

- Características
 - Tipado Dinámico
 - Fuertemente Tipado
 - Multi-Plataforma
 - Multi-Paradigma
 - Software Libre
- Modos de Ejecución
 - Interactivo
 - Usando módulos

Repaso (2)

- Historia
 - Creador: Guido Van Rossum
- Paradigmas
- Frameworks
- En dónde se utiliza Python
- Entornos de desarrollo
- **Variables**

Bibliografía

- Documentación **oficial**:
 - <https://docs.python.org/3/>
- PyAr – Python Argentina
 - <http://python.org.ar>
- Wiki de Python
 - <https://wiki.python.org/moin/PythonBooks>
- PyCon:
 - <http://www.pycon.org/>

Tipos numéricos

- **Números**

- **Enteros (int)**

- Ejemplos:

- 3 – 7 – 11 - 200

- **Reales (float)**

- Ejemplos:

- 15.47 – 19.20

- **Complejos (complex)**

- Ejemplos:

- 7+5j

Recordar que en Python no es necesario declarar el tipo de variable.

Para consultar por el tipo de una variable podemos usar la función `type()`

Para verificar el tipo de una variable utilizamos la función `isinstance(object, classinfo)`

Tipos numéricos (2)

- Enteros (int)
 - El tipo **int** almacena números de cualquier precisión, limitado por la memoria disponible en la máquina.
 - Los números enteros pueden ser positivos, negativos o cero.
 - Soporta todas los operadores básicos de matemática
- En la versión 2.x de Python teníamos otro tipo de datos llamado Long.
- En la versión 3.x la representación long paso a ser int.

Tipos numéricos (3)

- Reales (float)
 - PY implementa su tipo float a bajo nivel mediante una variable de tipo double de C (64 bits) luego
 - Utiliza el estándar IEEE 754
 - 1 bit para el signo, 11 para el exponente, y 52 para la mantisa. $\pm 5.0 \times 10^{-324}$ (0) hasta $\pm 1.8 \times 10^{308}$ (inf).
 - `real = 0.2703`
 - También se puede utilizar notación científica, y añadir una e (de exponente) para indicar un exponente en base 10. Por ejemplo:
 - `real = 0.1e-3` ($0.1 \times 10^{-3} = 0.1 \times 0.001 = 0.0001$)

Tipos numéricos (4)

- Complejos (complex)
 - Python provee manejo de números complejos en su librería estándar
 - Valores se expresan en formato $a+bj$
 - a y b son números reales.
 - b es la parte imaginaria del número.
- Ejemplo
 - $c = 3.14 + 2.73j$
 - $c.real \rightarrow 3.14$
 - $c.imag \rightarrow 2.73$
 - $c.conjugate() \rightarrow (3.14 - 2.73j)$

Operaciones

Operaciones aritméticas

Operador	Descripción	Ejemplo
+	Suma	$r = 3 + 2$ # r es 5
-	Resta	$r = 4 - 7$ # r es -3
-	Negación	$r = -7$ # r es -7
*	Multiplicación	$r = 2 * 6$ # r es 12
**	Exponente	$r = 2 ** 6$ # r es 64
/	División	$r = 3.5 / 2$ # r es 1.75
//	División entera	$r = 3.5 // 2$ # r es 1.0
%	Módulo	$r = 7 \% 2$ # r es 1

Operaciones (2) - Orden de evaluación



Si no hay paréntesis en la expresión primero se agrupan los exponentes, luego los productos y cocientes, y luego las sumas y restas

Operación	Operador	Aridad	Asociatividad	Precedencia
Exponenciación	**	Binario	Por la derecha	1
Identidad	+	Unario	—	2
Cambio de signo	-	Unario	—	2
Multiplicación	*	Binario	Por la izquierda	3
División	/	Binario	Por la izquierda	3
Módulo (o resto)	%	Binario	Por la izquierda	3
Suma	+	Binario	Por la izquierda	4
Resta	-	Binario	Por la izquierda	4

Tabla 2.1: Operadores para expresiones aritméticas. El nivel de precedencia 1 es el de mayor prioridad y el 4 el de menor.

Operaciones (3)

- Se pueden usar paréntesis para indicar un orden de evaluación
 - $((b * b) - (4 * a * c)) / 4$
- Cuidado con las divisiones: Si x e y son números enteros, entonces x / y se calcula como la división entera entre x e y (Python 2x)
- *La división de dos números siempre retorna un tipo float (Python 3x)*
- El resultado de operar con números de distinto tipo es del tipo «mas complejo»

Ejercicios



- Ejercicio 1. Escribir un programa que pregunte al usuario:
 - a) su nombre, y luego lo salude.
 - b) dos números y luego muestre el producto.
- Ejercicio 2. Hacer un programa que:
 - a) Calcule el perímetro y área de un rectángulo dado su base y su altura.
 - b) Calcule el perímetro y área de un círculo dado su radio (Se puede utilizar math).

