**David Martín**

Python Courses



[Course from zero to Heroe - Udemy - 3](#_Toc974734377)

[Instalación de Python y SetUp 4](#_Toc975717381)

[Data types 6](#_Toc1509456115)

[Uso de variables 6](#_Toc134824603)

[Strings 7](#_Toc2094734125)

# Course from zero to Heroe - Udemy -

## Instalación de Python y SetUp

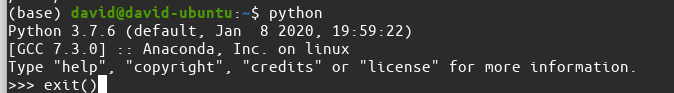
Bajaremos un paqueta q se llama anaconda compuesto por python y una serie de librerías útiles junto con editores de texto como Jupiter. Lo bajamos desde <https://www.anaconda.com/distribution/>

1. Instamos en nuestro direcotrio

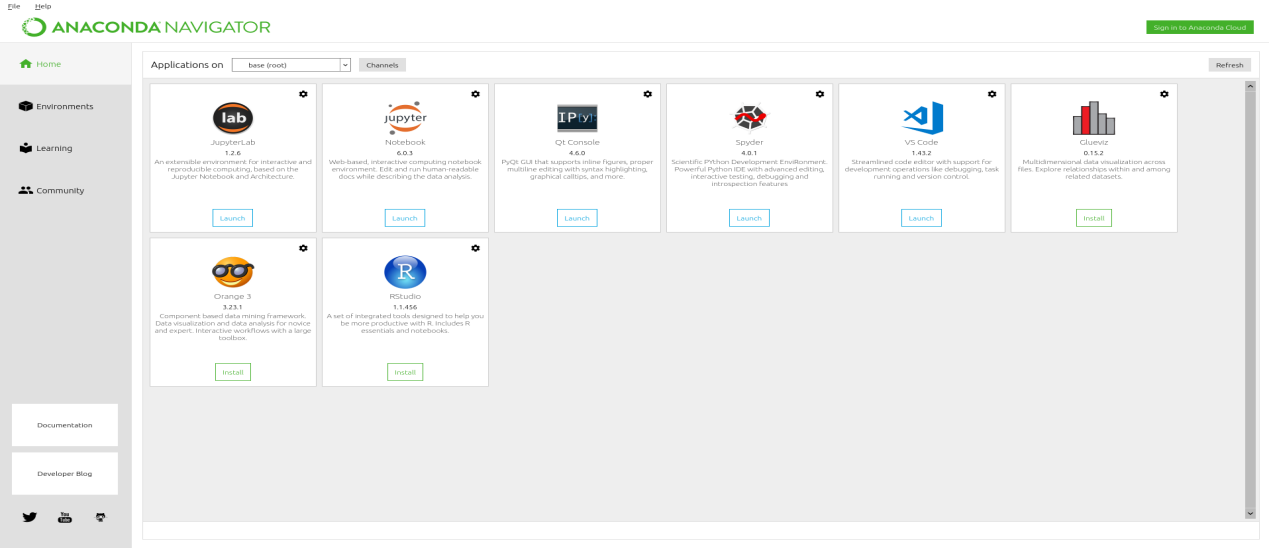
Ejecutamos el archivo de instalación



