

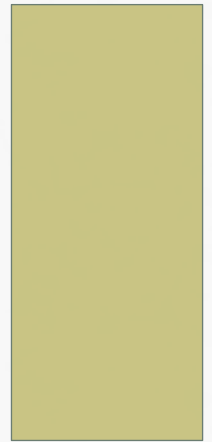


# python



“PYTHON EL LENGUAJE QUE ESTA  
REVOLUCIONANDO AL  
RADIOAFICIONADO ”

JOSE DE JESUS LOPEZ VILLALOBOS



# TEMARIO

- Introducción
- Ambiente
- Sintaxis básica
- Tipos de Variables
- Operaciones básicas
- Decisiones
- Ciclos
- Y aún más cosas
- Preguntas y dudas



# INTRODUCCION

Python es un lenguaje de propósito general interpretado, interactivo, orientado a objetos y de alto nivel. Fue creado por Guido Van Rossum durante 1985 a 1990. Al igual que otros lenguajes, el código fuente de Python esta disponible bajo Licencia General Pública GNU (GPL).



# AMBIENTE

- Python esta disponible para muchas plataformas y prácticamente se adapta a todos los sistemas operativos, en sus versiones 2.7 y 3 de Python. Las rutas de los archivos son autoajustables y es transparente para los usuarios. Para correr tus programas desde la red pueden usar esta terminal en la nube, accedendo a esta liga:



<http://www.tutorialspoint.com/codingground.htm>



# AMBIENTE

- Python se puede ejecutar en dos formatos:
- Modo Interactivo

```
$ python
Python 2.4.3 (#1, Nov 11 2010, 13:34:43)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
>>> print "Hello, Python!"
```

```
Hello, Python!
```

- Modo Script

```
$ python test.py
```

```
Hello, Python!
```



# SINTAXIS BÁSICA

Un identificador de Python es un nombre usado para identificar una variable, función, clase, módulo u otro objeto. Un identificador comienza con una letra A a la Z o A a la Z o un guión bajo (\_) seguido de cero o más letras, guiones y los dígitos (0 a 9).

Python no permite caracteres de puntuación tales como @, \$,% y dentro de los identificadores. Python es un lenguaje de programación entre mayúsculas y minúsculas. Por lo tanto, la mano de obra y mano de obra son dos identificadores diferentes en Python.

- Los nombres de clases comienzan con una letra mayúscula. Todos los otros identificadores comienzan con una letra minúscula.
- A partir de un identificador con un solo subrayado inicial indica que el identificador es privado.
- A partir de un identificador con dos subrayado iniciales indica un identificador fuerte privada.
- Si el identificador también termina con dos guiones de arrastre, el identificador es un nombre especial idioma definido.



# SINTAXIS BÁSICA

## Palabras reservadas

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield



# SINTAXIS BÁSICA

## Multilíneas

```
total = item_one + \  
        item_two + \  
        item_three
```

```
days = ['Monday', 'Tuesday', 'Wednesday',  
         'Thursday', 'Friday']
```

## Comillas

```
word = 'word'  
sentence = "This is a sentence."  
paragraph = """This is a paragraph. It is  
made up of multiple lines and sentences."""
```





# SINTAXIS BÁSICA

## Comentarios

```
#!/usr/bin/python  
  
# First comment  
print "Hello, Python!" # second comment
```



## Múltiple comentarios

```
# This is a comment.  
# This is a comment, too.  
# This is a comment, too.  
# I said that already.
```



# SINTAXIS BÁSICA

Espera de respuesta de usuario

```
#!/usr/bin/python  
raw_input("\n\nPress the enter key to exit.")
```

Múltiples instrucciones en una línea

```
import sys; x = 'foo'; sys.stdout.write(x + '\n')
```



# TIPOS DE VARIABLES

- Asignación de variables

```
#!/usr/bin/python

counter = 100          # An integer assignment
miles   = 1000.0      # A floating point
name    = "John"     # A string

print counter
print miles
print name
```



- Múltiple asignación

```
a = b = c = 1
```

```
a, b, c = 1, 2, "john"
```



# TIPOS DE VARIABLES

## Tipos de variables

- Números: Int, long, float, complex
- String
- List
- Tuple
- Diccionario

<b>int</b>	<b>long</b>	<b>float</b>	<b>complex</b>
10	51924361L	0.0	3.14j
100	-0x19323L	15.20	45.j
-786	0122L	-21.9	9.322e-36j
080	0xDEFABCECBDAECBFBAEI	32.3+e18	.876j
-0490	535633629843L	-90.	-.6545+0j
-0x260	-052318172735L	-32.54e100	3e+26j
0x69	-4721885298529L	70.2-E12	4.53e-7j



# TIPOS DE VARIABLES

## Conversión de variables (algunos ejemplos)

Function	Description
<code>int(x [,base])</code>	Converts <code>x</code> to an integer. <code>base</code> specifies the base if <code>x</code> is a string.
<code>long(x [,base] )</code>	Converts <code>x</code> to a long integer. <code>base</code> specifies the base if <code>x</code> is a string.
<code>float(x)</code>	Converts <code>x</code> to a floating-point number.
<code>complex(real [,imag])</code>	Creates a complex number.
<code>str(x)</code>	Converts object <code>x</code> to a string representation.



