



Python

2019 Proyecto Pathways PUCV

Que hemos visto hasta ahora



- En un mes ya son capaces de programar cosas sencillas.
- Hasta ahora nos hemos entretenido bastante en Scratch.
- Saben usar tal herramienta para solucionar problemas simples.
- ¿Todo se programa en Scratch?



Stage 1 backdrop

New backdrop:

Sprites

New sprite:

Drum1

Drum2

Drum-Snare

Cymbal

Drum-Bass

Scripts
Backdrops
Sounds

Motion
Looks
Sound
Pen
Data

Events
Control
Sensing
Operators
More Blocks

Make a Block

Add an Extension

Load Experimental Extension

GO:KEYS Extension

loop mix Trance

play Drums 1

stop All

key C

wait 1 measure

measure

beat

tick

key on

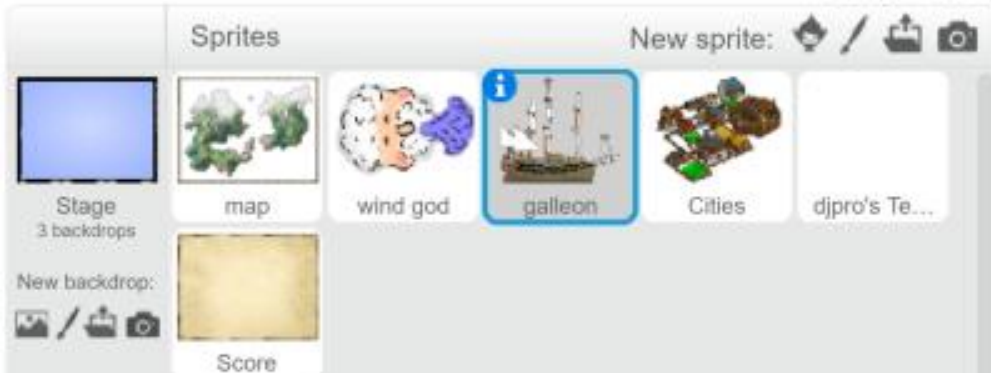
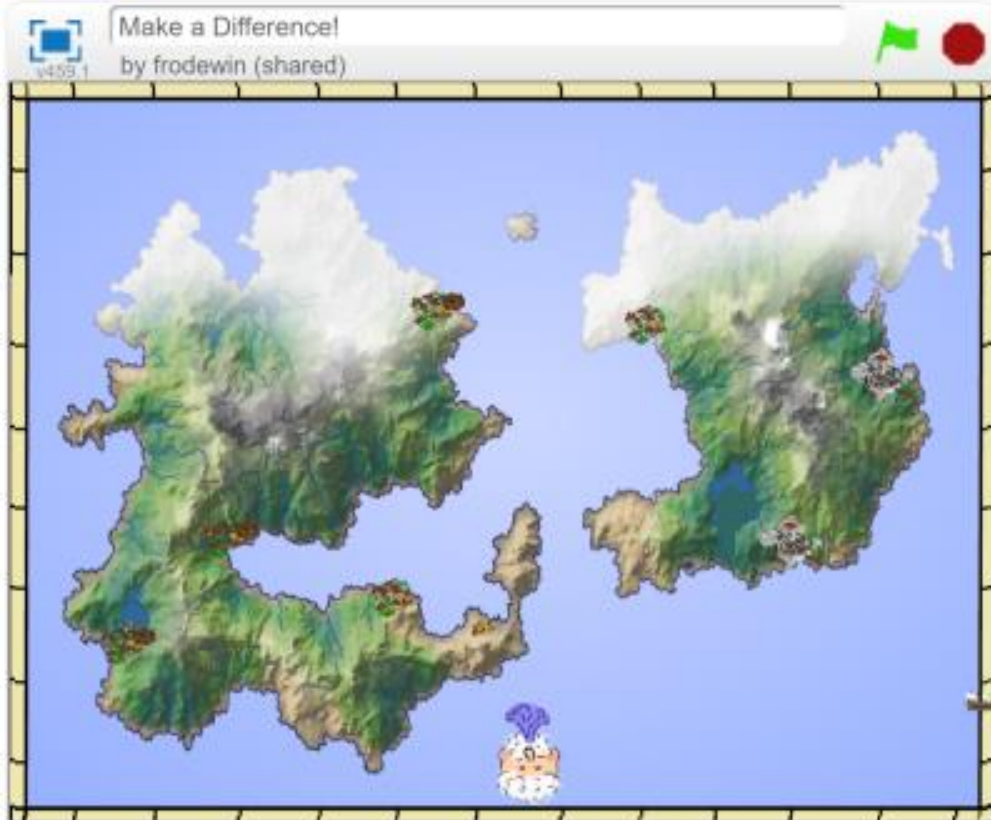
note

velocity

```

when clicked
  loop mix Pop
  key C
  play Drums 1
  wait 2 measure
  set i to 1
  repeat 8
    if measure = 3 then
      play Bass 1
    if measure = 5 then
      play Melody A 1
    if measure = 7 then

```

Scripts Costumes Sounds

Motion Looks Sound Pen Data Events Control Sensing Operators More Blocks

```

move 10 steps
turn 15 degrees
turn 15 degrees

point in direction 90
point towards mouse-pointer

go to x: 234 y: -116
go to mouse-pointer
glide 1 secs to x: 234 y: -116

change x by 10
set x to 0
change y by 10
set y to 0

if on edge, bounce

set rotation style left-right
    
```

```

when I receive Init Game
set size to 5 %
go to x: 49 y: -116
set dir to 90
set Gold to 666
point in direction 90
    
```

```

when I receive Stop Ship Movement
stop other scripts in sprite
stop all sounds
hide
    
```

```

when I receive Game Start
show
go to front
forever
control
    
```

```

when I receive Game Start
forever
play sound sailing.mp3 until done
    
```

```

when I receive Score
stop other scripts in sprite
    
```

