Linux Málaga

@linux_malaga

www.linux-malaga.org

Taller de Python

Juan Miguel Taboada Godoy

@centrologic_es

http://linkedin.com/user/centrologic

Juan José Soler Ruiz

@soleronline

http://es.linkedin.com/in/soleronline

Bienvenido - Welcome - Witam

Centrologic

Bio

Juan Miguel Taboada Godoy (1980 - …)

1996 – Primer ordenador y primera LAN (coaxial)

1999 – Universidad de Málaga y **Linux Málaga**

2001 – Investigación en la UMA

- Cluster de computación masiva
- Servidores y hosting
- Mercados bursátiles
- Beca Neurociencia en New York

2005 – Axargua (Adquisición de datos industriales)

2008 – Pontgrup Correduría de Seguros

2011 – Bética fotovoltáicas (Adquisición de datos para Red Eléctrica España)

2012 – Centrologic

Juan José Soler Ruiz

2001 – CFGS Administración Sistemas Informáticos

2003 – Primer premio en el concurso “Javier Benjumea”
- Adquisición de datos con Visual Basic

2003 – Montaje y configuración de:
“Cluster Heterogéneo de Computadoras”
bajo SO Red Hat 7.2.

2005 – STEA Telemática
- Desarrollador y analista de sofware

2007 – Primer CRM en PHP

2010 – Bética fotovoltáicas
- Administrador de sistemas
- Desarrollador y analista de software

2010 – Opositometro (Desarrollador web)

2012 – Centrologic

Centrologic

Linux Málaga

python™

Bio

Ángel José Martin Sánchez (1989 - …)

2008 – Universidad de Málaga

2010 – Curso desarrollo de aplicaciones en Android

2012 – Curso peritaje informáticos y análisis forense

2013 – Asesores Locales S.L.

- Desarrollo web

2013 – SecureKids (Socio fundador y desarrollador)

- Grupo Deide S. Coop. And.

2015 – Soluciones Salutic S.L.

- Desarrollo web

- QA (Responsable de Calidad)

2016 – Centrologic

Roberto Antonio Becerra García (1986 - …)

2001 – Primer programa en Pascal

2004 – Medalla de Oro en concurso nacional de Computación de Cuba e integrante de la Preselección nacional a participar en la IOI

2009 – Finalista del concurso internacional de Programación para universitaroios ACM-ICPC

2010 – Construcción de una plataforma de procesamiento de movimientos oculadores

2011 – Profesor de Arquitectura de Ordenadores e Inteligencia artificial en la Universidad de Holguin

2012 – Doctorado en la UMA en procesamiento de movimientos oculares

2017 – Centrologic

Centrologic

Linux Málaga

python™

Linux Málaga

Historia

Junio 1998 (Campus Party '98)

Mayo 1999 (Legal)

Noviembre 2003 (Final juvenil)

Y nuevo comienzo

Meetup

674 inscritos y 17 eventos celebrados

5 eventos programados en 2017

Contacto

@linux_malaga

www.linux-malaga.org

Centrologic

Linux Málaga

python™

Open South Code

Edicción 2017 - MAYO

open south code

Viernes día 5

- 3 salas (aforo 25 personas/sala)

Sábado día 6

- 40 charlas (8 charlas por sala)
- Stand específico
- Mesas de exposiciones
- Posible catering

Año 2016:

16 charlas y 5 talleres

Año 2017:

40 charlas planificadas

Centrologic

Linux Málaga

python™

Centrologic


Centrologic
python™
django
HTML
BOOTSTRAP


Centrologic
CODENERIX
python™
django
BOOTSTRAP


python™
Centrologic
Linux Málaga
python™

¿Quién? ¿qué? ¿por qué? ¿cuando? ¿cómo?

Guido van Rossum

Centrum Wiskunde & Informatica
(Paises Bajos)

Finales de los '80

Humoristas Monty Python

1991 :: 0.9.0 (POO)

1994 :: 1.0 (funcional)

2000 :: 1.6 y 2.0

2008 :: 2.6 y 3.0 (Unicode)

2010 :: 2.7

2014 :: 3.4

Lenguaje interpretado

Sintaxis favorece la lectura

Multiplataforma

Tipado dinámico

Pitónico => ZEN

Centrologic

Linux Málaga

python™

Algunos elementos del lenguaje

Números: 0, 1, 2, 2.3445, 4+3j

Cadenas: “Hola mundo”

Listas: [“Hola”, 123]

Tuplas: (“Hola”, 123)

Diccionarios: {“Hola”: “Mundo”}

Otros: None / True / False

< <= > >= == != is is not

not or and

if elif else while for break continue

abs() int() float() complex()

- + * / % ** divmod()

