



*Skína***IT**
Solutions



Tutorial de Python

Ing. Ricardo Naranjo Faccini, M.Sc.
Noviembre 2008

Primeros pasos

```
•[skina@calypso python]$ python
•Python 1.5.2 (#1, Feb  1 2000, 16:32:16) [GCC egcs-
2.91.66 19990314/Linux (egcs- on linux-i386
Copyright 1991-1995 Stichting Mathematisch Centrum,
Amsterdam

•>>> 1+1
2

•>>> 3*4+5
17

•>>> a=12

•>>> a
12

•>>> "Hola"
'Hola'
>>>
```

Scripts

```
print "Hola, Mundo!"
```

```
$ python hello.py  
Hola, Mundo!
```

```
#!/usr/bin/env python  
print "Hola, Mundo!"
```

```
$ chmod +x hello.py  
$ hello.py  
Hola, Mundo!
```

¡¡Orientado a Objetos!!

```
>>> variable = 1
>>> dir(variable)

['_abs_', '__add__', '__and__', '__class__', '__cmp__',
 '__coerce__', '__delattr__', '__div__', '__divmod__',
 '__doc__', '__float__', '__floordiv__', '__format__',
 '__getattr__', '__getnewargs__', '__hash__',
 '__hex__', '__index__', '__init__', '__int__',
 '__invert__', '__long__', '__lshift__', '__mod__',
 '__mul__', '__neg__', '__new__', '__nonzero__',
 '__oct__', '__or__', '__pos__', '__pow__', '__radd__',
 '__rand__', '__rdiv__', '__rdivmod__', '__reduce__',
 '__reduce_ex__', '__repr__', '__rfloordiv__',
 '__rlshift__', '__rmod__', '__rmul__', '__ror__',
 '__rpow__', '__rrshift__', '__rshift__', '__rsub__',
 '__rtruediv__', '__rxor__', '__setattr__', '__sizeof__',
 '__str__', '__sub__', '__subclasshook__', '__truediv__',
 '__trunc__', '__xor__', 'conjugate', 'denominator',
 'imag', 'numerator', 'real']
```

¡¡Orientado a Objetos!!

```
>>> variable = '''Hola'''
>>> dir(variable)

['_add_', '__class__', '__contains__', '__delattr__',
'_doc_', '__eq__', '__format__', '__ge__',
'_getattr_', '__getitem__', '__getnewargs__',
'_getslice_', '__gt__', '__hash__', '__init__',
'_le_', '__len__', '__lt__', '__mod__', '__mul__',
'_ne_', '__new__', '__reduce__', '__reduce_ex__',
'_repr_', '__rmod__', '__rmul__', '__setattr__',
'_sizeof_', '__str__', '__subclasshook__',
'_formatter_field_name_split', '_formatter_parser',
'capitalize', 'center', 'count', 'decode', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'index',
'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace',
'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip',
'partition', 'replace', 'rfind', 'rindex', 'rjust',
'rstrip', 'rsplit', 'rstrip', 'split', 'splitlines',
'startswith', 'strip', 'swapcase', 'title', 'translate',
'upper', 'zfill']
```

Operadores

Operación	Símbolo	Ejemplo
Adición	+	$1 + 2 == 3$
Substracción	-	$4 - 3 == 1$
Multiplicación	*	$2 * 3 == 6$
División	/	$14 / 3 == 4$
Módulo	%	$14 \% 3 == 2$
Exponenciación	**	$5 ** 2 == 25$

Tipos de datos mixtos

```
print "14 / 3 = ", 14 / 3  
print "14 % 3 = ", 14 % 3  
print
```


Entrada de datos

```
print "¡Alto ahí!"  
s = raw_input("¿Quién es?")  
print "Hola, ", s
```

```
¡Alto ahí!  
¿Quién es?
```

```
num = input("Digite un número: ")  
cadena = raw_input("Una cadena: ")  
print "num =", num  
print "num es un ", type(num)  
print "num * 2 =", num*2  
print "cadena =", cadena  
print "cadena es un ", type(cadena)  
print "cadena * 2 =", cadena*2
```

Operadores lógicos

operador	función
<	menor que
<=	menor o igual que
>	mayor que
>=	mayor o igual que
==	igual
!=	diferente
<>	diferente

Condicionales

```
n = input("Numero? ")  
if n < 0:  
    print "El valor absoluto de ",n," es ",-n
```

Ciclos repetitivos

```
a = 0
while a < 10:
    a = a + 1
    if a > 5:
        print a, " > ", 5
    elif a <= 7:
        print a, " <= ", 7
    else:
        print "¡NPI!"
```

Funciones

```
def abs(num):  
    if num < 0:  
        num = -num  
    return num
```

Los valores absolutos de 23 y -23 son iguales

Uso de las funciones

```
def hello():  
    print "Hola"
```

Arreglos

```
arr_meses = [\n    'Enero', 'Febrero', 'Marzo', \n    'Abril', 'Mayo', 'Junio', 'Julio', \n    'Agosto', 'Septiembre', 'Octubre', \n    'Noviembre', 'Diciembre']
```

Listas

```
demolist = ['vida',42, 'Colombia', 6, 'nudo',7]  
print 'demolist = ',demolist
```

```
demolist = ['vida', 42, 'Colombia', 6, 'nudo', 7]
```


Listas

```
demolist=['vida',42,'Colombia',6,'nudo',7,'todo']  
print 'demolist[1] =', demolist[1]
```

```
demolist[1] = 42
```

```
demolist=['vida',42,'Colombia',6,'nudo',7,'todo']  
del demolist[2]  
print "Tras remover a 'Colombia' de demolist queda así:"  
print demolist
```

```
Tras remover a 'Colombia' de demolist queda así:  
['vida', 42, 6, 'nudo', 7, 'todo']
```

Listas

```
demolist=['vida',42,6,'nudo',7,'todo']  
demolist.sort()  
print 'al ordenar demolist queda así'  
print demolist
```

Listas

```
demolist = \  
    ['vida',42,'Colombia', 6,'nudo',7,'todo']  
for item in demolist:  
    print "El ítem actual es:",  
    print item
```

```
>>> list = ['cero', 'uno', 'dos', 'tres', 'cuatro', 'cinco']
```

```
>>> list[0]
```

```
'cero'
```

```
>>> list = ['cero', 'uno', 'dos', 'tres', 'cuatro', 'cinco']  
>>> len(list)  
6
```

```
>>> list=[0,'Pedro',2,'SPAM','Medias',42,"Juan","María"]
```

```
>>> list[7]
```

```
'María'
```

Listas

```
>>> list=[0,'Fred',2,'SPAM','Medias',42,"Juan","María"]
```


Muchas gracias



skinait
Solutions

www.skinait.com
ventas@skinait.com

¿Preguntas?