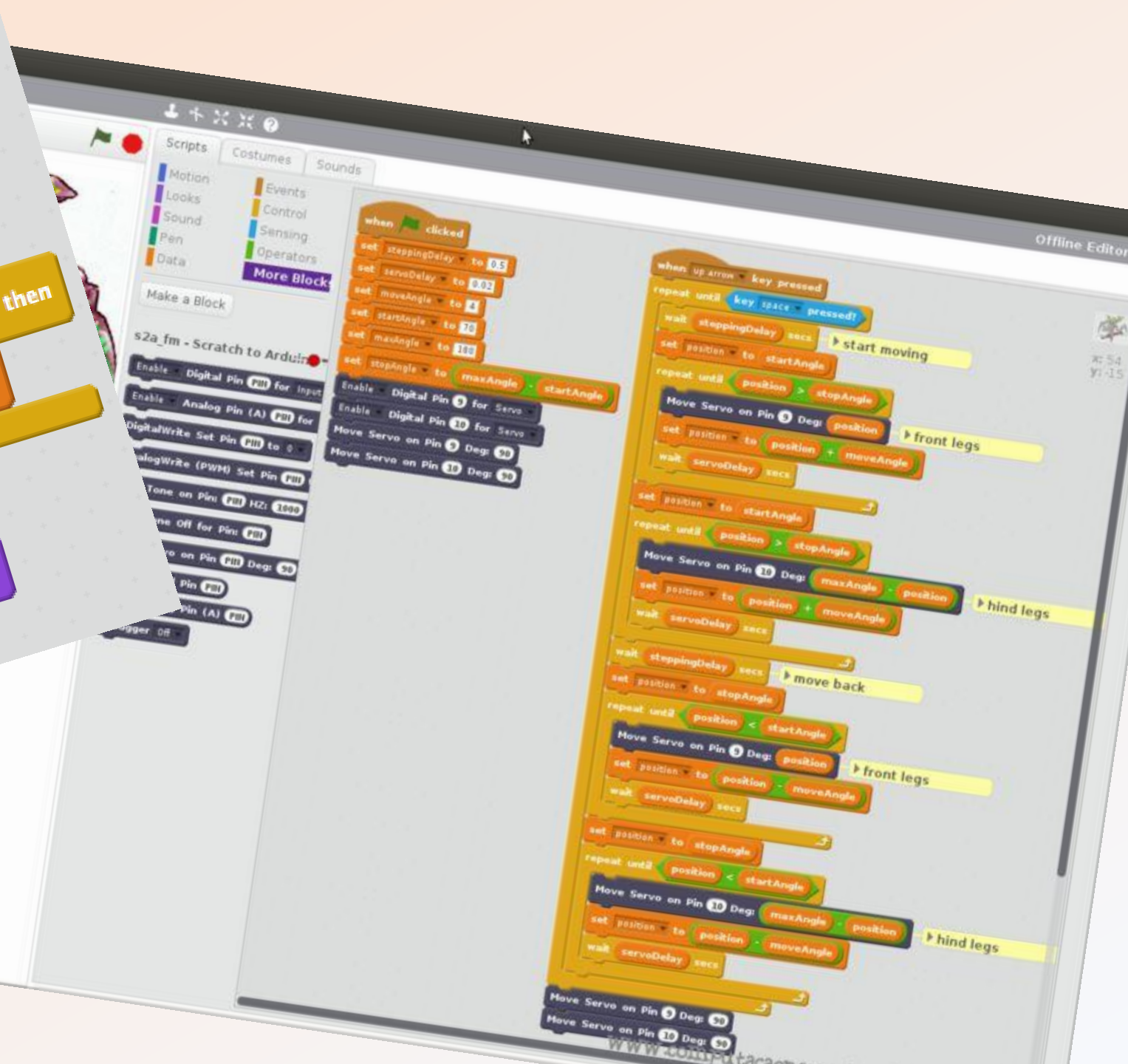
```

define control
if key right arrow pressed? then
change dir by 1
if key left arrow pressed? then
change dir by -1
set oldx to x position
set oldy to y position
set dot to windx * sin of dir +
change x by 1.1 + dot / 2 * sin of
change y by 1.1 + dot / 2 * cos of
switch costume to collision sphere
point in direction 90
if touching map? then
set x to oldx
set y to oldy
point in direction 90 + dir - round dir
    
```





