

Linux
Málaga
@linux_malaga
www.linux-malaga.org



Taller avanzado de Python



Juan Miguel Taboada Godoy
@juanmitaborda
<https://www.linkedin.com/in/juanmitaborda>



Juan José Soler Ruiz
@soleronline
<http://es.linkedin.com/in/soleronline>



Roberto Becerra García
@idertator
<https://www.linkedin.com/in/idertator>

Bienvenido - Welcome - Witam



Centroologic

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 fotovoltaicas ([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 software](#)

2007 – Primer [CRM en PHP](#)

2010 – Bética fotovoltaicas
- [Administrador de sistemas](#)
- [Desarrollador y analista de software](#)

2010 – Opositometro ([Desarrollador web](#))

2012 – Centrologic



Bio



Centrologic



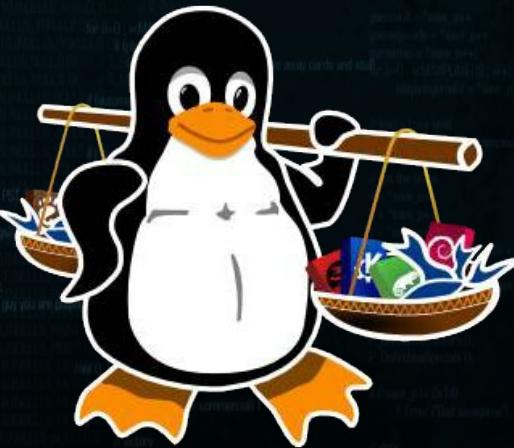
Linux
Málaga



python



Linux Málaga



Meetup

674 inscritos y 17 eventos celebrados

5 eventos programados en 2017

Historia

Junio 1998 (Campus Party '98)

Mayo 1999 (Legal)

Noviembre 2003 (Final juvenil)

Y nuevo comienzo

Linux Málaga

Contacto

@linux_malaga

www.linux-malaga.org



Centrologic



Linux
Málaga



python™

Open South Code

Edición 2017 - MAYO

Viernes día 5

- 3 salas (aforo 25 personas/sala)

Sábado día 6

- 40 charlas (8 charlas por sala)

- Stand específico

- Mesas de exposiciones

- Possible catering



open
south
code

Año 2016:

16 charlas y 5 talleres

Año 2017:

40 charlas planificadas



CentroLogic



python

django





Centrologic

CODENEX



python



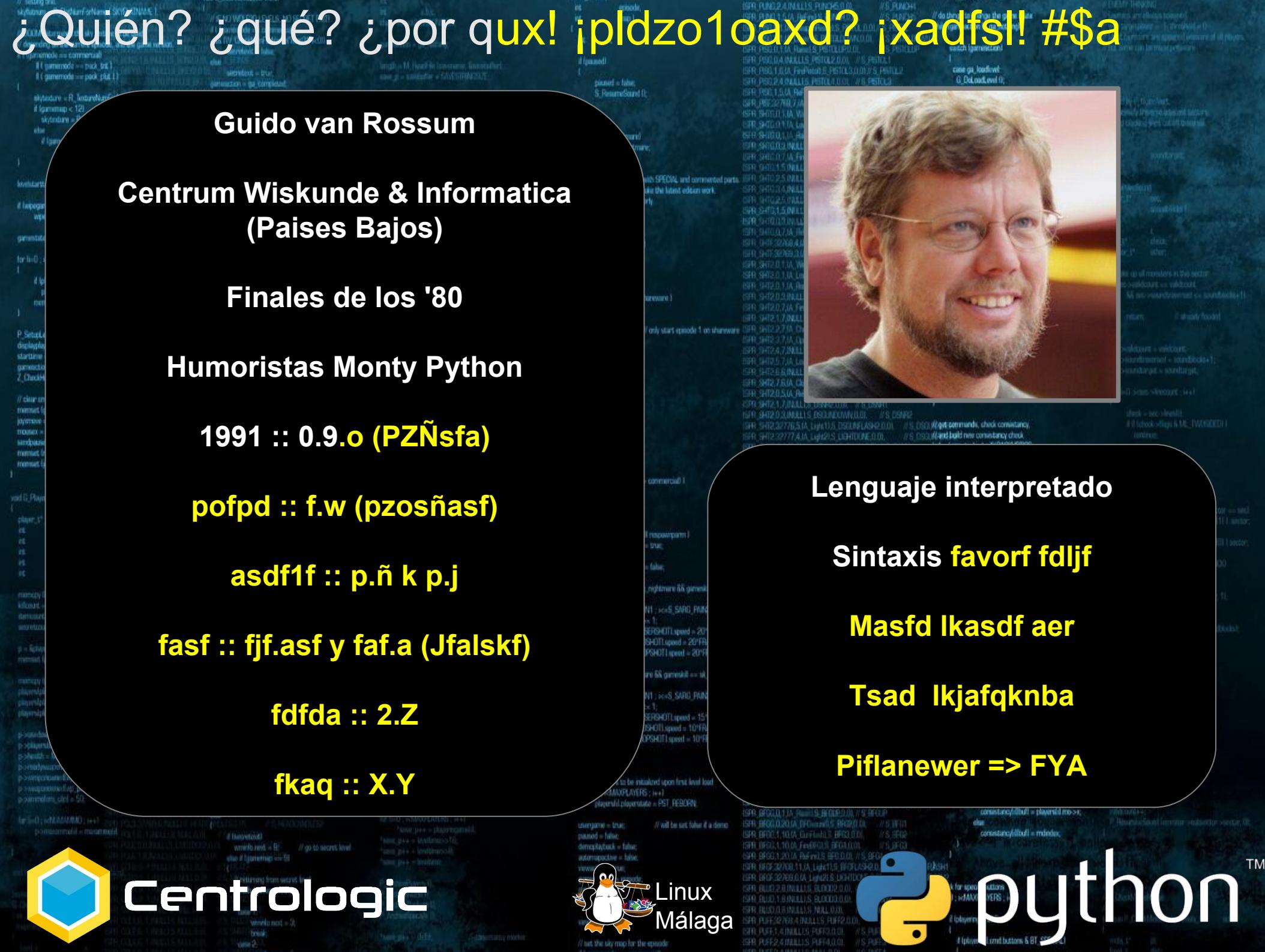


Centrologic



Linux
Málaga

python



¿Quién? ¿qué? ¿por qux! ¡pldzo1oaxd? ¡xadfs! #\$a

Guido van Rossum

Centrum Wiskunde & Informatica
(Paises Bajos)

Finales de los '80

Humoristas Monty Python

1991 :: 0.9.0 (PZNsfa)

pofpd :: f.w (pzosñasf)

asdf1f :: p.ñ k p.j

fasf :: fjf.asf yfaf.a (Jfalskf)

fdfda :: 2.Z

fkaq :: X.Y



Lenguaje interpretado

Sintaxis favorf fdljf

Masfd lkasdf aer

Tsad lkjafqknba

Piflanewer => FYA



python™



Centrologic

¡Vamos al lío!



Centrologic



The logo for Linux Málaga features a stylized white and orange cartoon character on the left, holding a wooden balance scale. The scale's pan contains several colorful icons representing different software or technologies. To the right of the character, the word "Linux" is written in a large, bold, black sans-serif font, and "Málaga" is written below it in a slightly smaller version of the same font.



python

Programación funcional (1)

Problema

lista = [1, 4, 3, 2, 5, 7, 2, 9, 8, 7, 8, 1]

Crear una función que retorne los cubos de los elementos impares de la lista.