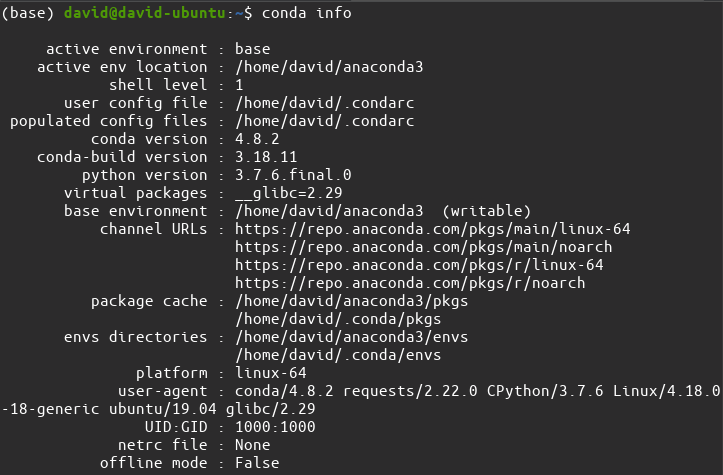
1. Añadimos python al PATH

1. Comprobamos que se ha instalado adecuadamente
2. Lanzamos la interfaz de anaconda





1. Para ver info de anaconda



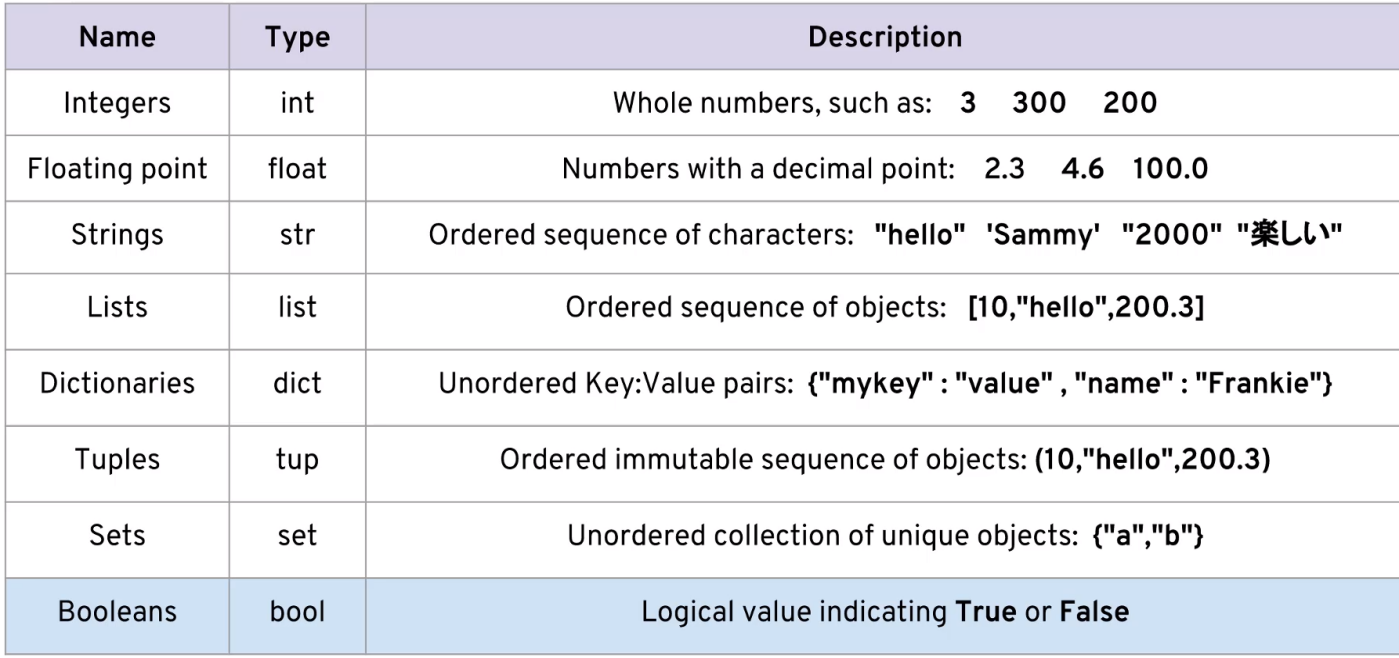
1. Vemos como en el prompt pone base eso es xq es el interprete de anaconda para desactivarlo



1. Para activarlo conda activate



## Data types



## Uso de variables

Python usa tipado dinámico como JavaScript esto significa que no es necesario especificar el tipo de dato que contendrá dicha variable.

Por ejemplo Java tiene un tipado estático porque requiere que especifiquemos durante la declaración el tipo de dato que contendrá la variable.

