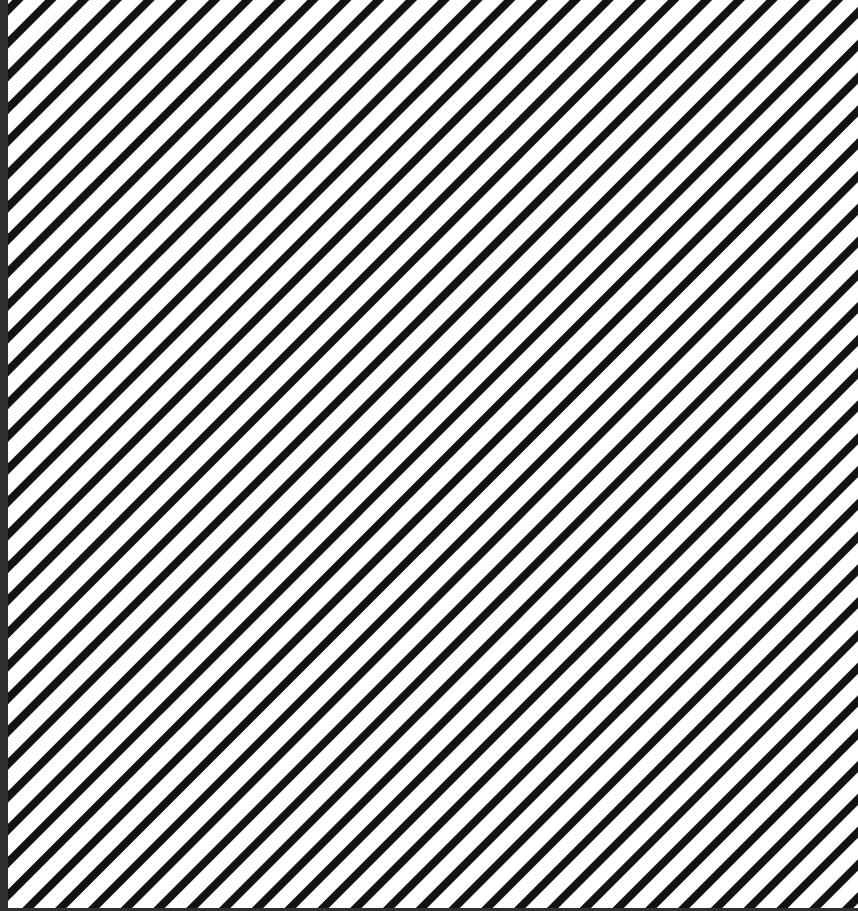


MIGRANDO A PYTHON 3

Sin perder la cabeza



S O B R E M I

Jonathan Vallejo Muñoz
Team Leader - Lendingfront

Hobbies: Scuba Diver & Trekking



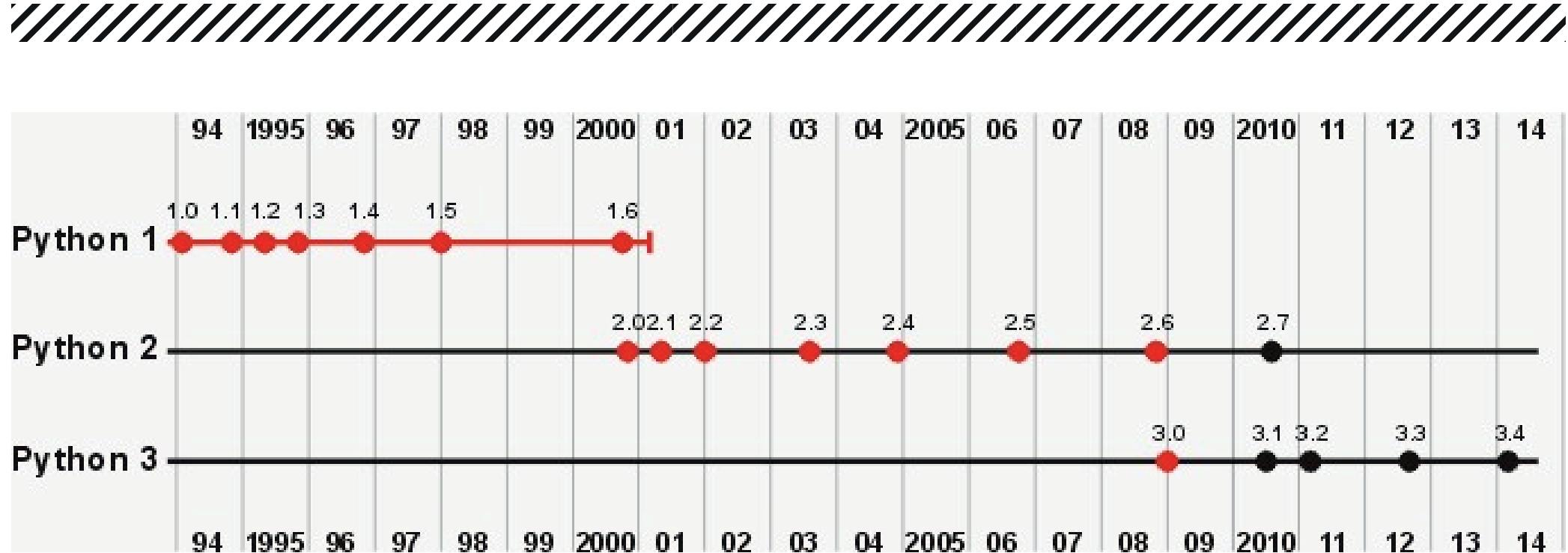


A G E N D A



1. Python Timeline
2. Diferencias entre Python 2 y 3
3. Motivación
4. Estrategias de migración
5. Lecciones aprendidas
6. Conclusiones
7. Preguntas y respuestas

P Y T H O N T I M E L I N E



PYTHON 2 OR PYTHON 3?

WTF!

http://www.burntfinger.net

“We knew Python’s popularity was on an upward trend and we hoped its growth would continue (which it thankfully has :)). But this also meant that if we were going to fix any design mistakes and help continue the language’s popularity, we needed to do it now rather than later.”

By: BRETT CANNON

<https://snarky.ca/why-python-3-exists/>

Diferencias entre Python 2 y 3

Cambios en el Core

1. Tipos de datos
2. Operadores
3. Sintaxis
4. Módulos de la librería estándar

Nuevas características

Print Function

Python 2

print

print 1

print 1, 2

print 1, 2,

print >>sys.stderr, 1, 2, 3

Python 3

print()

print(1)

print(1, 2)

print(1, 2, end=' ')

print(1, 2, 3, file=sys.stderr)

Integer Division

Python 2

`3 / 2 = 1`

`3 // 2 = 1`

`3.0 / 2 = 1.5`

Python 3

`3 / 2 = 1.5`

