



Limitantes y errores

Limitantes

Si bien Python se caracteriza por ser un lenguaje de programación de sintaxis simple, esto no significa que no tenga reglas para escribir el código. Por ejemplo, Python no nos permite sumar letras y números.

A lo largo del módulo y del curso, se irán viendo estas reglas en detalle según el contenido revisado. Una muy importante de recordar es respetar la indentación al interior de controles de flujo y funciones, lo cual se verá más adelante.

Siempre que Python no sea capaz de ejecutar una instrucción debido a un error de sintaxis, se mostrará un mensaje explicando el error.

```
In [2]: "gato" + 2
-----
TypeError                                Traceback (most recent call last)
<ipython-input-2-37334b11ca62> in <module>()
----> 1 "gato" + 2

TypeError: must be str, not int
In [3]:
```

Facilitar la escritura y evitar errores

Los editores de texto nos permiten facilitar la escritura del código, resaltando ciertas palabras que signifiquen instrucciones específicas con colores diferentes, o indentando automáticamente cuando corresponda.

Otra opción, estando en la terminal, es utilizar el kernel `iPython` en lugar de `python`. Al igual que en un editor de texto, se resaltan distintos tipos de palabras o instrucciones en distintos colores, y también realiza indentación de forma automática.

```
(base) C:\Users\Gianina>ipython
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bi
t (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.4.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: 1 + 1
Out[1]: 2

In [2]:
```