COMPUTAÇÃO
NA ESCOLA



**UN MOMENTO,
TODO ESTO ES
DEMASIADO**

- Vamos con calma. Esta gente puede hacer programas gigantes porque saben ir paso a pasito y resolver problemas, primero que todo.
- Al final no importa si usas Scratch, o algún otro programa mucho mas feo. Si sabes resolver problemas, el programa que uses para programar es solo un detalle.


```
g++ -std=c++11 -c HighLevelAssembly.cpp
```

```
/* Entry Point */
```

```
program: (opcode | NEWLINE)+;
```

```
/* Opcode Rules */
```

```
/* Naming convention:
```

```
=> "I": Indirect Addressing
```

```
=> "IPO": Indirect Addressing - positive Offset
```

```
=> "IPRO": Indirect Addressing - positive Register Offset
```

```
=> "INO": Indirect Addressing - negative Offset
```

```
=> "INRO": Indirect Addressing - negative Register Offset
```

```
*/
```

```
opcode:
```

```
/* MOV opcodes for 8-bit registers */
```

```
MOV register_8bit ',' register_8bit
```

```
| MOV register_8bit ',' '[' register_16bit ']'
```

```
| MOV register_8bit ',' '[' register_16bit plus_sign int ']'
```

```
| MOV register_8bit ',' '[' register_16bit plus_sign register_8bit ']'
```

```
| MOV register_8bit ',' '[' register_16bit minus_sign int ']'
```

```
| MOV register_8bit ',' '[' register_16bit minus_sign register_8bit ']'
```

```
| MOV register_8bit ',' int
```

```
| MOV register_8bit ',' EightBitBinaryValue
```

```
| MOV register_8bit ',' '0x' HexValue
```

```
| MOV register_8bit ',' HexValue 'h'
```

```
| MOV '[' register_16bit plus_sign int ']' ',' register_8bit
```

```
| MOV '[' register_16bit plus_sign int ']' ',' int
```

```
| MOV '[' register_16bit plus_sign int ']' ',' EightBitBinaryValue
```

```
| MOV '[' register_16bit plus_sign int ']' ',' '0x' HexValue
```

```
| MOV '[' register_16bit plus_sign int ']' ',' HexValue 'h'
```

```
| MOV '[' register_16bit plus_sign register_8bit ']' ',' register_8bit
```

```
| MOV '[' register_16bit plus_sign register_8bit ']' ',' int
```

```
| MOV '[' register_16bit plus_sign register_8bit ']' ',' EightBitBinaryValue
```

```
| MOV '[' register_16bit plus_sign register_8bit ']' ',' '0x' HexValue
```

```
| MOV '[' register_16bit plus_sign register_8bit ']' ',' HexValue 'h'
```

```
# MOV_R8_R8
```

```
# MOV_I_R8_R16
```

```
# MOV_IPO_R8_R16
```

```
# MOV_IPRO_R8_R16
```

```
# MOV_INO_R8_R16
```

```
# MOV_INRO_R8_R16
```

```
# MOV_CONST_DECIMAL
```

```
# MOV_CONST_BINARY
```

```
# MOV_CONST_HEX1
```

```
# MOV_CONST_HEX2
```

```
# MOV_IPO_R16_R8
```

```
# MOV_IPO_R16_CONST_DECIMAL
```

```
# MOV_IPO_R16_CONST_BINARY
```

```
# MOV_IPO_R16_CONST_HEX1
```

```
# MOV_IPO_R16_CONST_HEX2
```

```
# MOV_IPRO_R16_R8
```

```
# MOV_IPRO_R16_CONST_DECIMAL
```

```
# MOV_IPRO_R16_CONST_BINARY
```

```
# MOV_IPRO_R16_CONST_HEX1
```

```
# MOV_IPRO_R16_CONST_HEX2
```

```
/* MOV D, E */
```

```
/* MOV D, [SP] */
```

```
/* MOV D, [SP + 1] */
```

```
/* MOV D, [SP + 6] */
```

```
/* MOV D, [SP - 1] */
