

Please verify your email address to access all of GitHub's features.

[Configure email settings →](#)

An email containing verification instructions was sent to javicerodriguez@gmail.com.

 [python-unsam](#) / [UNSAM_2020c2_Python](#)

[Code](#)[Issues](#)[Pull requests](#) 1[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#) master ▾

...

[UNSAM_2020c2_Python](#) / [Notas](#) / [01_Introduccion](#) / **03_Numeros.md**



rgrimson actualizo repo



 1 contributor

[Raw](#)[Blame](#)

308 lines (224 sloc) 8.3 KB

[Contenidos](#) | [Anterior \(2 Un primer programa\)](#) | [Próximo \(4 Cadenas\)](#)

1.3 Números

Esta sección introduce los cálculos matemáticos.

Tipos de números

Python tiene 4 tipos de números:

- Booleanos
- Enteros
- Punto flotante
- Complejos (con parte real y parte imaginaria)

Booleanos (bool)

Las variables booleanas se llaman así en honor al lógico inglés George Bool. Pueden tomar dos valores: True y False (verdadero y falso).

```
a = True
b = False
```

Internamente, son evaluados como enteros con valores 1 , 0 .

```
c = 4 + True # 5
d = False
if d == 0:
    print('d is False')
```

No escribas código basado en esta convención. Sería bastante raro.

Enteros (int)

Representan números enteros (positivos y negativos) de cualquier magnitud:

```
a = 37
b = -299392993727716627377128481812241231
```

Incluso se pueden especificar en diferentes bases:

```
c = 0x7fa8      # Hexadecimal
d = 0o253       # Octal
e = 0b10001111  # Binario
```

Operaciones usuales:

x + y	Suma
x - y	Resta
x * y	Multipliación
x / y	División (da un float, no un int)
x // y	División entera (da un int)
x % y	Módulo (resto)
x ** y	Potencia
abs(x)	Valor absoluto

La unidad mínima de almacenamiento de una computadora es un bit, que puede valer 0 o 1. Los números, caracteres e incluso imágenes y sonido son almacenados en la máquina usando bits. Los números enteros positivos, en particular, suelen almacenarse mediante su representación binaria (o en base dos).

Número	Representación binaria
1	1

Número	Representación binaria
2	10
3	11
4	100
5	101
6	110

Hay algunas operaciones primitivas que se pueden hacer con los enteros a partir de su representación como bits:

```
x << n    Desplazamiento de los bits a la izquierda
x >> n    Desplazamiento de los bits a la derecha
x & y     AND bit a bit.
x | y     OR bit a bit.
x ^ y     XOR bit a bit.
~x        NOT bit a bit.
```

Al desplazar a la izquierda, simplemente agregamos un cero en la última posición. Así, por ejemplo si corremos el 1 dos lugares a la izquierda obtenemos un 4:

```
>>> 1 << 2 # 1 << 2 -> 100
4
>>> 6 & 3 # 110 & 011 -> 010
2
```

Al desplazar los bits de un número a la derecha un lugar, el último bit "se cae".

```
>>> 1 >> 1 # 1 -> 0
0
>>> 6 >> 1 # 110 -> 11
0
```

Punto flotante (float)

Usá una notación con decimales o una notación científica para especificar un valor de tipo punto flotante:

```
a = 37.45
b = 4e5 # 4 x 10**5 o 400,000
c = -1.345e-10
```

Los números de tipo floats son representados en la máquina como números de doble precisión usando la representación nativa del microprocesador [IEEE 754](#). Es el mismo tipo que los `double` en el lenguaje C (para los que conozcan).

Almacenan hasta 17 dígitos con un exponente entre -308 to 308

Cuidado que la aritmética de los números de punto flotante no es exacta.

```
>>> a = 2.1 + 4.2
>>> a == 6.3
False
>>> a
6.3000000000000001
>>>
```

Esto **no es un problema de Python**, si no el resultado de la forma en que el hardware de nuestras computadoras almacena los números de punto flotante.

Operaciones usuales:

<code>x + y</code>	Suma
<code>x - y</code>	Resta
<code>x * y</code>	Multiplicación
<code>x / y</code>	División (da un float, no un int)
<code>x // y</code>	División entera (da un float, pero con ceros luego del punto)
<code>x % y</code>	Módulo (resto)
<code>x ** y</code>	Potencia
<code>abs(x)</code>	Valor absoluto

Estas son las mismas operaciones que con los enteros. Otras operaciones usuales se encuentran en el módulo `math`.

```
import math
a = math.sqrt(x)
b = math.sin(x)
c = math.cos(x)
d = math.tan(x)
e = math.log(x)
```

El módulo `math` también tiene constantes (`math.e`, `math.pi`), entre otras cosas.