Programación funcional (2)

Solución tradicional

```
resultado = []
```

```
for x in lista:
```

```
    if x % 2 != 0:
```

```
        resultado.append(x * x * x)
```

```
print resultado
```



Centrologic



python™

Programación funcional (3)

Solución tradicional

```
resultado = []
```

```
for x in lista:
```

```
    if x % 2:
```

```
        resultado.append(x ** 3)
```

```
print resultado
```



Centrologic



Linux
Málaga



python™

Programación funcional (4)

```
def impares(lista):
```

```
    for x in lista:
```

```
        if x % 2:
```

```
            yield x
```

```
resultado = []
```

```
for x in impares(lista):
```

```
    resultado.append(x**3)
```

```
print resultado
```

Utilizando generadores

Programación funcional (5)

```
def impares(lista):
```

```
    for x in lista:
```

```
        if x % 2:
```

```
            yield x
```

```
def cubo(x):
```

```
    return x ** 3
```

```
print map(cubo, impares(lista))
```

Utilizando map

Programación funcional (6)

Utilizando map+lambda

```
def impares(lista):
```

```
    for x in lista:
```

```
        if x % 2:
```

```
            yield x
```

```
print map(lambda x: x ** 3, impares(lista))
```

Programación funcional (7)

Utilizando
comprepción
de listas

```
def impares(lista):
```

```
    for x in lista:
```

```
        if x % 2:
```

```
            yield x
```

```
print [x ** 3 for x in impares(lista))]
```

```
Programación funcional (8)
```

Utilizando comprensión de listas con filtrado

```
print [x ** 3 for x in lista if x % 2]
```



Centrologic



Clases y métodos (9)

Clase básica

```
class A(object):
    attr = None
    def __init__(self):
        self.attribute = True
    def foo(self,x):
        print "executing foo(%s,%s)"%(self,x)
    @staticmethod
    def static_foo(x):
        print "executing static_foo(%s)"%x a=A()
```

Herencia (10.1)

```
class A(object):
    def a(self):
        print "Soy a y estoy en la clase A"

    def b(self):
        print "Soy b y estoy en la clase A"

    def c(self):
        print "Soy c y estoy en la clase A"

class A1(object):
    def c(self):
        print "Soy c y estoy en A1"
        super(A1, self).c()
```

Herencia (10.2)

```
class B(A):
```

```
    def b(self):
```

```
        print "Soy b y estoy en la clase B"
```

```
class C(A1, B):
```

```
    def c(self):
```

```
        print "Soy c y estoy en la clase C"
```

```
        super(C, self).c()
```

```
a = A()
```

```
b = B()
```

```
c = C()
```