```

```
/* MOV D, [SP - 6] */
```

```
/* MOV D, 4 */
```

```
/* MOV D, 10101010 */
```

```
/* MOV D, 0xAB */
```

```
/* MOV D, ABh */
```

```
/* MOV [SP + 1], D */
```

```
/* MOV [SP + 1], 3 */
```

```
/* MOV [SP + 1], 10101010b */
```

```
/* MOV [SP + 1], 0xAB */
```

```
/* MOV [SP + 1], ABh */
```

```
/* MOV [SP + 6], D */
```

```
/* MOV [SP + 6], 3 */
```

```
/* MOV [SP + 6], 10101010b */
```

```
/* MOV [SP + 6], 0xAB */
```

```
/* MOV [SP + 6], ABh */
```



```
gsm_asm HighLevelAssembly;
```

```
/* Entry Point */  
program: (opcode | NEWLINE)+;
```

```
/* Opcode Rules */
```

```
/* Naming convention:
```

- => "I": Indirect Addressing
- => "IPO": Indirect Addressing - positive Offset
- => "IPRO": Indirect Addressing - positive Register Offset
- => "INO": Indirect Addressing - negative Offset
- => "INRO": Indirect Addressing - negative Register Offset

```
*/
```

```
opcode:
```

```
/* MOV opcodes for 8bit register */  
MOV register_8bit register_8bit # MOV_R8_R8 /* MOV D, E */  
| MOV register_8bit '[' register_16bit ']' # MOV_I_R8_R16 /* MOV D, [SP] */  
| MOV register_8bit '[' register_16bit plus_sign int ']' # MOV_IPO_R8_R16 /* MOV D, [SP + 1] */  
| MOV register_8bit '[' register_16bit plus_sign register_8bit ']' # MOV_IPRO_R16 /* MOV D, [SP + G] */  
| MOV register_8bit '[' register_16bit minus_sign int ']' # MOV_INO_R16 /* MOV D, [SP - 1] */  
| MOV register_8bit '[' register_16bit minus_sign register_8bit ']' # MOV_INRO_R16 /* MOV D, [SP - G] */  
  
| MOV register_8bit 'int' # MOV_CONST_R8_DECIMAL /* MOV D, 4 */  
| MOV register_8bit 'EightBitBinaryValue' # MOV_CONST_R8_BINARY /* MOV D, 10101010 */  
| MOV register_8bit '0x' HexValue # MOV_CONST_R8_HEX1 /* MOV D, 0xAB */  
| MOV register_8bit 'HexValue' HexValue # MOV_CONST_R8_HEX2 /* MOV D, ABh */  
  
| MOV '[' register_16bit plus_sign int ']' register_8bit # MOV_IPO_R16_R8 /* MOV [SP + 1], D */  
| MOV '[' register_16bit plus_sign int ']' int # MOV_IPO_R16_CONST_DECIMAL /* MOV [SP + 1], 3 */  
| MOV '[' register_16bit plus_sign int ']' 'EightBitBinaryValue' # MOV_IPO_R16_CONST_BINARY /* MOV [SP + 1], 10101010b */  
| MOV '[' register_16bit plus_sign int ']' '0x' HexValue # MOV_IPO_R16_CONST_HEX1 /* MOV [SP + 1], 0xAB */  
| MOV '[' register_16bit plus_sign int ']' HexValue 'h' # MOV_IPO_R16_CONST_HEX2 /* MOV [SP + 1], ABh */  
  
| MOV '[' register_16bit plus_sign register_8bit ']' register_8bit # MOV_IPRO_R16_R8 /* MOV [SP + G], D */  
| MOV '[' register_16bit plus_sign register_8bit ']' int # MOV_IPRO_R16_CONST_DECIMAL /* MOV [SP + G], 3 */  
| MOV '[' register_16bit plus_sign register_8bit ']' 'EightBitBinaryValue' # MOV_IPRO_R16_CONST_BINARY /* MOV [SP + G], 10101010b */  
| MOV '[' register_16bit plus_sign register_8bit ']' '0x' HexValue # MOV_IPRO_R16_CONST_HEX1 /* MOV [SP + G], 0xAB */  
| MOV '[' register_16bit plus_sign register_8bit ']' HexValue 'h' # MOV_IPRO_R16_CONST_HEX2 /* MOV [SP + G], ABh */
```