`3 // 2 = 1`

`3.0 / 2 = 1.5`



Exceptions

Python 2

```
try:  
    import mymodule  
except ImportError, e  
    pass
```

Python 3

```
try:  
    import mymodule  
except ImportError as e:  
    pass
```

Types Module Constants

Python 2	Python 3
types.UnicodeType	str
types.StringType	bytes
types.DictType	dict
types.IntType	int
types.LongType	int
types.ListType	list

Has Key

Python 2

a_dictionary.has_key('PapayaWhip')

a_dictionary.has_key(x) or
a_dictionary.has_key(y)

Python 3

'PapayaWhip' in a_dictionary

x in a_dictionary or y in a_dictionary

Renamed Reorganized Modules

Python 2

`import urllib`

`import urllib2`

`import urlparse`

`import robotparser`

`import httplib`

`import Cookie`

`import cookielib`

Python 3

`import urllib.request, urllib.parse, urllib.error`

`import urllib.request, urllib.error`

`import urllib.parse`

`import urllib.robotparser`

`import http.client`

`import http.cookies`

`import http.cookiejar`

Explicit Relative Imports

```
chardet/
|
+--__init__.py
|
+--constants.py
|
+--mbcharsetprober.py
|
+--universaldetector.py
```

Python 2

```
import constants
```

```
from mbcharsetprober import
MultiByteCharSetProber
```

Python 3

```
from . import constants
```

```
from .mbcharsetprober import MultiByteCharsetProber
```