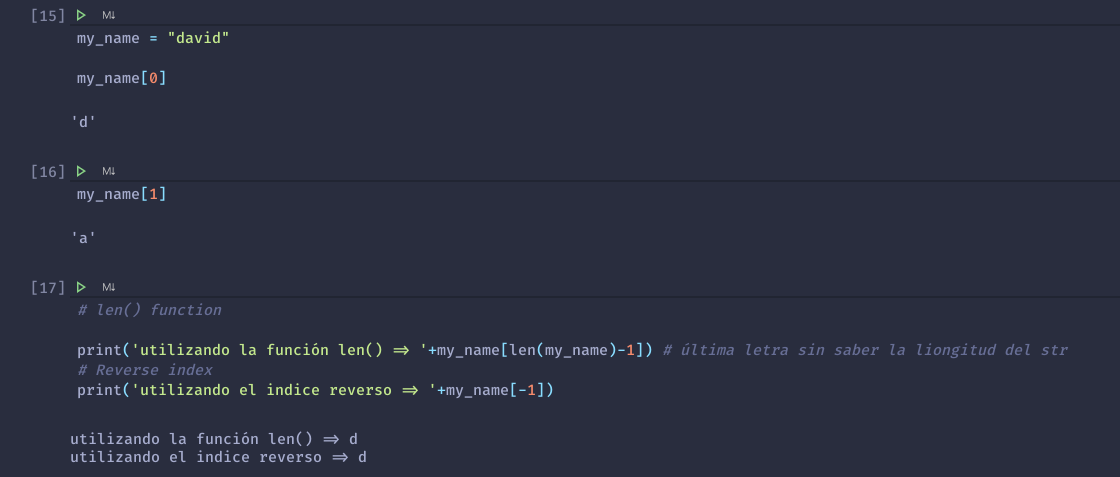
*Usado : str type()*



## Strings

Funcionan como arrays así que usando el índice podemos extraer letras del string.

Está habilitado el reverse index(indice reverso) así sin conocer la longitud del string podemos extraer el último carácter.

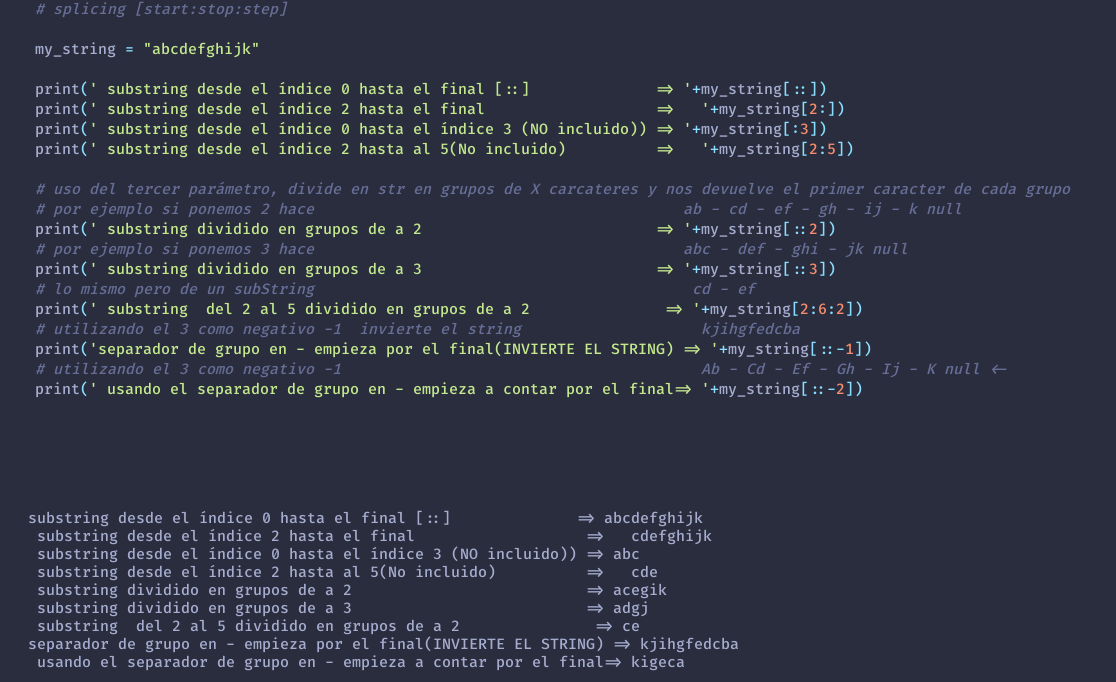


Tamaño de un str usamos la función len()