Comparaciones

Las siguientes comparaciones (suelen llamarse *operadores relacionales* ya que expresan una relación entre dos elementos) funcionan con números:

<code>x < y</code>	Menor que
<code>x <= y</code>	Menor o igual que
<code>x > y</code>	Mayor que
<code>x >= y</code>	Mayor o igual que
<code>x == y</code>	Igual a
<code>x != y</code>	No igual a

Observá que el `==` se usa para comparar dos elementos mientras que el `=` se usa para asignar un valor a una variable. Son símbolos distintos que cumplen funciones diferentes.

Podés formar expresiones booleanas más complejas usando

`and` , `or` , `not`

Acá mostramos algunos ejemplos:

```
if b >= a and b <= c:
    print('b está entre a y c')

if not (b < a or b > c):
    print('b sigue estando entre a y c')
```

Conversión de números

El nombre de un tipo (de datos) puede ser usado para convertir valores:

```
a = int(x)    # Convertir x a int
b = float(x)  # Convertir x a float
```

Probalo.

```
>>> a = 3.14159
>>> int(a)
3
>>> b = '3.14159' # También funciona con cadenas que represetan números.
>>> float(b)
3.14159
>>>
```

Cuidado: el separador decimal en Python es el punto, como en inglés, y no la coma del castellano. Así, el comando `float(3,141592)` da un "ValueError".

Ejercicios

Recordatorio: Asumimos que estás trabajando en el subdirectorio /Ejercicios .
Buscá el archivo hipoteca.py y hacé los ejercicios con un editor de texto en ese archivo. Ejecutalo desde la línea de comandos.

Ejercicio 1.7: La hipoteca de David

David solicitó un crédito a 30 años y a tasa fija para comprar una vivienda. Pidió \$500,000 a la compañía y acordó un pago mensual fijo de \$2684.11.

El siguiente es un programa que calcula el monto total que pagará David a lo largo de los años:

```
# hipoteca.py

saldo = 500000.0
tasa = 0.05
pago_mensual = 2684.11
total_pagado = 0.0

while saldo > 0:
    saldo = saldo * (1+tasa/12) - pago_mensual
    total_pagado = total_pagado + pago_mensual

print('Total pagado', round(total_pagado, 2))
```

Copíá este código y correlo. Deberías obtener 966279.6 como respuesta.

Ejercicio 1.8: Adelantos

Supongamos que David adelanta pagos extra de \$1000/mes durante los primeros 12 meses de la hipoteca.

Modificá el programa para incorporar estos pagos extra y que imprima el monto total pagado junto con la cantidad de meses requeridos.

Cuando lo corras, este nuevo programa debería dar un pago total de 929965.62 en 342 meses.

Ejercicio 1.9: Calculadora de adelantos

Modificá tu programa de forma que la información sobre pagos extras sea incorporada de manera más versátil. Agregá las siguientes variables antes del ciclo:

```
pago_extra_mes_comienzo = 61
pago_extra_mes_fin = 108
pago_extra = 1000
```

Hacé que el programa tenga en cuenta estas variables para calcular el total a pagar apropiadamente.

¿Cuánto pagaría David si agrega \$1000 por mes durante cuatro años, comenzando en el sexto año de la hipoteca?

Ejercicio 1.10: Tablas

Modicá tu programa para que imprima una tabla mostrando el mes, el total pagado hasta el momento y el saldo restante. La salida debería verse aproximadamente así:

```
1 2684.11 499399.22
2 5368.22 498795.94
3 8052.33 498190.15
4 10736.44 497581.83
5 13420.55 496970.98
...
308 874705.88 3478.83
309 877389.99 809.21
310 880074.1 -1871.53
Total pagado: 880074.1
Meses: 310
```

Ejercicio 1.11: Bonus

Ya que estamos, corregí el código anterior de forma que el pago del último mes se ajuste a lo adeudado.

Asegurate de guardar el archivo `hipoteca.py` en esta última versión en tu directorio `Ejercicios`. Vamos a volver a trabajar con él.

Ejercicio 1.12: Un misterio

Las funciones `int()` y `float()` pueden usarse para convertir números. Por ejemplo,

```
>>> int("123")
123
>>> float("1.23")
1.23
>>>
```

Con esto en mente, ¿podrías explicar el siguiente comportamiento?

```
>>> bool("False")
True
>>>
```

Ejercicio 1.13: El volúmen de una esfera

Escribí un programa llamdo `esfera.py` en el directorio de trabajo que le pida al usuario que ingrese por teclado el radio r de una esfera y calcule e imprima el volumen de la misma. *Sugerencia: recordar que el volúmen de una esfera es $\frac{4}{3} \pi r^3$.*

Finalmente, ejecutá el programa desde la línea de comandos para responder ¿cuál es el volumen de una esfera de radio 6? Debería darte `904.7786842338603`.

[Contenidos](#) | [Anterior \(2 Un primer programa\)](#) | [Próximo \(4 Cadenas\)](#)