



**Linux**  
**Málaga**  
 @linux\_malaga  
 www.linux-malaga.org



# Introducción a Python



**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>

**Bienvenido - Welcome - Witam**



**Centrologic**





## Juan Miguel Taboada Godoy ( 1980 - ... )

1996 – Primer ordenador y primera LAN (coaxial)

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

2001 – Grupo de investigación GEB.uma.es (4 años)

2002 – Asociación cultural **Málaga Wireless**

2003 – Beca en Neurociencia en SUNY (Julio-Agosto)  
Teleruta (Ministerio de Fomento – 2 años)

2004 – Globatic S.C. (1 año)

2005 – Autónomo:

- Nace **Centrologic**
- Polonia (2 años)
- **Likindoy**

2008 – Responsable comunicaciones en PontGrup

2011 – Bética Fotovoltáicas

2012 – **SAFECLON** y SCRUM/KANBAN

2013 – **MBA Executive** y nace **Bioengineering**

## Juan José Soler Ruiz

2001-2003 – CFGS Administración Sistemas Informáticos

2003 – Primer premio en el concurso “Javier Benjumea”

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

2005-2012 – STEA Telemática

2007-2009 – Primer CRM en PHP

2010-2011 – Administrador de BBDD / Programador  
Web en Bética Fotovoltáicas

2010-2012 – Opositometro

2012-.... – **Centrologic**

2013 – Dailymarkets

2013-2014 – CRM en Python/Django

2014-... – **Bioengineering Software ( SAFECLON )**



¿Quién? ¿qué? ¿por qué? ¿Cuándo? ¿cómo?

**Guido van Rossum**

**Centrum Wiskunde & Informatica  
(Países 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**



TM



# 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







# TU turno

Me gusta el:  
*import this*



```
... = R_RadJunkForName (SKYFLDLEVEL, NO SECTORS)
// (M) set the sky map for the episode
// depending on the current episode, and the game version
if (gameepisode == commercial)
    secretlevel = false;
else if (gameepisode == pack_bot)
    secretlevel = true;
else if (gameepisode == pack_plus1)
    gameaction = ga_completed;

skystexture = R_TextureForName ("SKYFLD");
if (gamemap < 12)
    skystexture = R_TextureForName ("SKY11");
else
    if (gamemap < 21)
        skystexture = R_TextureForName ("SKY21");
    gameaction = ga_nothing;

levelstarttic = gametic; // for time calculation
// (M) set the player's level
if (levelgamestate == GS_LEVEL)
    wpgamestate = -1; // force a wipe
gamestate = GS_LEVEL;

for (i=0; i<MAXPLAYERS; i++)
    if (playergame[i] && player[gamestate] == PST_DEAD)
        player[gamestate] = PST_REBORN;
memset (player[gamestate], 0, sizeof player[gamestate]);

P_SetupLevel (gameepisode, gamemap, 0, gameskill);
displayplayer = consoleplayer;
starttime = (int)time (0);
gameaction = ga_nothing;
Z_CheckHeap (0);

// clear and building stuff
memset (gamekeydown, 0, sizeof gamekeydown);
joystick = joystick = 0;
mousex = mousey = 0;
sendpause = sendwave = paused = false;
memset (mousebuttons, 0, sizeof mousebuttons);
memset (joybuttons, 0, sizeof joybuttons);

void G_PlayerReborn (int player)
{
    player[1] = R;
    for (i=0; i<MAXPLAYERS; i++)
        if (i != player)
            frags[i][MAXPLAYERS] = 0;
    kilcount = 0;
    democount = 0;
    secretcount = 0;

    memcopy (frags, player[gamestate].frags, sizeof frags);
    kilcount = player[gamestate].kilcount;
    democount = player[gamestate].democount;
    secretcount = player[gamestate].secretcount;

    p = &player[gamestate];
    memset (p, 0, sizeof *p);

    memcopy (player[gamestate].frags, frags, sizeof frags);
    player[gamestate].kilcount = kilcount;
    player[gamestate].democount = democount;
    player[gamestate].secretcount = secretcount;

    p->sectortic = p->starttic + 1;
    p->playerstate = PST_LIVE;
    p->health = MAXHEALTH;
    p->readyweapon = p->pendingweapon = wp_pistol;
    p->weaponowned[wp_fist] = true;
    p->weaponowned[wp_pistol] = true;
    p->armorindex = 0;

    for (i=0; i<NUMAMMO; i++)
        p->ammomax[i] = ammomax[i];
}