Filter - Map

Python 2

```
filter(a_function, a_sequence)
```

Python 3

```
list(filter(a_function, a_sequence))
```

```
list(filter(a_function, a_sequence))  No Change
```

Python 2

```
map(a_function, 'PapayaWhip')
```

Python 3

```
list(map(a_function, 'PapayaWhip'))
```

```
map(None, 'PapayaWhip')
```

```
list('PapayaWhip')
```

M O T I V A C I Ó N

Nuevas características

Mejoras en el Core

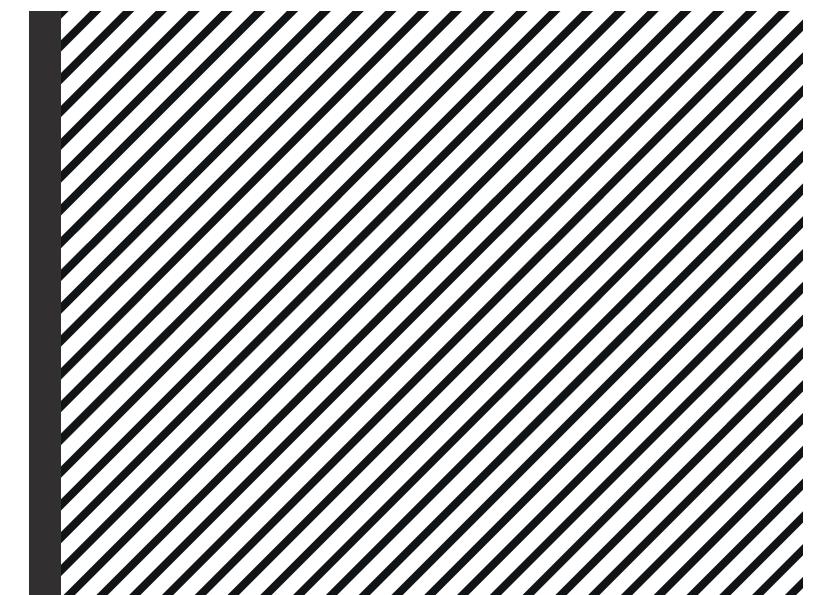
Soporte de librerías externas

Documentación

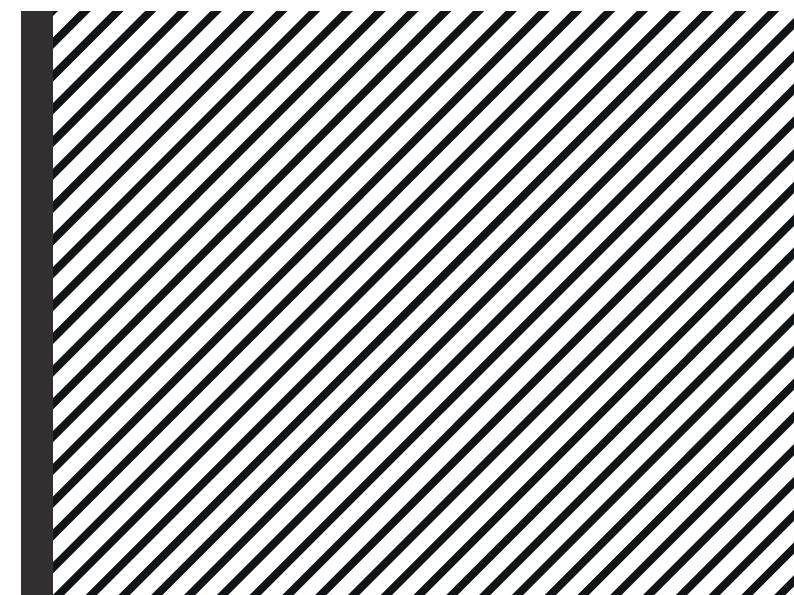
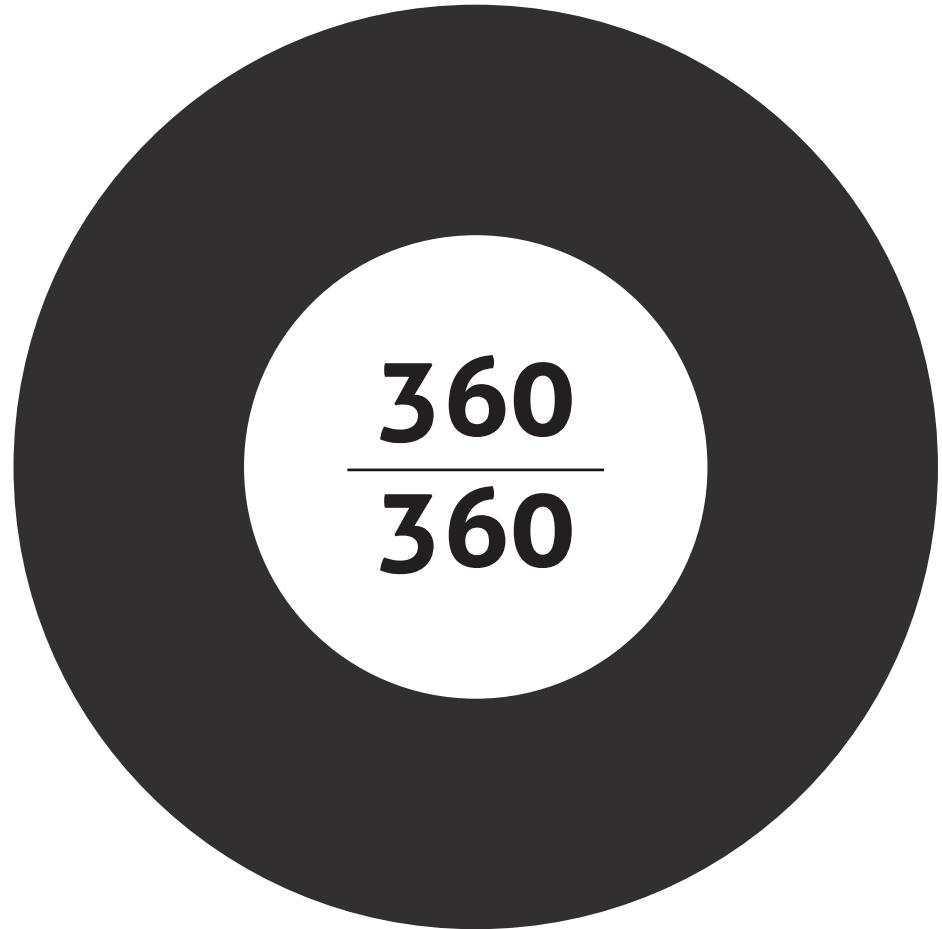
Seguridad (Soporte)

Costos

Menor interés en python 2 para los devs



