

# Introducción a



Juan Pedro Fisanotti  
fisadev@gmail.com

- Tiene ~20 años. 3 vs 2
- Abierto Y Gratuito
- Sintaxis limpia y simple
- “Con pilas incluídas”
- Interpretado
- Multiparadigma
- Tipado dinámico y fuerte
- Multiplataforma
- Muy usado



Lo básico

- Modo interactivo
  - “charlando con el intérprete”
- Modo scripting
  - Ejecutando un programa en un archivo
- Y si quiero un exe para windows?
  - py2exe



# Interpretado

- Bloques por indentación
- El viejo y querido "if" (y versión compacta)
- Ciclos for (solo "foreach")
- Ciclos while
- Try
- Imports
- Funciones



## Sintaxis

- Números: igual que siempre
- Nulo: None
- Boleanos: True, False, (y guarda!!, muchas otras cosas no booleanas son verdaderas! Ej: if "hola")
- Strings: "a", 'a', ""a"", 'a'
- Conjuntos, Diccionarios, Tuplas
- Listas merecen más detalle

Tipos básicos

- Lógicas
  - and, or, not. Y devuelven valores!
- Números:
  - + - / \* \*\* %
- Comparación:
  - las mismas de siempre
- El polémico "is"



## Operaciones comunes

- Parámetros
  - opcionales
  - nombrados
  - sin saber la cantidad
- Docstrings!!
- Lambdas



Más sobre funciones

- Tipos inmutables:
  - “apunto a un nuevo objeto”
  - Números, strings, ...
- Tipos mutables:
  - Pueden cambiar el valor y seguir siendo el mismo objeto
  - Listas, clases del usuario, diccionarios...



Valores o Referencias???



- Consultando:
  - Slicing (bastante útil)
  - Pertenencia
- Modificandolas:
  - Borrado (bastante diferente)
  - Asignación (bastante útil)
- List comprehensions



## Listas y Strings

- **Todo** es un objeto
- No todo tiene que ser una clase!
- Propiedades "al vuelo"
- Todo es público
- Herencia (y múltiple)
- El mágico `__init__` y variables de instancia
- Métodos de instancia ("que hace el self ahí??")



## Clases y OO

- Librería base MUY amplia, desde zip hasta json, xml, archivos csv, expresiones regulares, unit testing, ...
- Muchas librerías de terceros, para lo que busquen y para lo que no busquen :D  
(apis de redes sociales, estadísticas, pdf, bases de datos, ORM, ...)



Las pilas incluídas

- Funciones útiles
- Parecen “sueltas”, pero están en el módulo `__builtins__`
- Ejemplos (algunas ya las vimos):
  - `dir`, `len`, `str`, `range`, `min`, `max`, `sum`, `raw_input`, `file`, `map`, `reversed`



## Builtins

- Varios frameworks. Los más usados, no incluídos con la librería base:
  - Qt (todas las plataformas)
  - Wx (todas las plataformas)
  - GTK (solo linux con gtk)
- Separación entre código y pantallas.




Interfaz gráfica en  
escritorio

- Varios frameworks, desde muuuy simples hasta más complejos:
  - Django ← muy usado!
  - web2py
  - Bottle
  - Flask
  - Pyramid
  - ...y otros menos conocidos



Web

- Y más:
  - Lenguaje más usado para ciencia de datos e inteligencia artificial (Machine Learning, etc)
  - Muy usado para infraestructura (scripts y herramientas de sysadmins) 
  - Android (con Kivy)
  - ...

Web

- No hay una sola, hay para elegir, y **muchas**.
- Algunas integran todo (editor de código, diseño de pantallas, debug, etc...).
- O se pueden usar herramientas separadas para cada cosa.
- Algunos buenos: Sublime Text, Atom, VS Code, Vim con fisa-vim-config :)



## Herramientas de Desarrollo



- Mucha documentación y material abierto (hasta el código mismo del lenguaje :D)
- Comunidades muy activas y abiertas a ayudar. En especial, PyAr ([python.org.ar](http://python.org.ar))



Soporte, aprendizaje,  
ayuda