```

```
... do things to change the game state
while (gameaction != ga_completed)
{
    case ga_completed:
        G_DeConstruct (0);
        INFLASH;
        INFLASH;
        break;
    case ga_victory:
        F_StartTitle (0);
        break;
    case ga_worlddone:
        G_DeWorldDone (0);
        break;
    case ga_screenshot:
        M_Screenshot (0);
        gameaction = ga_nothing;
        break;
    case ga_nothing:
        break;
}

// wake up all monsters in this sector
if (sect->validcount == validcount)
    && (sect->soundtraced == soundblocks + 1)
        return; // already flooded

sect->validcount = validcount;
sect->soundtraced = soundblocks + 1;
sect->soundtarget = soundtarget;

for (i=0; i<sect->storecount; i++)
{
    check = sect->store[i];
    if (check->flags & M_TWIXED)
        continue;
    P_LinkOpening (check);
    continue; // closed door
}

INFLASH; end = &player[0].end;
INFLASH;
SECRET memcopy (end, &secret[0].end, sizeof secret);
else
    if (demoplayback)
        G_ReadDemoCommand (end);
    if (demorecording)
        G_WriteDemoCommand (end);
}

SLEEPFLASH; check for turbo cheats
SLEEPFLASH; if (end->armorindex > TURBOHEALTH)
SLEEPFLASH; && (gameepisode > 3) && (gameepisode < 5) == 1)
{
    static char (turbo message) (0);
    extern char *player_names[4];
    sprintf (turbo message, "%s is turbo! player: %s",
            player[gamestate].message = turbo message;
}

if (intgame && !end->secret && !gameepisode)
{
    if (gameepisode > BACKUPS)
        && consistency[fillbuf] != end->consistency)
            P_Nasallert;
        extern char *player_names[4];
        sprintf (turbo message, "%s is turbo! player: %s",
                player[gamestate].message = turbo message;
}

if (player[0].end->consistency != player[0].end->consistency)
    P_Nasallert;
    extern char *player_names[4];
    sprintf (turbo message, "%s is turbo! player: %s",
            player[gamestate].message = turbo message;
}

P_Nasallert;
extern char *player_names[4];
sprintf (turbo message, "%s is turbo! player: %s",
        player[gamestate].message = turbo message;
}

```





1)  $g\_max(1, 2) = 2$

2)  $g\_max(1, 3, 2) = 3$

3)  $g\_len([1, 1, 2, 2]) = 4$

4)  $vocal("a") = True$

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

“this is fun” =

“tothohisos isos fofunon”





