FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Procesamiento de señales, fundamentos

Maestría en sistemas embebidos
Universidad de Buenos Aires
MSE 5Co2020

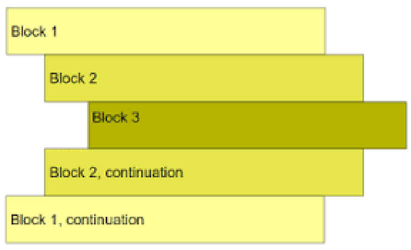
Introduccion a
Python y NumPy

Ing. Pablo Slavkin
<slavkin.pablo@gmail.com>

Introduccion

- Lenguaje Interpretado
- Tipado dinámico: El tipo se define en tiempo de ejecución
- Fuertemente tipado: Durante las operaciones se chequea el tipo

Indentacion



```python
mi_variable = 27
mi_flag     = True

if mi_flag:
    while mi_variable > 0:
        print(mi_variable)
        mi_variable-=1
```

Numeros

- Enteros: Con signo, sin límite
- Punto flotante
- Complejos
- Booleanos

```
nint   = 27
nfloat = 3.14

print(type(nint))
print(type(nfloat))
------------------------
>> <class ’int’>
>> <class ’float’>
```

```
flag1 = True
flag2 = False

print(type(flag1))
print(flag2)

if flag1:
    print("verdadero!")
------------------------
>> <class ’bool’>
>> False
>> verdadero!
```

Cadenas

- Strings: se guardan en codificadas. Ej. UTF8
- Byte array: sin codificar, raw

```
msg = "Hola mundo"
msg = ’Hola mundo’

print(msg)
print(msg[2])
print(msg[5:10])
-----------------------------
>> Hola mundo
>> l
>> mundo

b = bytearray()
b.append(0x02)
b.append(0x10)
b.append(0x05)
b.append(0x10)
b.append(0x03)
print(b)
print(b[3])
print(len(b))
-----------------------------
>> bytearray(b’\x02\x10\x05\x10\x03’)
>> 16
>> 5
```

Listas

```
l = [1, 2, 3, 4, 5 ];
print(l)
for i in l:
    print(i)
----------------------
>>
[1, 2, 3, 4, 5]
1
2
3
4
```

```
l = [1, 2, 3, 4, 5 ];
print(l[2])
print(l[-1])
print(l[1:3])
```

```
----------------------
>> 3
>> 5
>> [2, 3]
```

```
l.append(6)
l.remove(2) #por valor
print(l)
----------------------
>> [1, 3, 4, 5, 6]
```

```
lista = [1, 2, 3, 4, 5 ];
cantidad_elementos = len(lista)
print(cantidad_elementos)
----------------------
>> 5
```

funciones

```python
def miFuncion(arg1,arg2,arg3=1):
    print(arg1)
    print(arg2)
    print(arg3)
    return 5

r = miFuncion(1,2)
miFuncion("hola",2,3)
```

NumPy

arrays

```
a=np.array([1,2,3])
print(type(a))
print(a)
print(a+1)
-----------------------------
>>numpy.ndarray
>>[1 2 3]
>>[2 3 4]
```

NumPy

linspace

```python
import numpy as np

a=np.linspace(0,1,10)
print(a)
b=np.linspace(0,1,10,endpoint=False)
print(b)
c=np.arange(0,1,0.1)
print(c)
print(c[:2])
print(c[2:])
print(c[2:2])
print(c[::2])
print(c[::-1])
print(c[2])
print(c[-2])
for i in c:
    print(i)
-----------------------------
```

```
>>[0. 0.11111111 0.22222222 0.33333333
     0.44444444 0.55555556 0.66666667
     0.77777778 0.88888889 1.]
>>[0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
     0.9]
>>[0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
     0.9]
>>[0.  0.1]
>>[0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9]
>>[]
>>[0.  0.2 0.4 0.6 0.8]
>>[0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.
     ]
>>0.2
>>0.8
>>0.0
>>0.1
>>0.2
>>0.30000000000000004
>>0.4
>>0.5
>>0.6000000000000001
>>0.7000000000000001
>>0.8
>>0.9
```