gasmem HighLevelAssembly;

```
/* Entry Point */
program: (opcode | NEWLINE)+;
```

```
/* Opcode Rules */
```

```
/* Naming convention:
```

```
=> "I":      Indirect Addressing
=> "IPO":     Indirect Addressing
=> "IPRO":    Indirect Addressing
=> "INO":     Indirect Addressing
=> "INRO":    Indirect Addressing
```

```
*/
```

```
opcode:
```

```
/* MOV opcodes for 8bit
```

```
MOV register_8bit, register_8bit
| MOV register_8bit, register_8bit
| MOV register_8bit, register_8bit
| MOV register_8bit, register_8bit
| MOV register_8bit, register_8bit
| MOV register_8bit, register_8bit
```

```
| MOV register_8bit, register_8bit
| MOV register_8bit, register_8bit
| MOV register_8bit, register_8bit
| MOV register_8bit, register_8bit
```

```
| MOV '[' register_8bit, register_8bit
```

```
| MOV '[' register_8bit, register_8bit
```

```
| MOV '[' register_16bit plus_sign, register_8bit
| MOV '[' register_16bit plus_sign, register_8bit
| MOV '[' register_16bit plus_sign int ']' ,
```

```
| MOV '[' register_16bit plus_sign register_8bit ']' , register_8bit
| MOV '[' register_16bit plus_sign register_8bit ']' , int
| MOV '[' register_16bit plus_sign register_8bit ']' , EightBitBinaryValue
| MOV '[' register_16bit plus_sign register_8bit ']' , '0x' HexValue
| MOV '[' register_16bit plus_sign register_8bit ']' , HexValue 'h'
```

```
MOV D, E */
```

```
MOV D, [SP] */
```

```
/* MOV D, [SP + 1] */
```

```
/* MOV D, [SP + 6] */
```

```
/* MOV D, [SP - 1] */
```

```
/* MOV D, [SP - 6] */
```

```
/* MOV D, 4 */
```

```
/* MOV D, 10101010 */
```

```
/* MOV D, 0xAB */
```

```
/* MOV D, ABh */
```

```
/* MOV [SP + 1], D */
```

```
/* MOV [SP + 1], 3 */
```

```
/* MOV [SP + 1], 10101010b */
```

```
/* MOV [SP + 1], 0xAB */
```

```
/* MOV [SP + 1], ABh */
```

```
/* MOV [SP + 6], D */
```

```
/* MOV [SP + 6], 3 */
```

```
/* MOV [SP + 6], 10101010b */
```

```
/* MOV [SP + 6], 0xAB */
```

```
/* MOV [SP + 6], ABh */
```

```
#include <stdio.h>
#include <unistd.h>

int main ()
{
    printf("Running 'net join' with the following parameters: \n");
    char *domain="mydomain";
    char *user="domainjoinuser";
    char *pass="mypassword";
    char *vastool="/opt/quest/bin/vastool";
    char *ou="OU=test,DC=mtdomian,DC=local";
    char unjoin[512];

    sprintf(unjoin, "/opt/quest/bin/vastool -u %s -w '%s' unjoin -f", user, pass);

    printf("Domain: %s\n", domain);
    printf("User: %s\n", user);
    printf("-----\n");

    printf("\nUnjoin.....\n");
    system(unjoin);

    printf("\nJoin.....\n");
    execl("/opt/quest/bin/vastool", "vastool", "-u", user, "-w", pass, "join", "-c", ou,
"-f", domain, (char*)0);
}
```



```
#include <stdio.h>
#include <unistd.h>

int main ()
{
    printf("Running 'net join' with the following parameters: \n");
    char *domain="mydomain";
    char *user="domainjoinuser";
    char *pass="mypassword";
    char *vastool="/opt/quest/bin/vastool";
    char *ou="OU=Test,DC=ntdomain,DC=oc1";
    char unjoin[121];

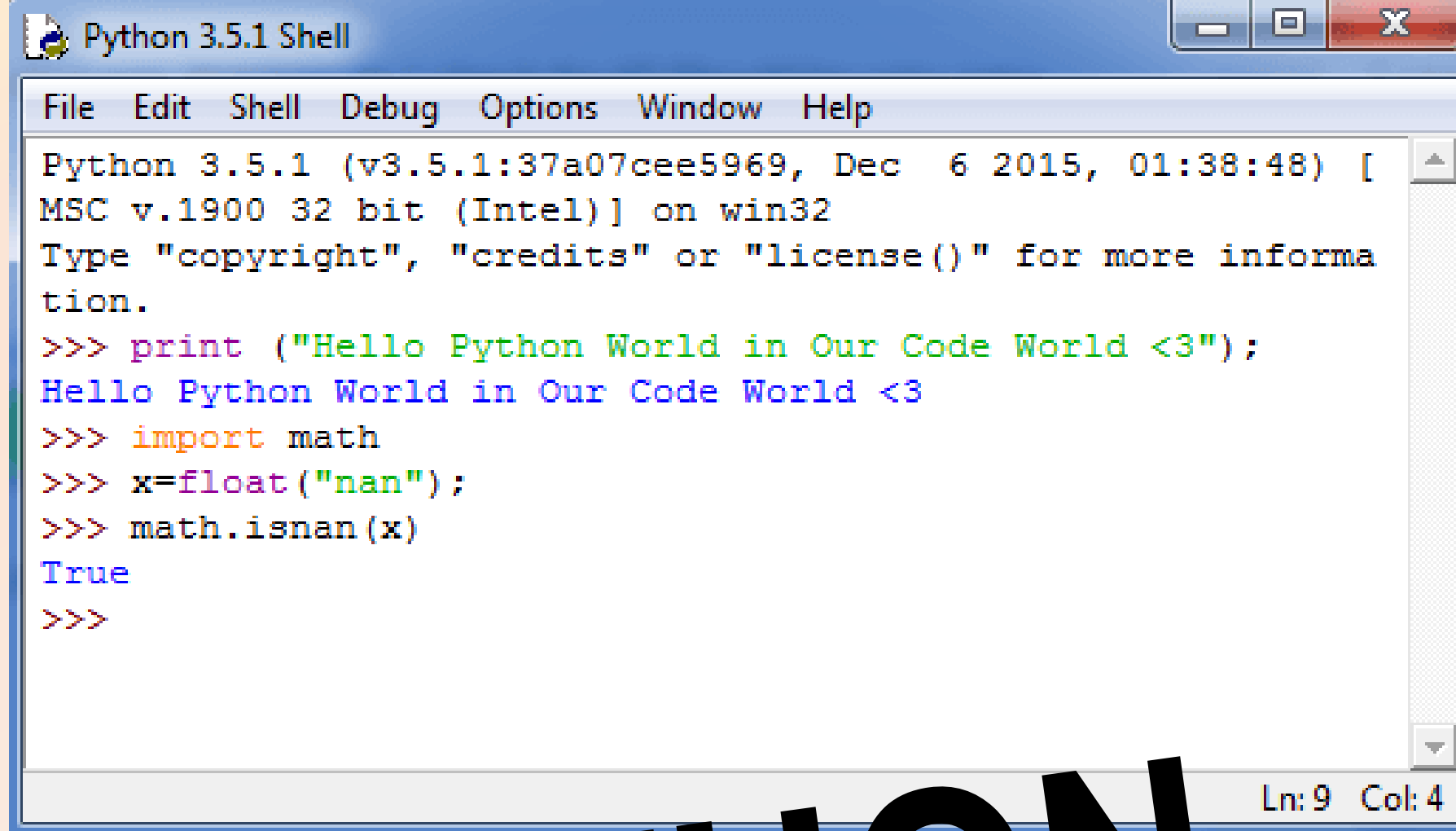
    sprintf(unjoin, "%s %s -u %s -w '%s' ujoin -f %s", vastool, user, user, pass, domain);

    printf("Domain: %s\n", domain);
    printf("User: %s\n", user);
    printf("-----\n");

    printf("\nUnjoin.....\n");
    system(unjoin);

    printf("\nJoin.....\n");
    execl("/opt/quest/bin/vastool", "vastool", "-u", user, "-w", pass, "join", "-c", ou,
"-f", domain, (char*)0);
}
```