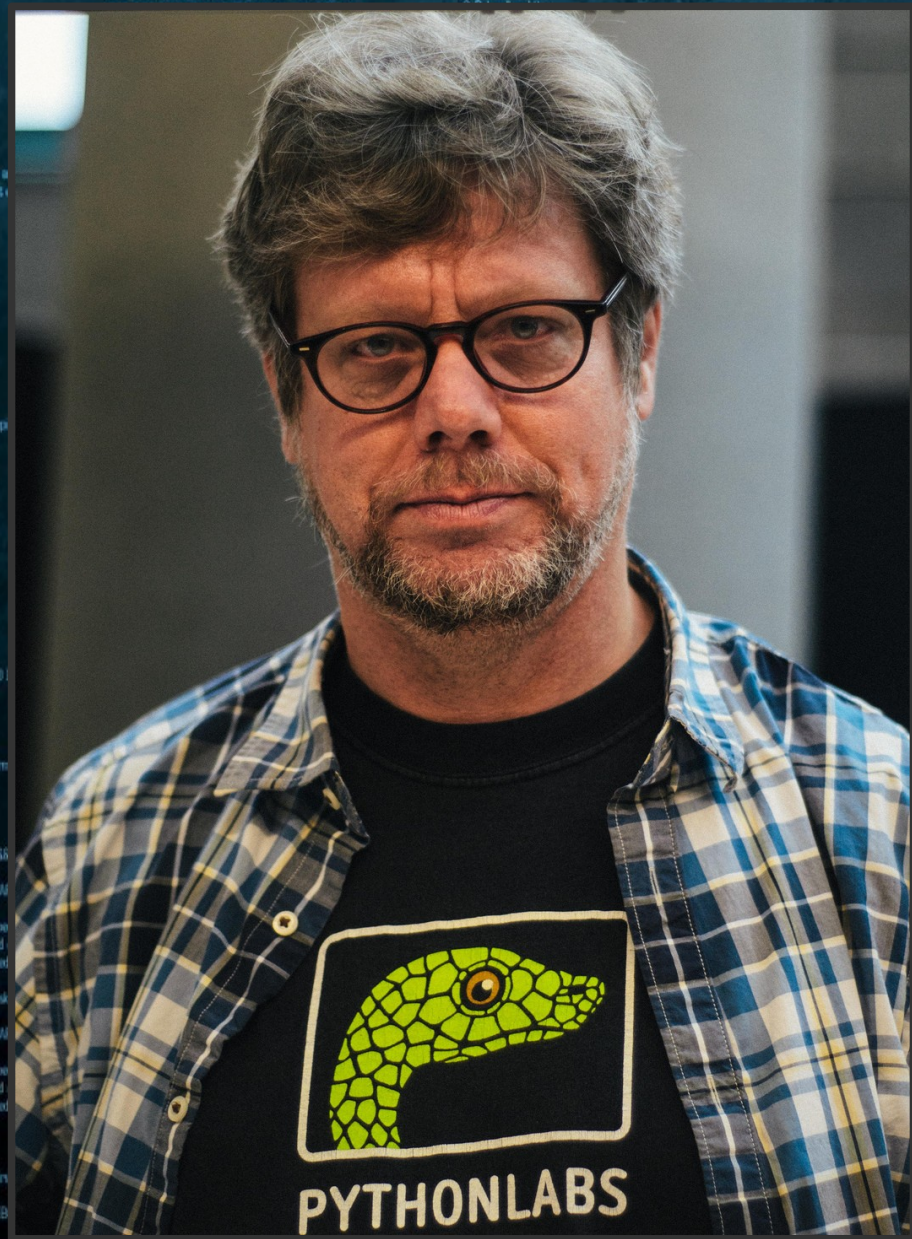
6)  $\text{suma}([1, 2, 3, 4]) = 16$

7)  $\text{multi}([1, 2, 3, 4]) = 24$

8)  $\text{voltea}(\text{"Es una prueba"}) = \text{"abeurp aun sE"}$

9)  $\text{pali}(\text{"radar"}) = \text{True}$  si radar es un palíndromo

10)  $\text{en}(xy, \text{lista}) = \text{True}$  si xy está en la lista

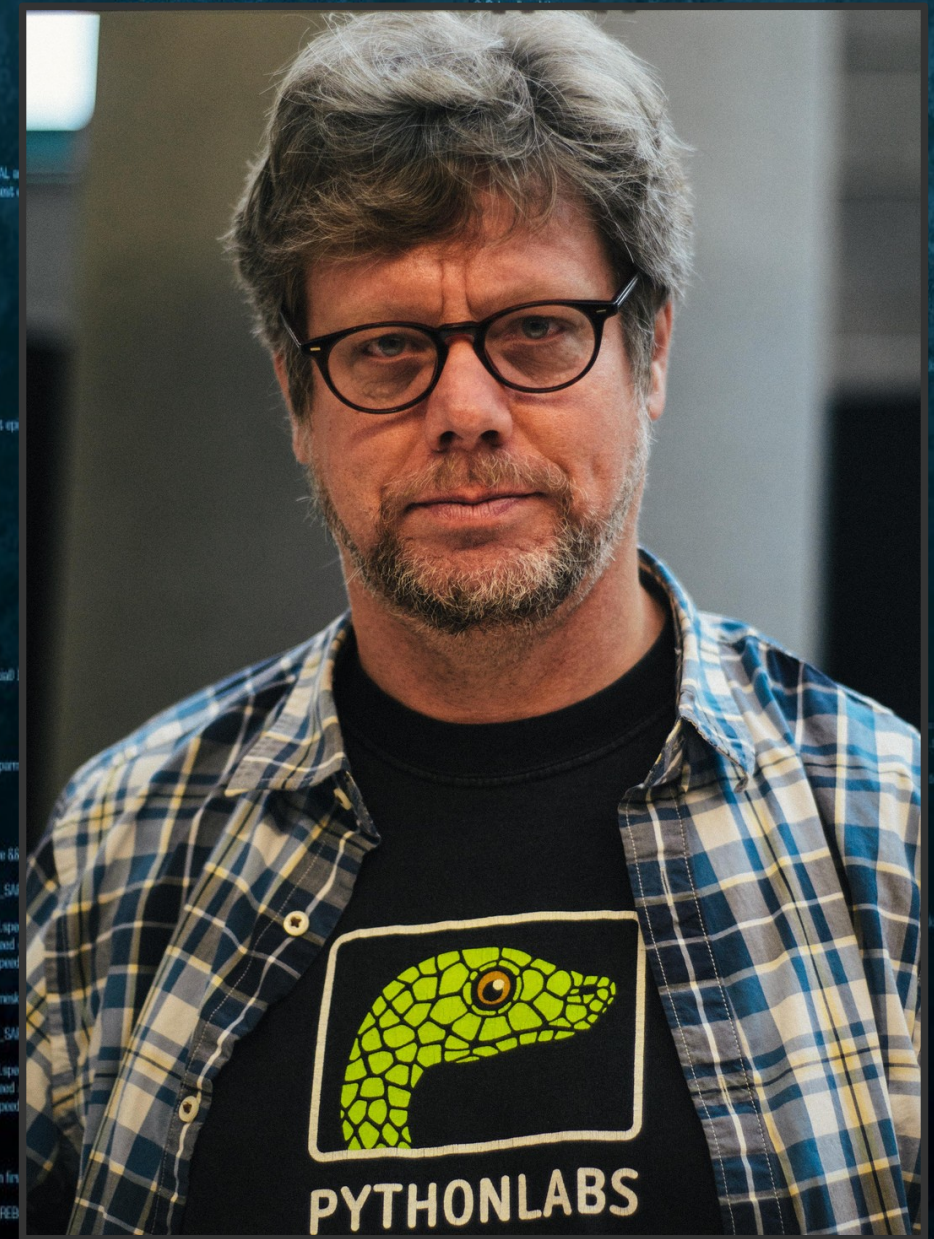




11) `solapa( lista1, lista2 )`  
= True si lista1 y lista2  
tienen al menos un  
elemento en común

12) `nchars( 'a' , 3 )` = “aaa”

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





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

15) `histochar( "abbabcb  
dbabdbdbabababcb  
cbab"`)  
a: `XXXXXXXX`  
b: `XXXXXXXXXXXXXXXXXX`  
c: `XXX`  
d: `XXX`

16) `rot13()` de Julio César





## 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):
```





17) adivina()  
Python elije num aleatorio  
el usuario pregunta  
y el sistema responde  
Indicando si está frío  
o caliente.

18) ahorcado()

19) tresenraya()

20) g2048() ó sudoku()











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

