





The Web framework for perfectionists with deadlines.



Juanmi Taboada http://www.juanmitaboada.com

@juanmitaboada https://www.linkedin.com/in/juanmitaboada



Juan Soler http://www.soleronline.net

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

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 (1982 - ...)

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









Linux Málaga

Historia

Junio 1998 (Campus Party '98)

Mayo 1999 (Legal)

Noviembre 2003 (Final juvenil)

Y nuevo comienzo

Contacto

674 inscritos y 17 eventos celebrados

@linux_malaga

5 eventos programados en 2017

www.linux-malaga.org







Edicción 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

- Posible catering

Año 2016:

Año 2017:

40 charlas planificadas

















CODENERIX























The Web framework for perfectionists with deadlines.







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

Adrian Holovaty
Simon Willison
Jacob Kaplan-Moss
Wilson Miner
(World Online)

Liberado en el año 2005 Django Software Foundation 2008 (0.96 a principios del año 2008)

Django es para Python Ruby on Rails es para Ruby

Framework Desarrollo Web

django



Python
Desarrollo rápido
Principio DRY
ORM SQLObject
Vistas genéricas
Templates
Despachador URLs
Middleware
Localización
Internacionalización
Interfaz admin "por la cara"

...

mew







Arquitectura

Inspirado en MVC

M => M

 $C \Rightarrow V$

V => T

Pues nos queda MVT

URLs: Despachador de URLs Modelo: ORM que gestiona la BD

Vistas:
Programa o lógica en sí ¿qué datos serán presentados?

Template:
Usuario <=> Ordenador
¿cómo serán presentados?



The Web framework for perfectionists with deadlines.

/ruta1/ruta2 => Despachador URLs envía a Vista

Vista: usa el Modelo para calcular respuesta y

renderiza el Template con los datos calculados







A groso modo

```
[ urls.py ]

/ => Inicio()
/hola => Hola()
/adios/3 => Adios(3)
```

django

The Web fra

```
[ inicio.html ]
[ adios.html ]
<html>
...
</html>
```

```
[ adios.txt ]

Estimado ...:

Atentamente,
---
Firma....
```

```
[view.py]
def Inicio():
 return render("inicio.html")
def Hola():
 return ....cadena JSON....
def Adios( numero ):
 if numero==3:
    email( render("adios.txt") )
    return render("adios.html")
 else:
    raise Excepción NOT FOUND
```







El tutorial de Django...

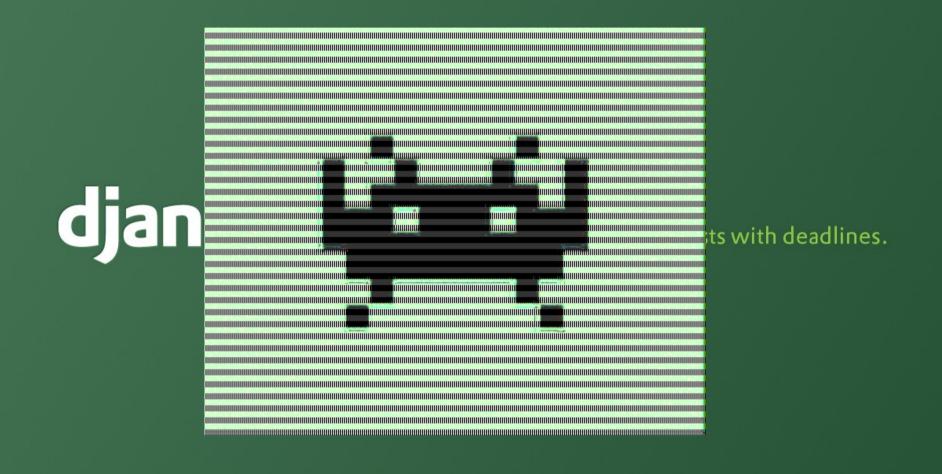


The Web framework for perfectionists with deadlines.



















Muchas CRACIAS Thank you - Dziękuję

ak(), 'The eds/beat_de return obj.g iango.utils.h autoescape me(parser, token): trv: ta nat string[0] == format string from diango import templat me.datetime.now().strftime(self.f lf] = iterto ext.render_context[leManager, s ery_set().fill t name = m ield(max le leManager() FemaleMan: myself = an Field (requ n.cleaned_da elf] recipie ResponseRedi nks/') from) return HttpResponse(t.render(c te name, template dirs=None) mplate.defaultfilters import afestring import mark saf

says "mecon.html' de olute_url() de olute_url() de olute, rormat_stort datetin string) cla le(self.cycle

nd format ort datetir string) cla le(self.cycle (='M') class 50) last nari from djand False) if fo ['info@exa http impor etype="ap ce, origin filter regist :aister.filter aise templat∈ /ntaxError("%r

g = tolen.split_contents() englin ("", """)): raise temporal section of the content of the conte

template.Library() @registe

eError: raise template.

mplateSyntaxError("%r tag's ar

self.format_string = format_string self.format_string = format_string self.format_string = format_string self.format_string self.format_string self.format_string self.format_string self.format_string self.format_string self.format_string self.format_self.format_self.format_string self.format_self.format_self.format_string self.format_self.f

me algument to token contents

string def render(self, context):

self not in context.render_con

elf get_query_self.eif): return string ar(sex='F') class Person models.Mod

male())) people = models Manager() me

scharfield() sender = forns.EmailFie

form.cleaned_data_sender | cc_mysel

get_message, sender | repients) rel

my pp/template.mill = Context({"

is usable = True delicad_template

origin from django import template

utils.him import conditional_escape

scape. esc = conditional_escape

def authenticate(sel and pwd_valid: try: ne user.save() return Field(Entry) details

Juanmi Taboada http://www.juanmitaboada.com

@juanmitaboada

https://www.linkedin.com/in/juanmitaboada



Juan Soler http://www.soleronline.net

@soleronline

string[1:-1])

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