LENGUAJES DE PROGRAMACION

A screenshot of a Python 3.5.1 Shell window. The window has a blue title bar with the text "Python 3.5.1 Shell" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with the following options: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains a text editor with the following text:

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [
MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informa
tion.
>>> print ("Hello Python World in Our Code World <3");
Hello Python World in Our Code World <3
>>> import math
>>> x=float("nan");
>>> math.isnan(x)
True
>>>
```

The text is color-coded: "Python 3.5.1" is blue, "MSC v.1900 32 bit (Intel)] on win32" is black, "Type "copyright", "credits" or "license()" for more informa" is black, "tion." is black, ">>> print ("Hello Python World in Our Code World <3");" is black, "Hello Python World in Our Code World <3" is blue, ">>> import math" is black, ">>> x=float("nan");" is black, ">>> math.isnan(x)" is black, "True" is blue, and ">>>" is black. On the right side of the text editor, there is a vertical scrollbar. At the bottom of the window, there is a status bar that says "Ln: 9 Col: 4".

Python 3.5.1 Shell

File Edit Shell Debug Options Window Help

Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

```
>>> print ("Hello Python World in Our Code World <3");
Hello Python World in Our Code World <3
>>> import math
>>> x=float("nan");
>>> math.isnan(x)
True
>>>
```

Ln: 9 Col: 4

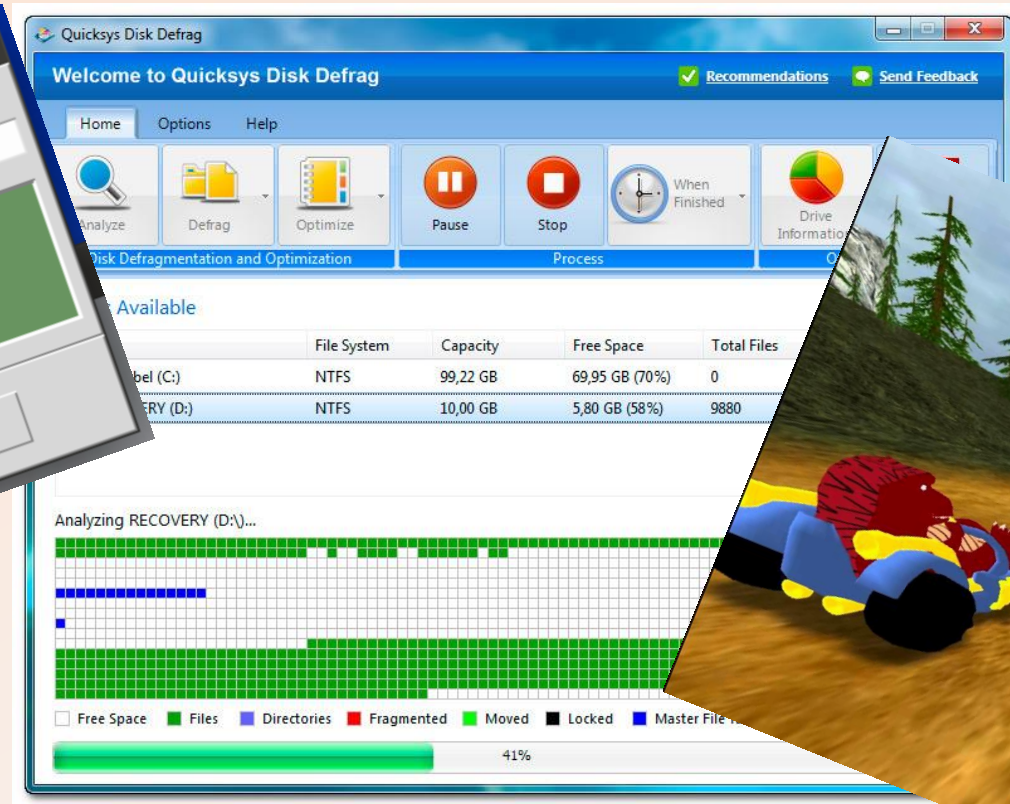
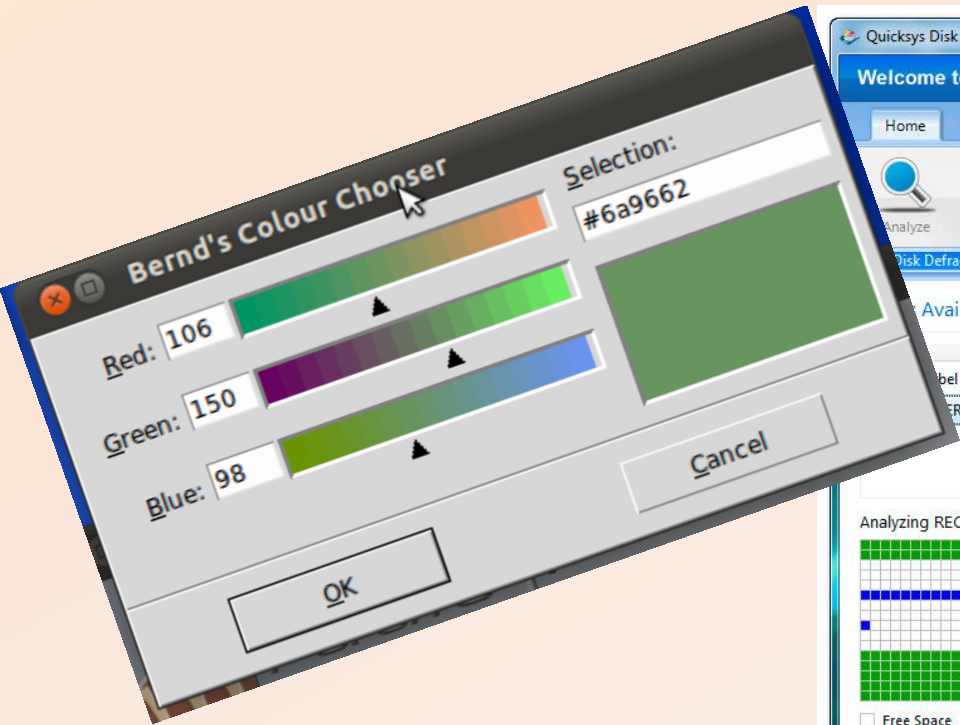
PYTHON

¿Python?

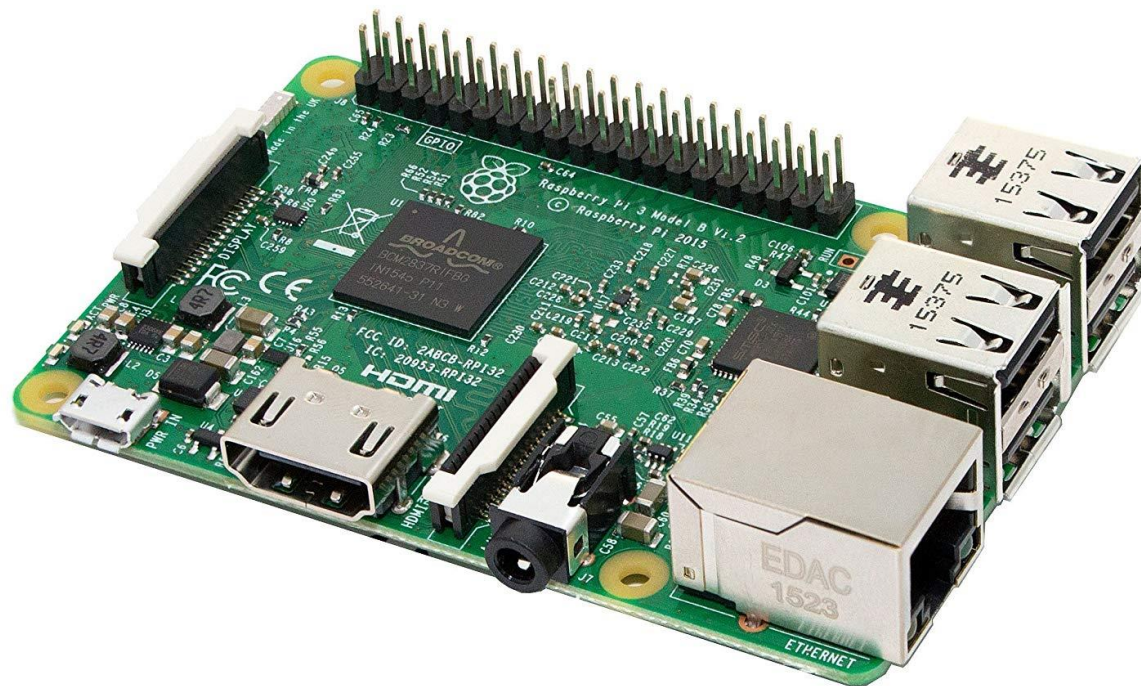
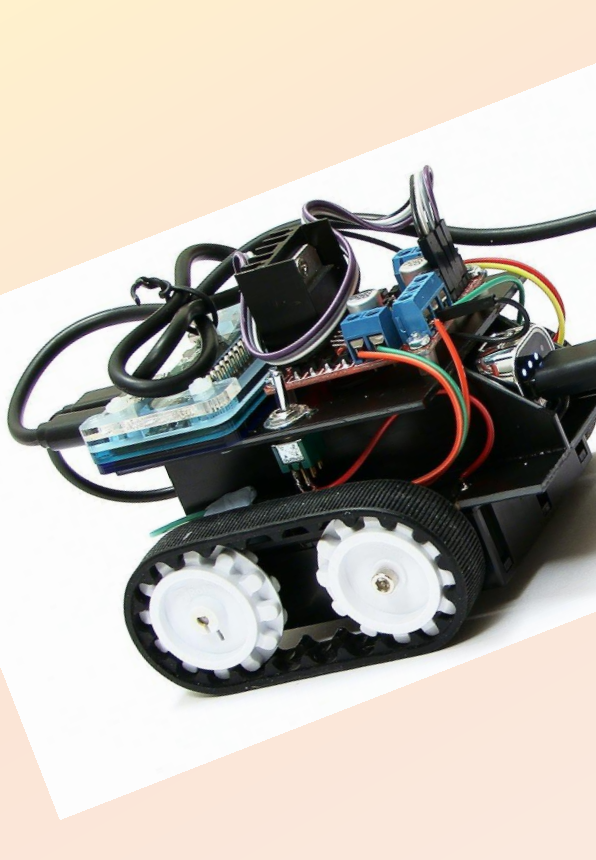


- Si, Python.
- Python es uno mas de esos lenguajes de programación de todo este universo de programas.