**Python 3 support graph for the 360
most popular Python packages**





Python Software ⚡ @ThePSF · 8 Aug 2018

We're excited to see all the latest functionality and features of Python 3.7 available on Google App Engine. cloud.google.com/blog/products/...
@GCPcloud



Introducing App Engine Second Generation runtime...

Python 3.7 is available today in beta on the App Engine standard environment.

cloud.google.com

3

99

284



Guido van Rossum ⚡

@gvanrossum

Follow

Replying to @ThePSF @GCPcloud

Your last excuse to stay on Python 2 is gone!

7:43 AM - 10 Aug 2018

96 Retweets 455 Likes



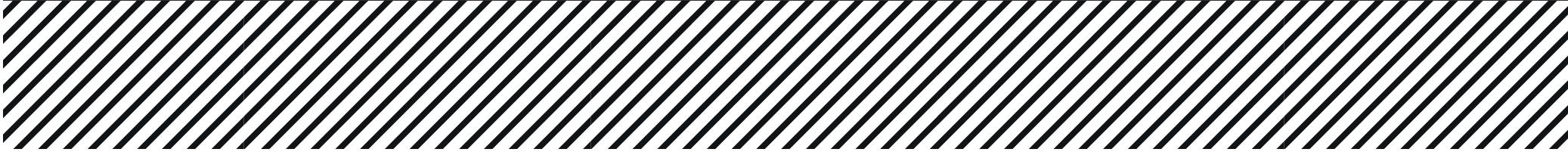
6

96

455

¡Usa Python 3 en vez de 2.7!





