

# Introducción a Python

¿Que es Python?, lenguajes de programación, ¡Hola Mundo!, variables, condicionales y lectura de archivos

Prof. Sebastian Saaibi & David Cardozo<sup>1</sup>

<sup>1</sup>Física

Lectura 3 Herramientas Computacionales  
Universidad de los Andes

27 de febrero de 2015

# Aprendiendo Python de la manera más difícil

Existen muchas investigaciones en el ámbito de como enseñar un lenguaje de programación. En general un programador realiza las siguientes 3 actividades cuando quiere aprender un lenguaje de programación:

- Seguir paso a paso cada ejercicio y ejemplo.

# Aprendiendo Python de la manera más difícil

Existen muchas investigaciones en el ámbito de como enseñar un lenguaje de programación. En general un programador realiza las siguientes 3 actividades cuando quiere aprender un lenguaje de programación:

- Seguir paso a paso cada ejercicio y ejemplo.
- Escribir cada ejemplo directamente

# Aprendiendo Python de la manera más difícil

Existen muchas investigaciones en el ámbito de como enseñar un lenguaje de programación. En general un programador realiza las siguientes 3 actividades cuando quiere aprender un lenguaje de programación:

- Seguir paso a paso cada ejercicio y ejemplo.
- Escribir cada ejemplo directamente
- Ejecutar

# Aprendiendo Python de la manera más difícil

Existen muchas investigaciones en el ámbito de como enseñar un lenguaje de programación. En general un programador realiza las siguientes 3 actividades cuando quiere aprender un lenguaje de programación:

- Seguir paso a paso cada ejercicio y ejemplo.
- Escribir cada ejemplo directamente
- Ejecutar

# Aprendiendo Python de la manera más difícil

Existen muchas investigaciones en el ámbito de como enseñar un lenguaje de programación. En general un programador realiza las siguientes 3 actividades cuando quiere aprender un lenguaje de programación:

- Seguir paso a paso cada ejercicio y ejemplo.
- Escribir cada ejemplo directamente
- Ejecutar

Es una actividad **difícil**, pues requiere ser un estudiante activo.

## Introducción (Continuación ...)

Es objetivo de esta clase, enseñar 3 habilidades esenciales que todo programador debe tener:

- Leer y Escribir

Aprender a escribir los símbolos raros que aparecen en las muestras de código \, &, %, /, ==

- Leer y Escribir
- Aprender a escribir los símbolos raros que aparecen en las muestras de código \, &, %, /, ==

Aprender a escribir los símbolos raros que aparecen en las muestras de código `\`, `&`, `%`, `/`, `==`

## Introducción (Continuación ...)

- Leer y Escribir  
Aprender a escribir los símbolos raros que aparecen en las muestras de código \, &, %, /, ==
- Atención a los detalles.  
La separación entre buenos y malos programadores es cuan atentos son a los detalles, con esta habilidad evitas **bugs**.

- Aprender a escribir los símbolos raros que aparecen en las muestras de código `\, &, %, /, ==`

- La separación entre buenos y malos programadores es cuan atentos son a los detalles, con esta habilidad evitas **bugs**.



# Introducción (Continuación ...)

Es objetivo de esta clase, enseñar 3 habilidades esenciales que todo programador debe tener:

- Leer y Escribir

Aprender a escribir los símbolos raros que aparecen en las muestras de código `\, &, %, /, ==`

- Atención a los detalles.

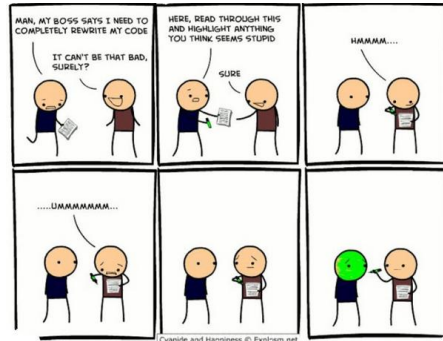
La separación entre buenos y malos programadores es cuan atentos son a los detalles, con esta habilidad evitas **bugs**.

