Contribuyendo a Python

Alimenta a la víbora

Rafael Santos Pérez y David Suárez Pascal

Plan de vuelo para el taller

- Introducción
- Preparándonos para contribuir código a Python

- Más allá del código: Todas las maneras de contribuir
- Retos de familiarización con CPython

Lunch

- Retos de familiarización con CPython

- Espacio abierto de contribución

Introducción

Sobre nosotros

David Suárez

Rafael Santos

Introducción

Comunidad

python.org/community

- PSF
  - *Promover, proteger e impulsar Python*
- Grupos
  - Grupos de Usuarios Locales, SIG
- Listas de correo
- Py*Slack*ers, Freenode IRC

Introducción

Comunidad

- Eventos
  - PyCon US
  - PyCon LATAM, SciPy LATAM
  - DjangoCon, EuroPython, PyData, Pycon .*

Introducción

Python vs CPython

Introducción

Punto de partida

devguide.python.org

Introducción

Familiarízate con git

devguide.python.org/gitbootcamp

1. Haz un *fork* del repositorio oficial de python/CPython con tu cuenta de *Github*
2. Clona tu nuevo repositorio

```
git clone git@github.com:<usuario>/cpython.git
```

Introducción

Familiarízate con git

devguide.python.org/gitbootcamp

3. Agrega el repositorio oficial como upstream

```
git remote add upstream git@github.com:python/cpython.git
```

Introducción

Familiarízate con git

devguide.python.org/gitbootcamp

3. Asegúrate de que la configuración global de git incluye tus datos correctos

```
git config --global user.name "Tu Nombre"
git config --global user.email <usuario>@users.noreply.github.com
```

Introducción

Familiarízate con git

devguide.python.org/gitbootcamp

3. Crea un *branch* a partir del *master branch*

```
git checkout -b <nombre> master
```

Introducción

Prepara el proyecto para desarrollo

devguide.python.org/setup/#compile-and-build

- Configura y Compila con debug symbols

```
./configure --with-pydebug

make -s -j2
```

Introducción

¡Listo!

¡Ya estás corriendo Python 3.9!

Bienvenid@ al futuro

```
$ ./python.exe
Python 3.9.0a0 (heads/ccoss2019-dirty:3f43cef, Sep 10 2019, 19:33:
[Clang 6.0 (clang-600.0.54)] on darwin
Type "help", "copyright", "credits" or "license" for more informat
>>>
```

Primer Reto

Agrega tu propio mensaje en el encabezado del intérprete

```
$ ./python.exe
Python 3.9.0a0 (heads/ccoss2019-dirty:3f43cef, Sep 10 2019, 19:33:
[Clang 6.0 (clang-600.0.54)] on darwin
[CCOSS 2019 - Python Contributor Workshop] <==
Type "help", "copyright", "credits" or "license" for more informat
>>>
```

Primer Reto

Pasos a seguir:

1. Encuentra la función principal del intérprete de CPython
2. Encuentra la función que imprime el encabezado
3. Modifica el mensaje
4. Recompila el proyecto

```
make -s -j2
```

Segundo Reto

Modifica el Zen de Python

```
>>> import this
The Zen of Python, by Tim Peters and Rafael Santos

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
```

Segundo Reto

Pasos a seguir:

1. Identifica el módulo de la librería estándar que implementa el Zen
2. Identifica el método de cifrado utilizado por el módulo
3. Modifica el texto o agrega código para incluir los cambios

```
Hint: from codecs import encode
```

Introducción

Issue Tracker (BPO)

Bugs.Python.Org

Plan de vuelo para el taller

3. Formas de contribución:
    - Reporte de bugs
    - Triaje: Revisa, comenta y comprueba los reportes
    - Documentación: completa, corrige y traduce los documentos de Python
    - Evangelización: ¿qué más podemos hacer para favorecer la adopción de Python?
    - Conviértete en un core developer de Python

Formas de contribuir

Buscando bugs

- Encuentra alguno
- Corre las pruebas de Python:

```
./python -m test -j2 -rW

./python -m test -h
```

- Si tienes "suerte", verifica el módulo que arroja el error y busca a ver si nadie lo ha reportado en bpo.

Formas de contribuir

Un poco de entomología

- Si no tienes "suerte", siempre puedes checar la lista de bugs reportados en bpo:
  - "Easy" issues
  - Issues with patch
  - Newcomer friendly

Formas de contribuir

¿En qué consiste el triaje (*triage*)?

Revisión de *issues* y *pull requests (PR)*

- Renombrar PRs
- Revisar PRs
- Ayudar a los contribuidores
- Notificar a los *core developers*
- Etiquetar apropiadamente: *DO-NOT-MERGE, expert-asyncio, invalid, needs backport to X.Y, OS-X, skip issue, skip news, sprint, stale, type-bugfix, type-documentation, type-enhancement, type-performance, type-security, type-tests*.

Formas de contribuir

¿Cómo ayudar a documentar Python?

Para empezar

- Python es famoso por su énfasis en la legibilidad y en la documentación del código: indentación, *docstrings*, tutorial, biblioteca estandar, guía del desarrollador.
- Issues relacionados con la documentación

Formas de contribuir

Documentación de Python

1. Documentación para usuarios del lenguaje: tutorial, biblioteca estándar
   - Incluida con el código fuente de CPython.
2. Documentación para desarrolladores:
   - https://github.com/python/devguide

Formas de contribuir

Traduciendo la documentación

Según la PEP 545, cada traducción completa:

- Tiene una etiqueta apropiada para cada idioma (ISO 639-1): *es, pt-br, fr, de, ...*
- Es distribuida bajo CC0
- Se alberga en: https://github.com/python/python-docs-{LANGUAGE_TAG}
- Debe incluir: tutorial/, library/stdtypes y library/functions
- Está disponible en https://docs.python.org/{LANGUAGE_TAG}/{VERSION_TAG}/

Formas de contribuir

Traduciendo la documentación

¡https://docs.python.org/es/3.7/ genera un error HTTP 404!

Formas de contribuir

Traducción al español

https://github.com/raulcd/python-docs.es/

Formas de contribuir

Evangelizando con Python

1. Ayuda resolviendo las dudas de los principiantes
2. Comparte tu conocimiento en Meetups, Python Days, Pycons, etc.
3. No participes en flame wars sobre lenguajes
4. Enfócate en las fortalezas del lenguaje: consistencia, legibilidad, documentación, potencia, ...

Formas de contribuir

El camino del core developer

1. Desarrollador de Python/Usuario de CPython
2. Contribuidor
3. *Triager*
4. *Core developer*
5. Mentor

Recursos adicionales

- CPython Source Code Guide
- Documentations of CPython internals

Plan de vuelo para el taller

4. Manos a la obra:
    1. Encuentra un bug susceptible de ser exterminado
    2. Crea una rama para atacar el problema
    3. Implementa y documenta tu solución
    4. Haz un *pull request*
    5. Ajusta tu solución de acuerdo a la retroalimentación
    6. Repite desde el paso 3