$$C = 2\pi r = \pi d$$

$$A = \pi r^2 = \pi \left(\frac{d}{2}\right)^2$$

Tipos bool

Solo tiene dos valores: True y False

Operador	Descripción	Ejemplo
==	¿son iguales a y b?	<code>r = 5 == 3 # r es False</code>
!=	¿son distintos a y b?	<code>r = 5 != 3 # r es True</code>
<	¿es a menor que b?	<code>r = 5 < 3 # r es False</code>
>	¿es a mayor que b?	<code>r = 5 > 3 # r es True</code>
<=	¿es a menor o igual que b?	<code>r = 5 <= 5 # r es True</code>
>=	¿es a mayor o igual que b?	<code>r = 5 >= 3 # r es True</code>

Python permite comparaciones de este tipo: `1 <= a <= 7` (rangos)

Operadores booleanos



Operador	Descripción	Ejemplo
and	¿se cumple a y b?	<code>r = True and False # r es False</code>
or	¿se cumple a o b?	<code>r = True or False # r es True</code>
not	No a	<code>r = not True # r es False</code>

Tipo string

- Dentro de las comillas se pueden añadir caracteres especiales escapando con el carácter '\'
 - 'Help! Help! I\'m being repressed!'
 - “Help! Help! I'm being repressed!”
- Podemos encerrar la cadena en comillas simples, dobles o incluso triples.
 - >>> triple = """ Primera linea
 - ... otra linea
 - ... otra linea mas"""
- La triple ahorra poner un '\n' en la cadena

Tipo string (2)

- Operador +: operador de concatenación
- Operador *: operador de repetición
- Función len: largo de la cadena

```
a = "uno"  
b = "dos"
```

```
c = a + b # c es "unodos"  
c = a * 3 # c es "unounouno"
```

Tipo string (3)

- Cada carácter en un string tiene un índice o desplazamiento que comienza en 0
- Ejemplo:
 - La palabra PYTHON contiene 6 caracteres: Para recuperar la letra Y podemos acceder `"PYTHON"[1]`
- También podemos acceder con índices

0	1	2	3	4	5	6	7	8	9	10	11
H	o	l	a	,		m	u	n	d	o	.
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Tipo string (4)

- El operador de corte ":" se introduce dentro de la indexación `a[i:j]` y significa que la cadena resultado (o subcadena) estará formada por los elementos: $a[i], a[i+1], \dots, a[j-1]$

Podemos omitir los índices:

- `palabra[:j]` es igual a `palabra[0:j]`
- `palabra[i:]` es igual a `palabra[i:largo de la palabra]`
- `palabra[:]` es igual a `palabra[0:largo de la palabra]`
- Otra forma: acepta tres valores separados por ":"
palabra[inicio:fin:salto]
 - *palabra[0:4:2]* o bien *palabra[0:len(palabra):2]*
 - Obtenemos todas las letras de las posiciones pares.

Tipo string (5)

- No podemos modificar una cadena existente en Python.
- Lo que sí podemos hacer es crear una nueva cadena modificando la original.
 - Ejemplo
 - `palabra = "Bienbenido al mundo Python"`
 - `palabra[4] = "v"`
 - Traceback (most recent call last):
 - File "<stdin>", line 1, in <module>
 - TypeError: 'str' object does not support item assignment
 - `palabra = palabra[:4] + "v" + palabra[5:]`

Tipo string (6)

- `len()`: Retorna la longitud de la palabra.
- `str.lower()`: Convierte el string a una palabra en minúscula
- `str.upper()`: Convierte el string a una palabra en mayúscula
- `str.replace("a","b")`: Sustituye todas las ocurrencias de "a" con la letra "b"
- `str()`: casting

Cast (int, float, str, bool)

```
a = 10
```

```
a_casteado = str(a)
```

```
print ("tipo de a_casteado es: ", type(a_casteado))
```

```
#cast falla
```

```
txt = "bla"
```

```
txt_casteado = int(txt)
```

```
print ("tipo de txt_casteado es: ", type(txt_casteado))
```

```
#cast ok
```

```
txt = "true"
```

```
txt_casteado = bool(txt)
```

```
print ("tipo de txt_casteado es: + str(type(txt_casteado)))
```

Entrada y asignación

- ***input()*** devuelve el valor ingresado por teclado.
 - `valor = input("Ingrese un valor")`
 - El tipo ingresado es texto.
- **Asignación Simple**
 - `<var> = <expr>`
 - `miVar = oldValue * foo + skip`
- **Asignación Simultanea**
 - `<var1>, <var2>, ... = <expr1>, <expr2>`
 - `a, b = 1,2`

Ejercicios (2)

- Ejercicio 3
 - Dada una palabra ingresada desde el teclado mostrar.
 - Largo de la palabra
 - Imprimir las dos primeras letras de la cadena
 - Imprimir las tres ultimas letras de la cadena
 - Toda la palabra escrita en mayúscula
 - Toda la palabra escrita en minúscula

Ejercicios (3)



- Ejercicio 4
 - Hacer un programa para averiguar si una frase es o no palíndrome.
 - Por ejemplo
 - “Añora la Roña”
 - “Acaso hubo búhos acá”
 - “luz azul”
 - “la ruta natural”
 - “sé verla al revés”
 - “Dabale arroz a la zorra el Abad”