Python 2.7 will retire in...

0

Years

0

Months

21

Days

22

Hours

22

Minutes

35

Seconds



Outstanding move



Estrategias de migración

Fuerza Bruta

1. Parar el proceso de desarrollo
2. Dedicar todo el equipo a la migración
3. Hasta no tener todo el código soportado en Python 3 no mezclar
4. Desplegar en producción (Switch a Python 3)

Ideal para bases de código pequeñas y equipos pequeños

Busqueda Completa Iterativa

1. Dedicar parte del equipo a la migración
2. No parar el proceso de desarrollo
3. Crear una rama alterna con la migración y no mezclar hasta tener todo el código soportado en Python 3.
4. Actualizar la rama todos los días, refactorizando el código mezclado para que soporte Python 3
5. Desplegar en producción (Switch a Python 3)

**Ideal para bases de código pequeñas - medianas.
Equipos Pequeños**

Divide y vencerás

1. Dedicar parte del equipo a la migración.
2. Escoger una sección de código (Microservicio, libreria, Controller...)
3. Soportar Python 2 y Python 3 al mismo tiempo.
4. Mezclar y desplegar en Producción.
5. Iterar sucesivamente hasta que el código sea compatible en ambas versiones.
6. esplegar en Producción (Switch a Python 3)

**Ideal para bases de código medianas - grandes.
Equipos medianos - grandes**



Lecciones aprendidas

(Mi versión de Python)

1. Revisa cuidadosamente tus librerías para saber cual versión de Python 3 soportan
2. ¿Cuál versión de Python 3 representa menos esfuerzo/costo ?
3. ¿Seguridad vs Estabilidad ? Revisar Changelog de las versiones

Utilidades

1. Caniusepython3
2. Pip-check

Lecciones aprendidas

(Python 2 y 3 __future__)

Future es un módulo que permite hacer uso de comportamientos de Python 3 en código ejecutado en Python 2

```
from __future__ import division

1 / 2 --> 0.5
4 / 2 --> 2.0
1 // 2 --> 0
4 // 2 --> 2
```



Lecciones aprendidas

(Python 2 y 3 con six)

Six al igual que __future__, permite escribir código compatible para ambas versiones. Contiene un set de utilidades para casos donde la implementación en ambas versiones es excluyente. Ejemplo:

Python 2: **dictionary.iteritems()**

Python 3: **dictionary.items()**

Six: **six.iteritems()**

Lecciones aprendidas (Tox)

1. Tox es un administrador de ambientes virtuales que te permite ejecutar tests en diferentes versiones de Python
2. Agregar pruebas unitarias en Python 2 y Python 3 para los cambios más peligrosos (La división de Enteros, Unicode Strings, refactorizaciones de librerías, entre otros)
3. <https://tox.readthedocs.io/en/latest/>

Lecciones aprendidas

(Automatizar la migración)

1. **2to3** es un módulo que permite la automatización de la migración a Python 3 por medio de la aplicación de una serie de **Fixers**

```
def greet(name):
    print("Hello, {0}!".format(name))
print("What's your name?")
name = input()
greet(name)
```



```
def greet(name):
    print "Hello, {0}!".format(name)
print "What's your name?"
name = raw_input()
greet(name)
```



Lecciones aprendidas

(Culturizando tu equipo)

1. La unión hace la fuerza
2. Meetings de sincronización con los líderes de equipos
3. Entre más personas se involucren en el proceso (QA + Devs + Devops + Team leaders + Product Owners + ...) mas rápido se alcanzará la meta.
4. Crear material de apoyo: Sliders, Videos, Meetings internos
5. Automatizar la validación de soporte de Python 3 (**pylint**) - (**tox**)
6. Genera un plan de riesgos e impacto con tu equipo de QA