- Encontrar las diferencias.

Esta una habilidad que se desarrolla con el tiempo, ver ejemplos de códigos muy similares y encontrar las diferencias.

# Lo mas importante

No hacer “copiar y pegar” sin entender cada linea de código.



Cyanide and Happiness © Exotism.net

# Características generales

## Lenguaje en sí (ver *import this*)

- Muy alto nivel
- Sintaxis uniforme y sencilla
- Completamente orientado a objetos
- Multiparadigma
- Énfasis está en la eficiencia de desarrollo

```
# -*- encoding: utf-8 -*-
```

```
"""
>>> import this # doctest: +NORMALIZE_WHITESPACE, +ELLIPSIS
The Zen of Python, by Tim Peters
<BLANKLINE>
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
...
"""
```

- Prof. Sebastian Saaibi & David Cardozo (Universidad de los Andes)

## Listing 1: Bash y Revision de Python

```
$ python
Python 2.6.5 (r265:79063, Apr 1 2010, 05:28:39)
[GCC 4.4.3 20100316 (prerelease)] on linux2
Type "help", "copyright", "credits" or "license"

for more information.

>>>
$ mkdir clase
$ cd clase
# ... Utilizar cualquier editor de texto para editar test.txt
$ ls
test.txt
$
```

# Un buen programa de inicio

## Listing 2: mi primer programa

```
print "Hola Mundo!"  
print "Hola otra vez"  
print "Me gusta teclear."  
print "Esto es divertido."  
print 'Si! a teclear'  
print "Yo preferiria que 'no'."  
print 'Yo "dije" no tipear esto'
```

Para poder utilizar acentos y  
eñes en español, requerimos el  
encoding UTF-8

## Listing 3: preambulo en python

```
# -*- coding: utf-8 -*-
```

## Ejercicio Avanzado

Coloque una # (Almohadilla ó octothorpe) al inicio de una linea. ¿Que hace?

# Comentarios y el caracter de almohadilla

Los comentarios son muy importantes, explican en “cristiano” lo que quieres hacer. [Regla](#)

## Listing 4: comentarios

```
# Un comentario, estos son para que
# puedas leer tu programa despues.
# Todo lo que va despues de # es
# ignorado por Python.

print "Podria tener codigo como esto" # y el comentario

# puedes utilizar la almohadilla para "desactivar"
# codigo o comentar un pedazo de codigo como:
# print "Esto no correra"

print "Esto correra"
```





# Números y Matemáticas

Listing 5: Problema de granja

```
print "Tipo de problemas de una granja"

print "Gallinas", 25 + 30 / 6
print "Toros", 100 - 25 * 3 % 4

print "Contemos ahora los gallos"

print 3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6

print "Es verdad que: 3 + 2 < 5 - 7?"

print 3 + 2 < 5 - 7

print "Cuanto es 3 + 2?", 3 + 2
print "Cuanato es 5 - 7?", 5 - 7

print "Oh, Por eso es que es falso"

print "How about some more."

print "Es mayor?", 5 > -2
print "Es mayor o igual?", 5 >= -2
print "Es menor o igual?", 5 <= -2
```

Cambiemos una división por 7.0/4.0

## Listing 6: Problema de Wheels Uniandes

```
carros = 100
espacio_en_un_carro = 4.0
conductores = 30
pasajeros = 90
carros_no_manejados = carros - conductores
carros_manejados = conductores
capacidad_de_carpooling = carros_manejados * espacio_en_un_carro
promedio_de_pasajeros_por_carro = pasajeros / carros_manejados

print "Existen", carros, "carros disponibles"
print "Tan solo hay", conductores, "conductores disponibles."
print "Van a haber", carros_no_manejados, "carros vacios hoy"
print "Podemos llevar", capacidad_de_carpooling, "personas"
print "Tenemos", pasajeros, "personas para hacer carpooling"
print "Necesitamos al menos", promedio_de_pasajeros_por_carro, "en cada carro"
```

## Preguntas

- ¿Que diferencia hay entre `=` y `==` ?