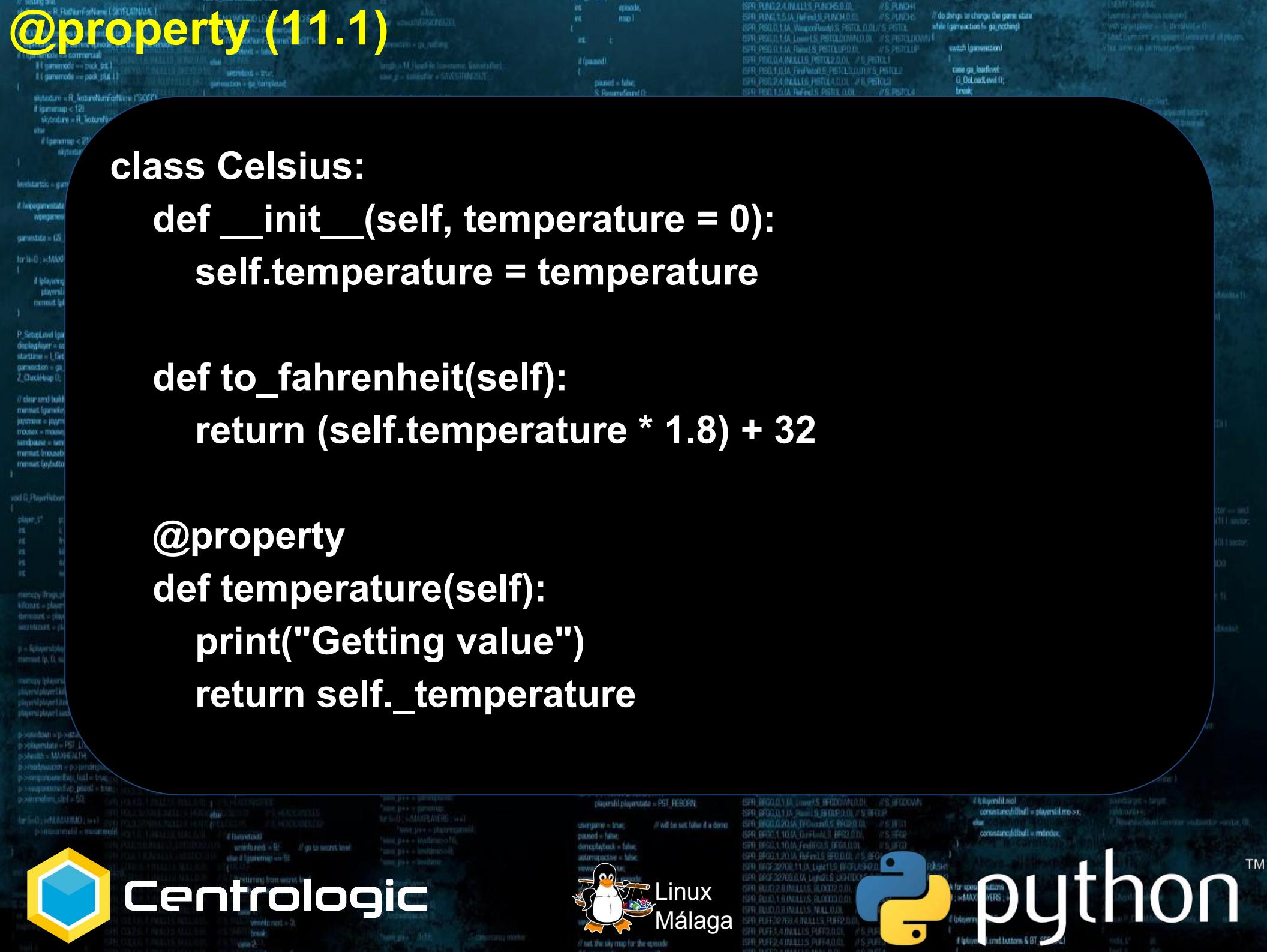
Centrologic



Linux
Málaga



python™



@property (11.2)

@temperature.setter

```
def set_temperature(self, value):
```

```
    if value < -273:
```

```
        raise ValueError("Temperature below -273 is not possible")
```

```
    print("Setting value")
```

```
    self._temperature = value
```

```
c = Celsius()
```

```
print c.temperature
```

```
c.temperature = 2
```

```
print c.temperature
```