Conclusiones

Coordinar entre diferentes equipos de trabajo es clave

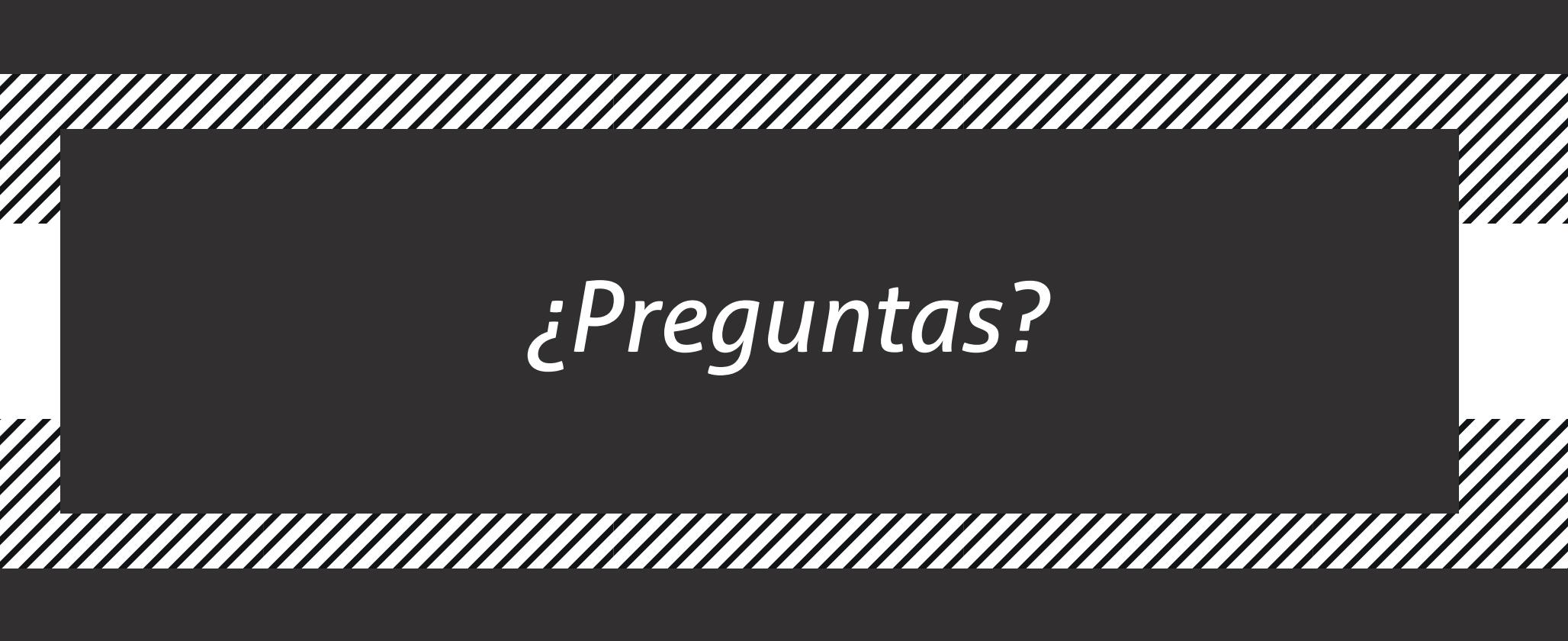
El proceso de migración de Python 3 es más fácil de lo que piensas.

Existen muchas herramientas disponibles para hacer el proceso de migración más sencillo (2to3, future, tox, entre otros)

Cualquier problema que enfrentes en la migración, probablemente alguien ya lo ha enfrentando.

El apoyo de la comunidad es enorme.

Migrar a Python 3 no es opcional.



¿Preguntas?



LENDINGFRONT

We are Hiring!

*Si eres un apasionado por el desarrollo de software en Python, te interesa el crecimiento profesional y el aprendizaje constante, envía tu hoja de vida al correo electrónico **carrers@lendingfront.com***

¡Gracias!

https://es.wikipedia.org/wiki/Historia_de_Python

<https://link.springer.com/content/pdf/bbm%3A978-1-4302-2416-7%2F1.pdf>

<https://pythonclock.org/>

<http://py3readiness.org/>

<https://mail.python.org/pipermail/tutor/2008-March/060886.html>

<https://tox.readthedocs.io/en/latest/>

<https://docs.python.org/2/library/2to3.html>

<https://docs.python.org/3/howto/pyporting.html>

<https://python-future.org/quickstart.html>

<https://snarky.ca/why-python-3-exists/>