

Python

para no programadores

Módulo 2

Conversiones entre tipos de dato

Conversiones entre tipos de dato

En la clase anterior vimos los cuatro tipos de dato básicos. Es una acción muy común y muy sencilla la de convertir de un tipo de dato a otro.

Por ejemplo, podemos convertir un número de coma flotante a un número entero vía la instrucción `int`.

```
>>> pi = 3.14  
>>> pi_entero = int(pi)  
>>> pi_entero  
3
```

Como los números enteros no pueden tener parte decimal, convertir un número de coma flotante a un número entero implica despreciar los decimales.



Conversiones entre tipos de dato

Pero hay una conversión en particular que es la que más nos interesa y la más empleada: la conversión de una cadena a un número entero y viceversa. Para el primer caso usamos asimismo la instrucción `int`:

```
>>> numero_en_cadena = "30"
>>> numero_entero =
int(numero_en_cadena)
>>> numero_entero
30
>>> type(numero_en_cadena)
<class 'str'>
>>> type(numero_entero)
<class 'int'>
```

Para la conversión de un número entero a una cadena, usamos `str`:

```
>>> numero_entero = 30
>>> numero_en_cadena =
str(numero_entero)
>>> numero_en_cadena
'30'
>>> type(numero_en_cadena)
<class 'str'>
```

Conversiones entre tipos de dato

La diferencia entre tener un número almacenado como una cadena o como un entero radica en las operaciones que queremos ejecutar sobre ellos. Si el número treinta lo queremos sumar a otro número, será mejor tenerlo almacenado como un número entero (puesto que las operaciones aritméticas solo funcionan sobre números enteros y de coma flotante). Por otro lado, si el número treinta lo necesito para imprimir en pantalla "Tu edad es: 30", entonces conviene tenerlo como una cadena, para poder realizar la concatenación vía el operador `+`.

Ilustremos esto que acabamos de decir:

```
>>> numero_treinta = 30
>>> cadena_treinta = "30"
>>> # Permitido: suma entre dos enteros.
... numero_treinta + 10
40
>>> # No permitido: suma entre una cadena y un entero.
... cadena_treinta + 10
[Error]
>>> # Permitido: suma entre dos cadenas (concatenación).
... print("Tu edad es: " + cadena_treinta)
Tu edad es: 30
... # No permitido: suma entre una cadena y un entero.
>>> print("Tu edad es: " + numero_treinta)
[Error]
```

¡Muchas gracias!

¡Sigamos trabajando!