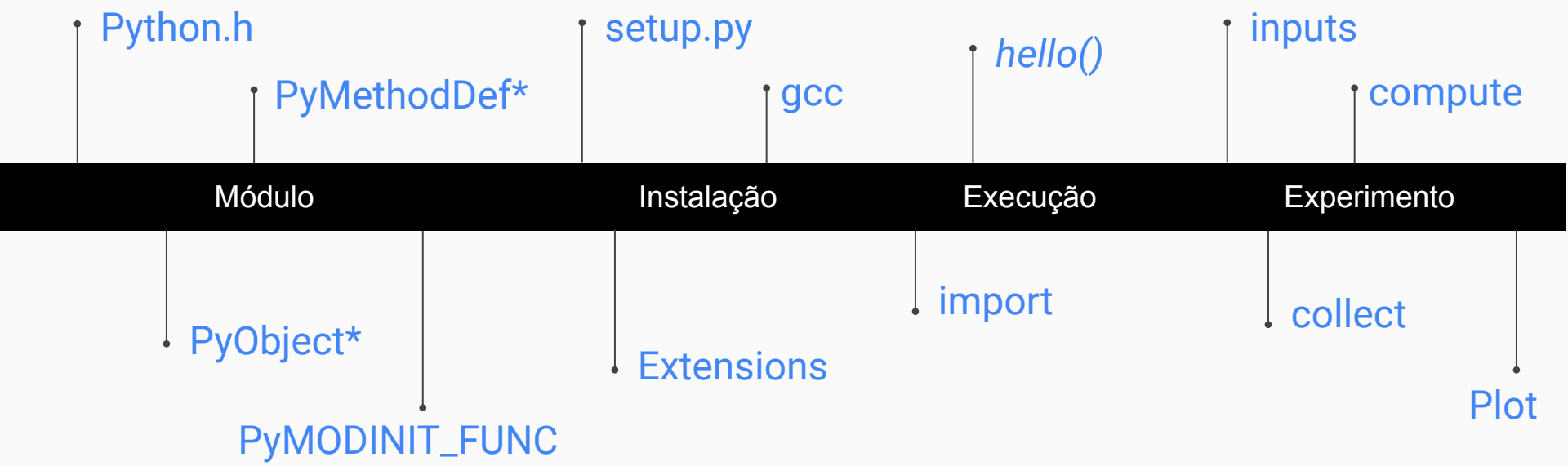
https://github.com/python/cpython/blob/master/Lib/heapq.py#L33

Ferramentas

Simplicidade e automação do processo

- https://github.com/swig/**swig**
- http://**cython**.org/
- https://www.ics.uci.edu/~dock/manuals/**sip**/sipref.html

Resumindo


Python.h
PyMethodDef*
setup.py
gcc
hello()
inputs
compute
Módulo
Instalação
Execução
Experimento
PyObject*
PyMODINIT_FUNC
Extensions
import
collect
Plot

Material de consulta

https://blog.pantuza.com/tutoriais/criando-modulos-python-atraves-de-extensoes-em-c

https://github.com/pantuza/cpython-modules

GoDaddy

30% OFF na compra de novos produtos GoDaddy.

https://br.godaddy.com/promos/coupon-promo-codes?isc=gt02br20

GoDaddy

Desconto de **R$30** na compra de domínio .dev

https://br.godaddy.com/tlds/dev-domain?isc=GDSOR19&countryview=1&currencyType=brl


Dúvidas?
https://blog.pantuza.com
https://github.com/pantuza
https://twitter.com/gpantuza