

# Código destacado

---

Algoritmos Computacionales. Grupo 3009

Facultad de Ciencias. Universidad Nacional Autónoma de México

## Imprimir en pantalla

---

	Julia	Python
Imprimir <code>x</code> y bajarse a una línea nueva	<code>println(x)</code>	<code>print(x)</code>
Imprimir <code>x</code> y no crear línea nueva	<code>print(x)</code>	<code>print(x,end="")</code>

## Tipos de datos

---

	Julia	Python
enteros	<code>Int8 , Int16, Int32, Int64</code>	<code>int</code>
reales/flotantes	<code>Float16, Float32, Float64</code>	<code>float</code>
cadenas/strings	<code>String</code>	<code>str</code>
Booleanos	<code>Bool</code>	<code>bool</code>
Listas	<code>Array{Any,1}</code>	<code>lst</code>
Arreglos	<code>Array{Int8,1}, ...</code>	-

## Lectura de Datos

---

	Julia	Python
Leer datos como string y asignarlo a <code>x</code>	<code>x = readline()</code>	<code>x = input()</code>
Leer datos como tipo <code>T</code> y asignarlo a <code>x</code>	<code>x = parse(T,readline())</code>	<code>x = T(readline())</code>

## Operaciones matemáticas

	Julia	Python
$a + b, a - b$	<code>a+b</code> , <code>a-b</code>	<code>a+b</code> , <code>a-b</code>
$a \cdot b, a/b$	<code>a*b</code> , <code>a/b</code>	<code>a*b</code> , <code>a/b</code>
$a^b, \sqrt[b]{a}$	<code>a**b</code> , <code>a**(1/b)</code>	<code>a^b</code> , <code>a^(1/b)</code>
$e^x, \ln x$	<code>np.exp(x)</code> , <code>np.log(x)</code>	<code>exp(x)</code> , <code>log(x)</code>
$\sin x, \cos x$	<code>np.sin(x)</code> , <code>np.cos(x)</code>	<code>sin(x)</code> , <code>cos(x)</code>
$\lfloor x \rfloor, \lceil x \rceil$	<code>np.floor(x)</code> , <code>np.ceil(x)</code>	<code>floor(x)</code> , <code>ceil(x)</code>

## Operadores lógicos

	Julia	Python
$P \ \& \ Q$	<code>P &amp;&amp; Q</code>	<code>P and Q</code>
$P \ \vee \ Q$	<code>P    Q</code>	<code>P or Q</code>
$\neg P$	<code>!P</code>	<code>not P</code>
$a \leq b, a \geq b$	<code>a &lt;= b</code> , <code>a &gt;= b</code>	<code>a &lt;= b</code> , <code>a &gt;= b</code>
$a < b, a > b$	<code>a &lt; b</code> , <code>a &gt; b</code>	<code>a &lt; b</code> , <code>a &gt; b</code>
$a == b, a \neq b$	<code>a == b</code> , <code>a != b</code>	<code>a == b</code> , <code>a != b</code>

## Ciclos y estructuras de control

### Estructura if

#### Python

```

alumnos = 30
if alumnos <= 5:
    print("La clase no se abra")
else:
    print("Todo está bien")

```

#### Julia

```
alumnos = 30
if alumnos <= 5
    print("La clase no se abra")
else
    print("Todo está bien")
end
```

## Ciclo While

### Python

```
c = 1
itera = 8
while c <= itera:
    c = c + 1
    print("Vamos en la iteracion",end="")
    print(c)
print("Salimos")
```

### Julia

```
c = 1
itera = 8
while c <= itera
    global c
    c = c + 1
    print("Vamos en la iteracion",end="")
    print(c)
end
print("Salimos")
```

## Ciclo For

### Python

```
suma = 0
for i in range(1,6):
    suma = suma + i
print(suma)
```

### Julia

```

suma = 0
for i in range(1,6):
    global suma
    suma = suma + i
end

```

## Listas

### Funciones para crearlas

	Julia	Python	Resultado
numeros enteros entre dos valores	<code>range(1, stop = 5)</code>	<code>range(1, 6)</code>	<code>[1, 2, 3, 4, 5]</code>
intervalo con una distancia entera	<code>range(1, stop = 9, step = 2)</code>	<code>range(1, 10, 2)</code>	<code>[1, 3, 5, 7, 9]</code>
intervalo con una distancia real	<code>range(1, stop = 3, step = 0.5)</code>	<code>np.arange(1, 3.5, 0.5)</code>	<code>[1, 1.5, 2, 2.5, 3]</code>
intervalo con <code>n</code> puntos igualmente espaciados	<code>range(0, stop = 1, length = n)</code>	<code>np.linspace(0, 10, n)</code>	<code>[0, ..., 1]</code>

### Listas por comprensión

	Julia	Python
$\{g(x) \mid x \in A\}$	<code>[g(x) for x in A]</code>	<code>[g(x) for x in A]</code>

## Definición de funciones

### Python

```

def sucesionFacil(n):
    a = (-1)**n
    return a / n^2

```

### Julia

```
function sucesionFacil(n)
    a = (-1)^n
    return a / n^2
end
```