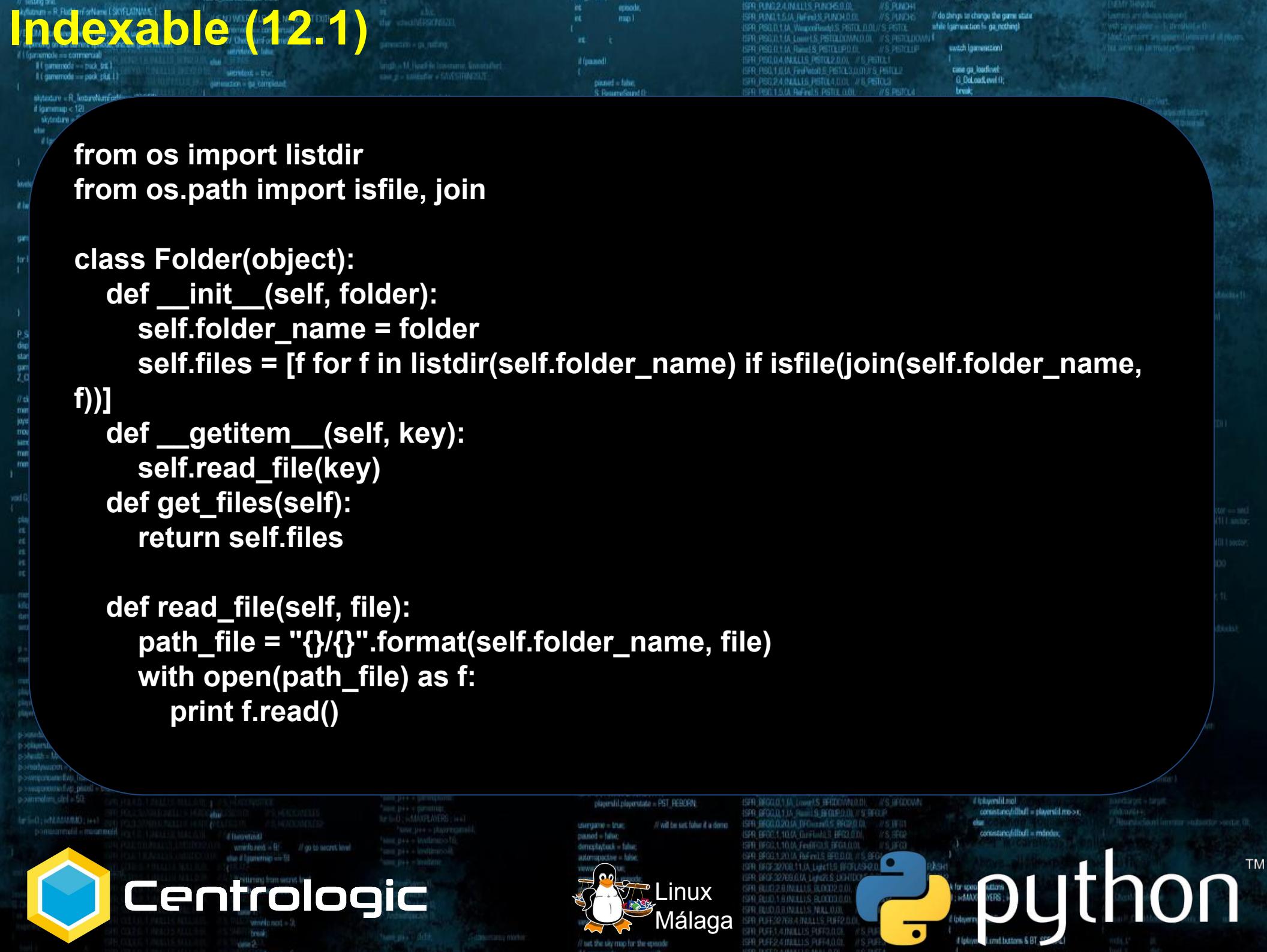
```
print c.to_fahrenheit()
```



Centrologic



python™



Indexable (12.1)

```
from os import listdir
from os.path import isfile, join

class Folder(object):
    def __init__(self, folder):
        self.folder_name = folder
        self.files = [f for f in listdir(self.folder_name) if isfile(join(self.folder_name, f))]

    def __getitem__(self, key):
        self.read_file(key)

    def get_files(self):
        return self.files

    def read_file(self, file):
        path_file = "{}{}".format(self.folder_name, file)
        with open(path_file) as f:
            print f.read()
```



Centrologic



TM
python

Indexable (12.2)

ruta = '/tmp'

directorio = Folder(ruta)

print directorio.get_files()

directorio['jojo.txt']