In not in s[x:y] len() min() max()

def pass return class import

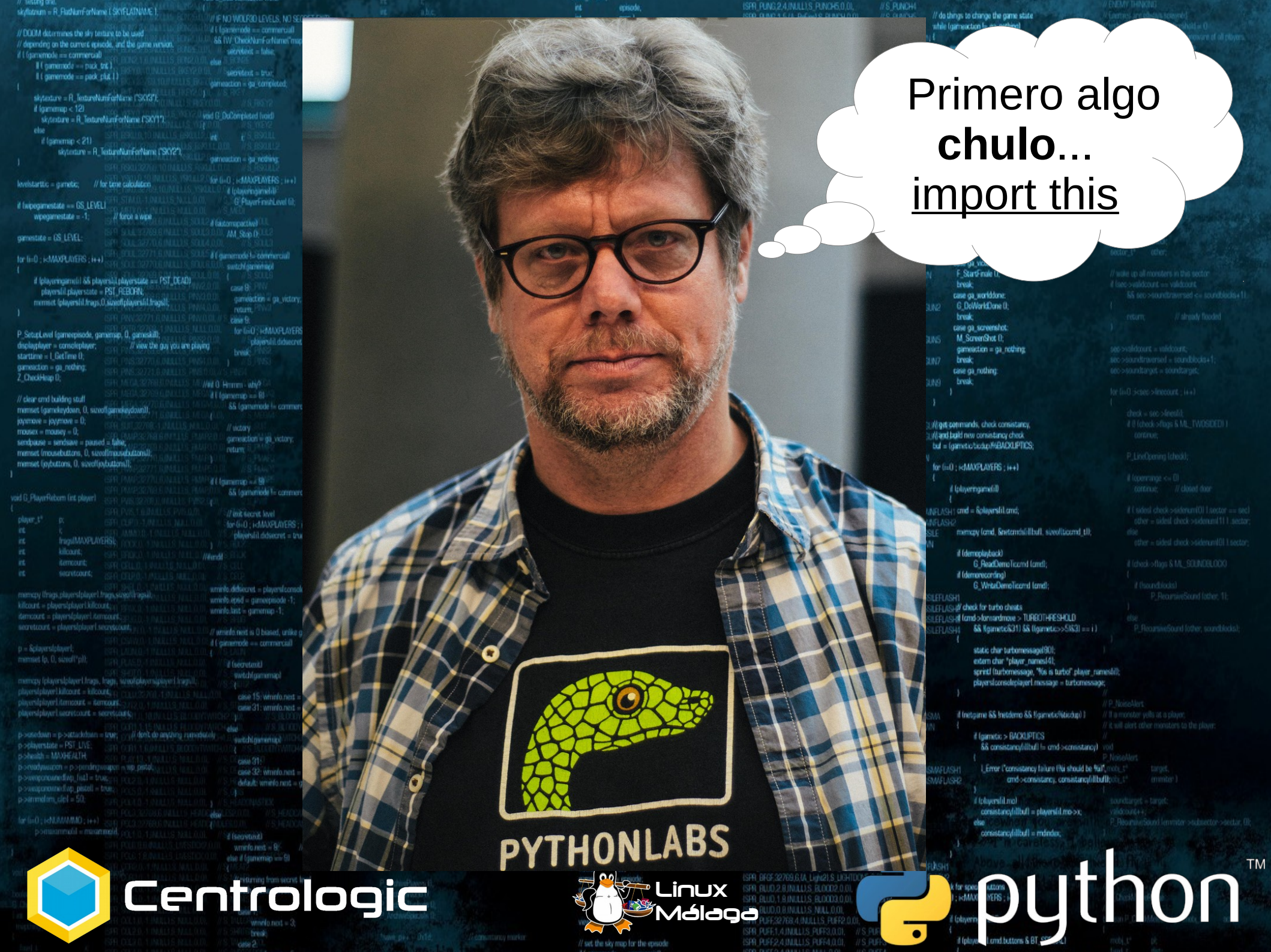
sin cos pi ceil exp floor sqrt

Prácticamente todo es un objeto

Centrologic

Linux Málaga

python™



Centrologic

Linux Málaga

python™

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

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.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

¿Mejor en español?

Centrologic

Linux Málaga

python™

```
>>> import this
El Zen de Python, por Tim Peters

Hermoso es mejor que feo.
Explícito es mejor que implícito.
Simple es mejor que complejo.
Complejo es mejor que complicado.
Plano es mejor que anidado.
Disperso es mejor que denso.
La legibilidad cuenta.
Los casos especiales no son suficientemente especiales
                                                como para romper las reglas.
Aunque lo pragmático gana a la pureza.
Los errores nunca deberían dejarse pasar silenciosamente.
A menos que se silencien explícitamente.
Cuando te enfrentes a la ambigüedad, rechaza la tentación de adivinar.
Debería haber una "y preferiblemente sólo una" manera obvia de hacerlo.
Aunque puede que no sea obvia a primera vista a menos que seas holandés.
Ahora es mejor que nunca.
Aunque muchas veces nunca es mejor que *ahora mismo*.
Si la implementación es difícil de explicar, es una mala idea.
Si la implementación es sencilla de explicar, puede que sea una buena idea.
Los espacios de nombres son una gran idea — ¡tengamos más de esos!
>>>
```