- ¿Que podrá hacer

```
print "Hola %s, me llamo %s ." %("tu", "david")
```

- ¿Por que utilizamos `espacio_en_un_carro = 4.0`?

Vamos a trabajar con **strings**, un string es como se un programa va dar un pedazo de información a un humano

## Listing 7: Problema de Wheels Uniandes

```
mi_nombre = 'David Cardozo'
mi_edad = 19 #
mi_altura = 170 # cms
mi_peso = 180 # lbs
mis_ojos = 'negros'
mis_dientes = 'blancos'
mi_cabello = 'cafe'

print "Vamos hablar de %s." % mi_nombre
print "El es %d centimetros de alto" % mi_altura
print "El tiene %d libras de peso." % mi_peso
print "Soy bien pesado"
print "El tiene ojos de color %s y cabello %s ." % (mis_ojos, mi_cabello)
print "Sus dientes %s dependen del cafe." % mis_dientes

print "Si yo sumo %d, %d, y %d obtengo %d." % (
mi_edad, mi_altura, mi_peso, mi_edad + mi_altura + mi_peso)
```

## Listing 8: Conversación con un Programador

```
x = "Existen %s tipos de personas" % 10
binario = "binario"
no = "no"
y = "Aquellas que saben %s y aquellos que %s." % (binario, no)

print x
print y

print "Yo dije %r." % x
print "I tambien dije: '%s'." % y

chistoso = False
evaluacion_chiste = "No es chistoso el chiste?! %r"

print evaluacion_chiste % chistoso

w = "Esta es la parte izquierda de..."
e = "un string con esta parte a la derecha."

print w + e
```

### importante

¿Que podría hacer: `print "." * 10` ?

## Listing 9: Dias de la semana

```
# -*- coding: utf-8 -*-
dias = "Lun Mar Mie Jue Vie Sab Dom"
meses = "Ene\nFeb\nMar\nAbr\nMay\nJun\nJul\nAgo"

print "Estos son los dias: ", dias
print "Estos son los meses: ", meses

print """
Hay algo extraño aqui
con tres doble citas.
Vamos a ser poder meter un monton de cosas.
Casi 4 lineas si queremos, o 5, o 6.
"""
```

## Listing 10: leer un archivo

```
from sys import argv
script, filename = argv

txt = open(filename)

print "Aqui esta su archivo %r:" % filename
print txt.read()

print "Escribirlo otra vez:"
file_again = raw_input("> ")

txt_again = open(file_again)

print txt_again.read()
```

## Listing 11: Escribiendo un texto

```
from sys import argv
script, filename = argv

print "We're going to erase %r." % filename
print "If you don't want that, hit CTRL-C (^C)."
```

print "If you do want that, hit RETURN."

```
raw_input("?")

print "Opening the file..."
target = open(filename, 'w')

print "Truncating the file. Goodbye!"
target.truncate()

print "Now I'm going to ask you for three lines."

line1 = raw_input("line 1: ")
line2 = raw_input("line 2: ")
line3 = raw_input("line 3: ")

print "I'm going to write these to the file."

target.write(line1)
target.write("\n")
target.write(line2)
target.write("\n")
target.write(line3)
target.write("\n")

print "And finally, we close it."
target.close()
```

## Listing 12: Condicionales

```
from sys import argv

personas = 20
gatos = 30
perros = 15

if personas < gatos:
    print "Too many cats! The world is doomed!"

if personas > gatos:
    print "Not many cats! The world is saved!"

if personas < perros:
    print "The world is drooled on!"

if personas > perros:
    print "The world is dry!"

perros += 5

if personas >= perros:
    print "People are greater than or equal to dogs."

if personas <= perros:
    print "People are less than or equal to dogs."

if personas == perros:
    print "People are dogs."
```



## Listing 13: loops

```
numero_leido = raw_input("inserta un numero >> ")
numero = int(numero_leido)
contador = 0
for i in range(1,numero+1):
    if (numero%i)==0:
        contador = contador + 1

if contador==2:
    print "el numero es primo"
else print "el numero no es primo"
```