**jupyter shortcuts y naming convenction**

* **control + enter**: ejecuta la celda
* **shift + enter:** ejecuta la celda y te pasa a la siguiente
* **alt + enter:** ejecuta la celda y genera una nueva en la siguiente.

comando magic de python:

* **%%time** → nos dice el tiempo que va a tardar en ejecutarse el código

## 

## **The two modes of operation**

While working on our notebooks, we will be constantly changing between two modes, the command mode and the edit mode. And the best way to move around and make life easier for ourselves is to learn a few keyboard shortcuts, or hotkeys.

## **Changing modes**

* Esc : activates command mode
* Enter : activates edit mode

## **Both Modes**

* Run commands
  + Ctrl + Enter : run selected cells
  + Shift + Enter : run current cell and select cell below
  + Alt + Enter : run current cell and insert new cell below.
* Ctrl + S : save and checkpoint

## **In command mode**

* H : show all shortcuts
* UP and DOWN : move between cells
* Shift + UP or Shift + DOWN : select cells in either direction
* A : insert new cell above
* B : insert new cell below
* C : copy selected cells
* X : cut selected cells
* V : paste cells below
* Shift + V : paste cells above
* D + D (press twice) : delete selected cells
* Z : undo cell deletion
* Y : convert current cell type to Code
* M : convert current cell type to Markdown
* P : open command pallete
* SPACE, Shift + SPACE : scroll down and up
* Shift + M : merge selected cells into one

## **In edit mode**

* Tab : autocomplete or ident
* Shift + Tab : dedent
* Ctrl + A : select all
* Ctrl + Z : undo
* Ctrl + Y : redo
* Ctrl(⌥) + LEFT or Ctrl(⌥) + RIGHT : move cursor one word left or right
* Ctrl(^) + Shift + - : split cell into two at current cursor position
* # → comenta la línea
* “”” al principio y “”” al final comenta tas las líneas que hay dentro

**NAMING CONVENTION**

* todo en inglés
* sin caracteres especiales
* minúsculas
* sin espacios (con “\_”) → sin espacios es obligatorio
* explícitas (descriptivas)
* mayúscula → SOLO para aquellas variables que esperamos que nunca cambie (ejemplo un token, email, …)
* Las clases suelen ser así → class CoolClass()
* one statement per line