### Slicing de un str [start:stop:step] obtener un subString

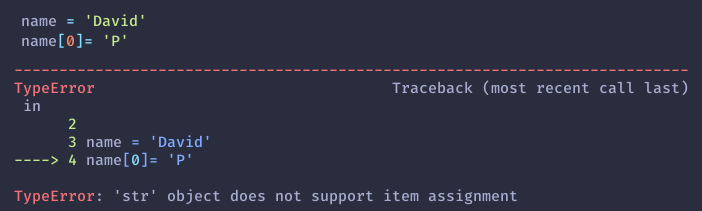
- stop no incluimos ese índice

- step tamaño del subString a generar



### Propiedades de los strings

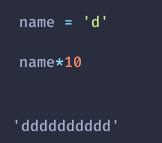
1. Son inmutables, no podemos modificar un string



1. Concatenables

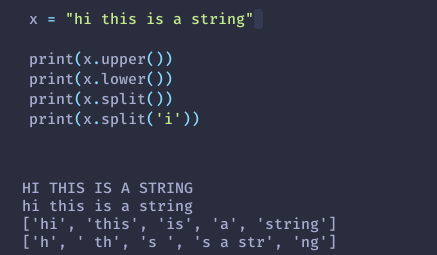


1. Multiplicables

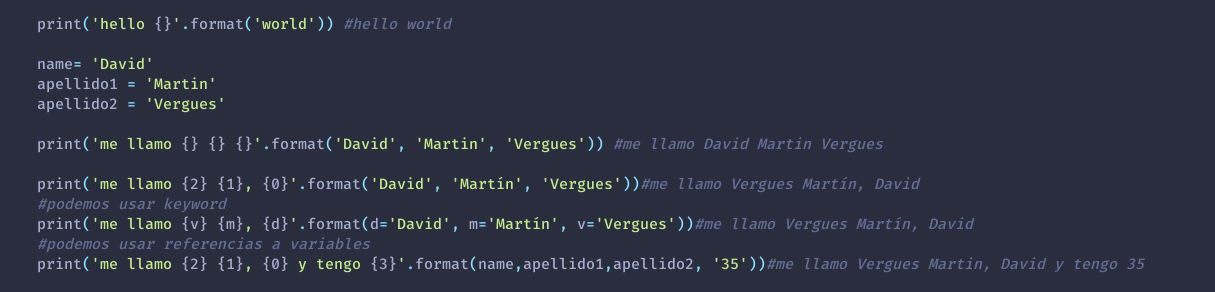


### Métodos de los strings

#### .upper() .lower() .split()



#### .format()

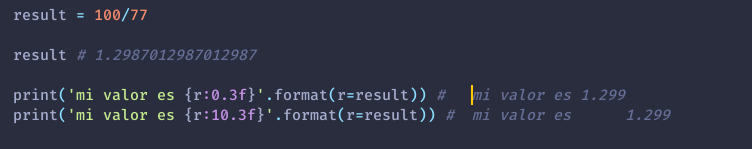


##### 2.1 Float formatting {value:width.precision f}

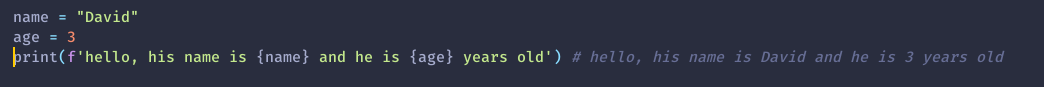
Value => ponemos el valor del número

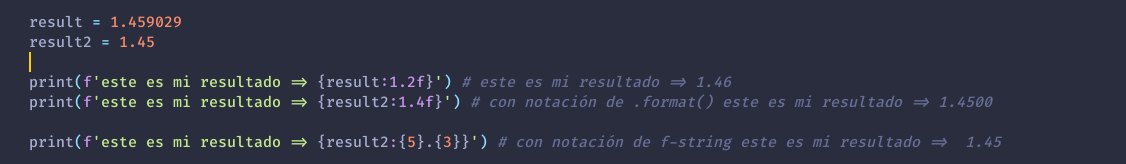
Width => tamaño que ocupará el número hecho string (nº de caracteres)

Precision => número de decimales que tendrá



#### f-string = formated string literal





En este caso hace referencia al número total de dígitos que tendrá el número, no como en .format() que hace referencia al número de dígitos decimales