DUDAS

```

if ! NO_WOLFSO_LEVELS, NO_SECRET_EXIT
if (gameMode == commercial)
  if (WV_CheckWinOrName("magDT")>0)
    winOrLose = false;
else
  winOrLose = true;
generation = ga_completed;

if (G_DuCompleted(wv))
  if (wv < 100)
    if (ga_completed > 0)
      generation = ga_nothing;
  else
    generation = ga_victory;
  return;
}

if (wv>0) //MAXPLAYERS ;++)
  if (wv==ga_completed)
    G_PlayerFinalLevel(); // take over

if (wv==ga_completed)
  AM_Skip();
else
  if (ga_completed == commercial)
    switch(gamemode)
    {
    case 0:
      generation = ga_victory;
      return;
    case 1:
      for (i=0; i<MAXPLAYERS ; ++i)
        playfield.defected = true;
      break;
    }
}

if (0 < generation < 10)
  if (gamemode == 0)
    if (ga_completed < commercial)
    {
    // exit
    generation = ga_victory;
    return;
    }

  if (gamemode == 10)
    if (ga_completed < commercial)
    {
    // init secret level
    for (i=0; i<MAXPLAYERS ; ++i)
      playfield.defected = true;
    }
  endif;
endif;
}

if (wv!=defected = playfield.consolePlayerDefect)
  if (wv>0)
    gamemode = gameMode - 1;
  else
    gamemode = gameMode + 1;

  // amode not a 0 based, unlike gamemode
  if (gamemode == commercial)
  {
    if (secretLevel)
      switch(gamemode)
      {
      case 15: wmode.net = 30; break;
      case 31: wmode.net = 31; break;
      default: wmode.net = 0;
      }
    else
      switch(gamemode)
      {
      case 31: break;
      case 32: wmode.net = 15; break;
      default: wmode.net = gamemode;
      }
  }
  else
    if (secretLevel)
      switch(gamemode)
      {
      case 15: wmode.net = 30; break;
      case 31: wmode.net = 31; break;
      default: wmode.net = 0;
      }
    else
      switch(gamemode)
      {
      case 31: break;
      case 32: wmode.net = 15; break;
      default: wmode.net = gamemode;
      }
  }
  else
    if (secretLevel)
      switch(gamemode)
      {
      case 15: wmode.net = 30; break;
      case 31: wmode.net = 31; break;
      default: wmode.net = 0;
      }
    else
      switch(gamemode)
      {
      case 31: break;
      case 32: wmode.net = 15; break;
      default: wmode.net = gamemode;
      }
  }
}

```

length = 11; Head-Resonance, Resonance;
name_p++ = Resonance + "GAVESTRINGS2025";

if skip the description field
resonate = (int)0x1337000000000000L;
skip_length = version_Nr * 1000000000L;
if (length >= skip_length)
 return;
name_p++ = VERGINIZE;

resonate = "Name_p++";
resonate_p++ = "Name_p++";
resonate_p++ = "Name_p++";
resonate_p++ = "Name_p++";
resonate_p++ = "Name_p++";

cards and stuff
if (skip_length <= length)
 playresonate = "Name_p++";

if load a Name File
Q_InName (resonate, resonance, resonate);

I DIDN'T O



THE BUG

resonate_name_p = resonance; GAVESTRINGS2025;
name_p++ = GAVESTRINGS2025;
resonate_name2025 = resonance_name2025;
skip_length_name2025 = version_Nr * 1000000000L;
resonate_name_p = name_p, VERGINIZE;
name_p++ = VERGINIZE;

"Name_p++ = resonate;
"Name_p++ = resonance;
"Name_p++ = resonate;
for i=0;i<MAXPLAYERS;i++
 "Name_p++ = playresonate;
 "Name_p++ = testresonate[i];
 "Name_p++ = testresonate[i];
 "Name_p++ = testresonate;

ogic



Centrologic



Linux
Málaga



python



Linux
Málaga
@linux_malaga
www.linux-malaga.org



Muchas
Gracias



Juan Miguel Taboada Godoy
@juanmitaboada
<https://www.linkedin.com/in/juanmitaboada>



Juan José Soler Ruiz
@soleronline
<http://es.linkedin.com/in/soleronline>



Roberto Becerra García
@idertator
<https://www.linkedin.com/in/idertator>

Thank you - Dziękujemy



Centroologic