# OPERACIONES BÁSICAS

- Aritméticas

```
a = 21
b = 10
c = 0

c = a + b
print "Line 1 - Value of c is ", c

c = a - b
print "Line 2 - Value of c is ", c

c = a * b
print "Line 3 - Value of c is ", c

c = a / b
print "Line 4 - Value of c is ", c

c = a % b
print "Line 5 - Value of c is ", c

a = 2
b = 3
c = a**b
print "Line 6 - Value of c is ", c

a = 10
b = 5
c = a//b
print "Line 7 - Value of c is ", c
```



# OPERACIONES BÁSICAS

- lógicas

```
#!/usr/bin/python

a = 60          # 60 = 0011 1100
b = 13         # 13 = 0000 1101
c = 0

c = a & b;     # 12 = 0000 1100
print "Line 1 - Value of c is ", c

c = a | b;     # 61 = 0011 1101
print "Line 2 - Value of c is ", c

c = a ^ b;     # 49 = 0011 0001
print "Line 3 - Value of c is ", c

c = ~a;        # -61 = 1100 0011
print "Line 4 - Value of c is ", c

c = a << 2;    # 240 = 1111 0000
print "Line 5 - Value of c is ", c

c = a >> 2;    # 15 = 0000 1111
print "Line 6 - Value of c is ", c
```



# DECISIONES

- Se tiene el clásico.....

```
#!/usr/bin/python  
  
var = 100  
  
if ( var == 100 ) : print "Value of expression is 100"  
  
print "Good bye!"
```





# CICLOS

- Se tiene el WHILE

```
#!/usr/bin/python

count = 0
while (count < 9):
    print 'The count is:', count
    count = count + 1

print "Good bye!"
```



# CICLOS

- Se tiene el FOR

```
#!/usr/bin/python

for letter in 'Python':    # First Example
    print 'Current Letter :', letter

fruits = ['banana', 'apple', 'mango']
for fruit in fruits:      # Second Example
    print 'Current fruit :', fruit

print "Good bye!"
```



- Se pueden anidar los WHILE's y los FOR's



# Y AÚN MAS COSAS

- Se tienen además funciones (como la que se muestra abajo), listas, strings, manejo de números, manejo de archivos, módulos, tiempo y fecha, y un gran número de aplicaciones en la red disponibles para su uso.

```
#!/usr/bin/python

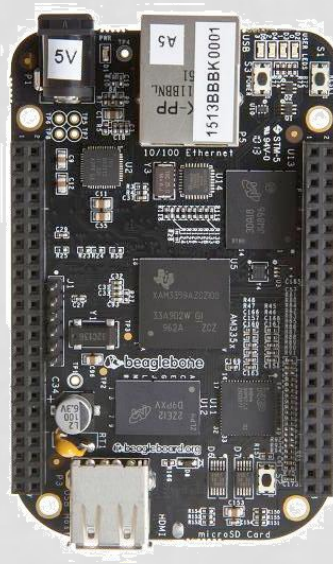
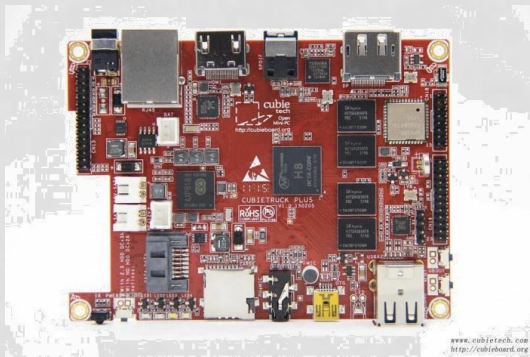
# Function definition is here
def changeme( mylist ):
    "This changes a passed list into this function"
    mylist = [1,2,3,4]; # This would assign new reference in mylist
    print "Values inside the function: ", mylist
    return

# Now you can call changeme function
mylist = [10,20,30];
changeme( mylist );
print "Values outside the function: ", mylist
```



# ¿EN DONDE LO APLICO?

- Se tienen una inmensidad de plataformas en donde Python tiene presencia Cubieboard, Raspberry, Arduino, Kontron Cards, etc. Simple cualquier tarjeta con sistema operativo: Windows, iOS o Linux lo soporta.



# ¿EN DONDE LO APLICO?

- Programa ejemplo:

```
##Imprimir los numeros del 1 al 100 y calcular la suma de todos los numeros
###pares por un lado, y por otro, la de los impares.
n = 1
p = 0
i = 0
while n <= 100:
    print n,
    if n%2 == 0:
        p += n
    else:
        i += n
    n += 1
print '\nLa suma de los pares es igual a %i' % p
print 'La suma de los impares es igual a %i' % i
```



# PREGUNTAS Y DUDAS

- Todas sus dudas por favor canalizarlas a través de mi correo electrónico, el cuál pongo a su disposición:

[xe2n@fmre.mx](mailto:xe2n@fmre.mx)

Además de las vías tradicionales de comunicación para radioaficionados.





GRACIAS POR SU ATENCION