Grábatelo a fuego

Centrologic

Linux Málaga

python™

Ejercicio 1: el texto "ofuscado"

```
python2.7 -c "import base64; exec(base64.b64decode('
eJydj8tOxDAMRff9iks33Q4rpErsgA2P0cD8gNu4kqW0Do4zYvh6
MpQFFTsiS0mcc0+Shs3Ubh8oZm7czn2DOmROao7JU5Sh4Y+Rk68n
K360wt/bZLI42iNjouiEWEsGYxNCt8a79o/3+fx2eAr/FgdG96Oo
cpmwqK/5fpPVoNg/XuE+O+Uaz65IZASOGItlbX/jrzwWtkB4L4xR
Z+Hlk3C5OeP6pt/tNvSLQuNJAmd4AQdxtfrUk1pdbcg7zqlUbNEM
MZ7rRHCdyS6dslR9N4gZ5fqZL1bYgF8=').decode('zlib'))"
```

Centrologic

Linux Málaga

python™

Ejercicio 1: el texto "ofuscado"

```
>>> import base64; exec(base64.b64decode('
Ejydj8tOxDAMRff9iks33Q4rpErsgA2P0cD8gNu4kqW0
Do4zYvh6MpQFFTsiS0mcc0+Shs3Ubh8oZm7czn2DOmRO
ao7JU5Sh4Y+Rk68nK360wt/bZLI42iNjouiEWEsGYxNC
t8a79o/3+fx2eAr/FgdG960ocpmwqK/5fpPVoNg/XuE+
O+Uaz65IZASOGItlbX/jrzwWtkB4L4xRZ+Hlk3C5OeP6
pt/tNvSLQuNJAmd4AQdxtfrUk1pdbcg7zqlUbNEMMZ7r
RHCdyS6dslR9N4gZ5fqZL1bYgF8=').decode('zlib'))
Todo OK! Estas listo para el curso
Recuerda que comienza a las 17:00
No olvides tu editor favorito
Despues nos iremos a tomarnos unas 'birras'
>>>
```

Centrologic

Linux Málaga

python™

1) g_max(1, 2) = 2

2) g_len([1, 1, 2, 2]) = 4

3) g_max(1, 3, 2) = 3

4) vocal(“a”) = True

5) traduce(rövarspråket)
consonante*2 + 'o' en medio

“this is fun” =
“tothohisos isos fofunon"

Centrologic

Linux Málaga

python™

6) suma([1, 2, 3, 4]) = 10

7) multi([1, 2, 3, 4]) = 24

8) voltea(“Es una prueba”)
= “abeurp anu sE”

9) pali(“radar”) = True si
radar es un palíndromo

10) esta_en(x, lista) = True
si x está en la lista

Centrologic

Linux Málaga

python™

11) solapa(lista1, lista2)
= True si lista1 y lista2 tienen al menos un elemento en común

12) nchars('a' , 3) = “aaa”

13) histograma([2, 5, 3, 4])
XX
XXXXX
XXX
XXXX

Centrologic

Linux Málaga

python™

14) max(...n...):
    max(3, 5, 4) = 5
    max(3, 5, 4, 1, 6) = 6

15) histochar(“abbabcb
    dbabdbdbababababcb
    cbab")
    a: XXXXXXX
    b: XXXXXXXXXXXXXXX
    c: XXX
    d: XXX

16) rot(s, 13) de Julio César

Centrologic

Linux Málaga

python™

```
18) class calculadora:
        def __init__(self,x,y):
        def sumar(self):
        def restar(self):
        def multiplicar(self):
        def dividir(self):

19) class alumno:
        def __init__(self, nombre,
                apellido, [notas] ):
        def __str__(self):
        def ponnota(self, nota):
        def notamedia(self):
        def ver_ficha(self):
```

Centrologic

Linux Málaga

python™

17) adivina()
Python elije num aleatorio del 1 al 100 y el usuario pregunta y el sistema responde indicando si está frío o caliente.

18) ahorcado()

19) tresenraya()

20) g2048() ó sudoku()

Centrologic

Linux Málaga

python™

DUDAS

Centrologic

Linux Málaga

python™

Linux Málaga

@linux_malaga

www.linux-malaga.org

Muchas GRACIAS

Juan Miguel Taboada Godoy

http://www.centrologic.com

@centrologic_es

http://linkedin.com/user/centrologic

Juan José Soler Ruiz

@soleronline

http://es.linkedin.com/in/soleronline

Thank you - Dziękuję

Centrologic