- Si, sirve para programar computadores, al igual que esos otros lenguajes feos, ¡y al igual que Scratch!
- Es simple, pero MUY poderoso.

¿Qué se puede hacer en Python?

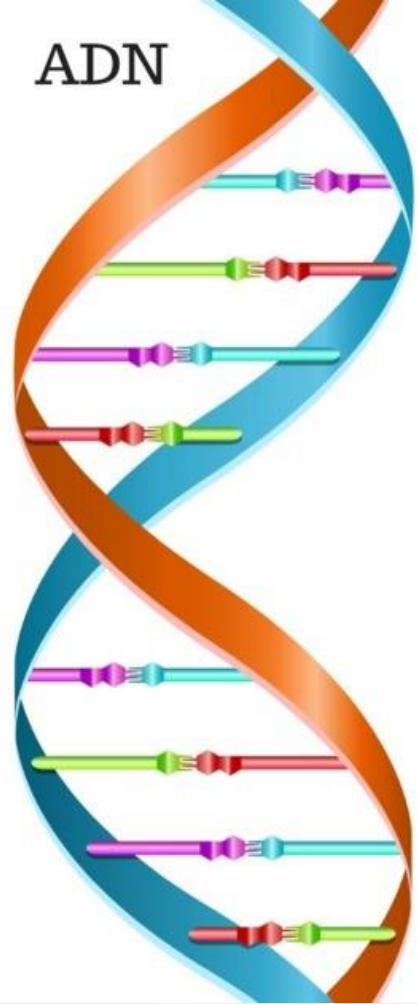


Toda clase de programas

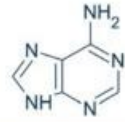
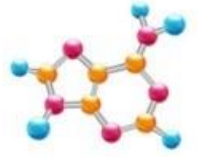


Crear y programar aparatos

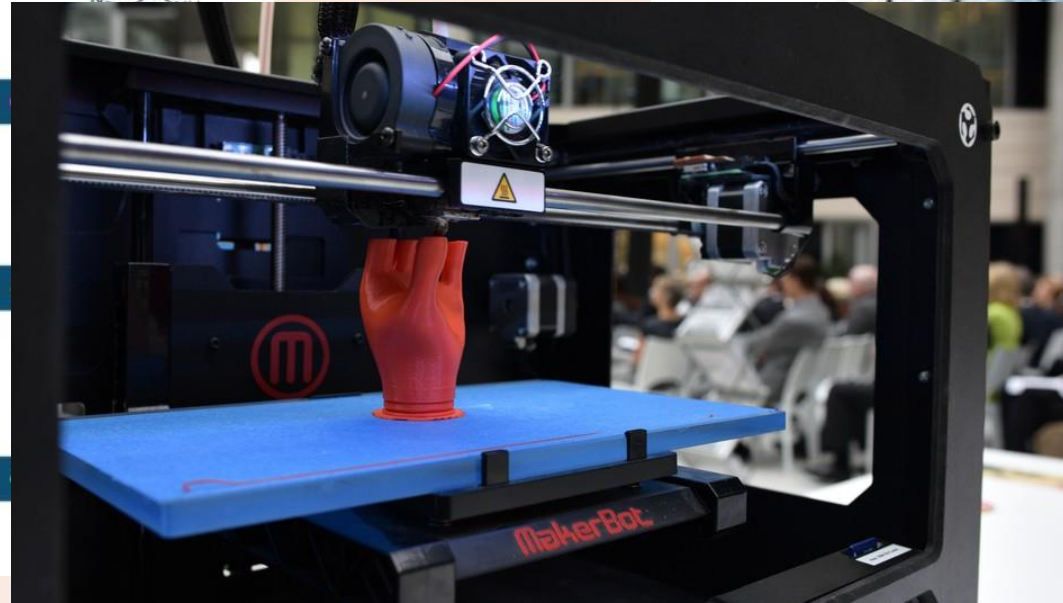
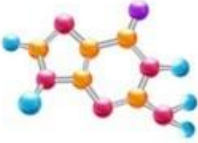
ADN



Bases nitrogenadas

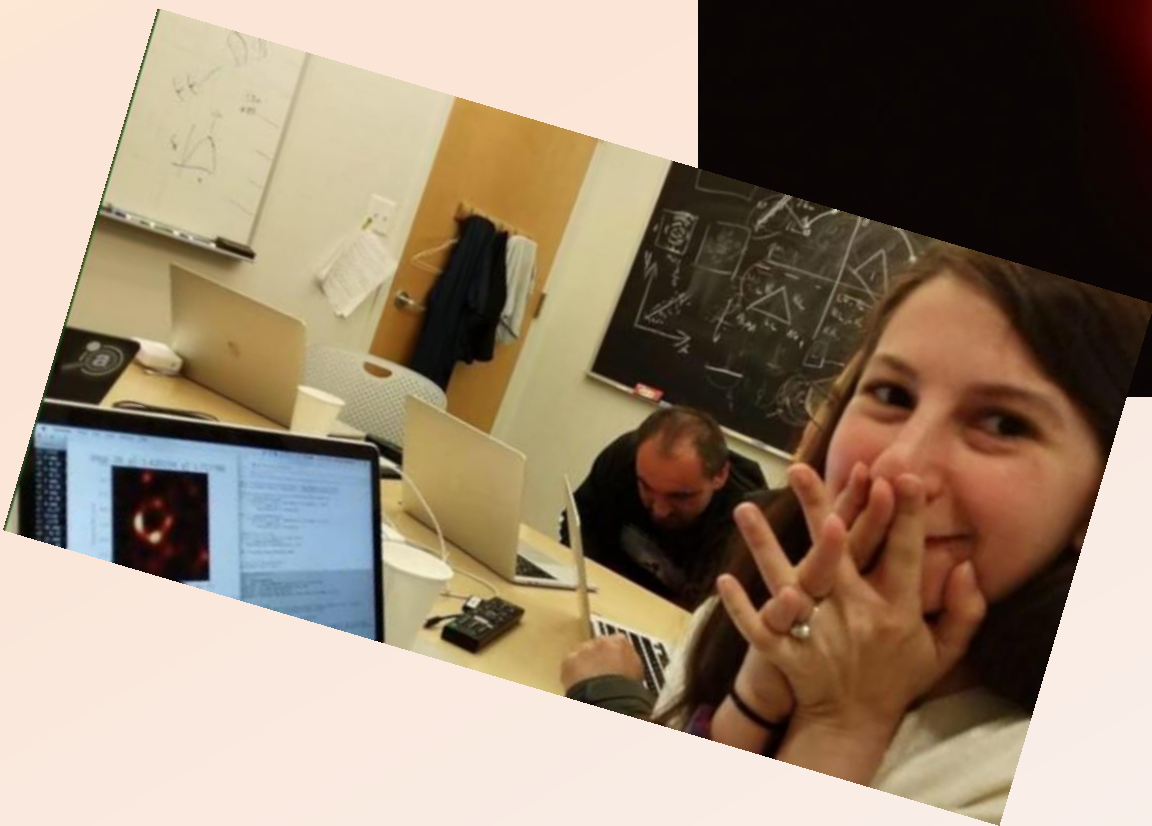


Adenina



¡Hacer ciencia!





**¡La imagen del agujero negro se sacó con un programa
hecho en Python!**

**Y MUCHO MUCHO
MUCHO MAS**

Hey, eso suena interesante, pero ... ¿Cómo programo en Python?

- Pues escribiendo programas como estos:

```
# define a function
def lcm(x, y):
    """This function takes two
    integers and returns the L.C.M."""

    # choose the greater number
    if x > y:
        greater = x
    else:
        greater = y

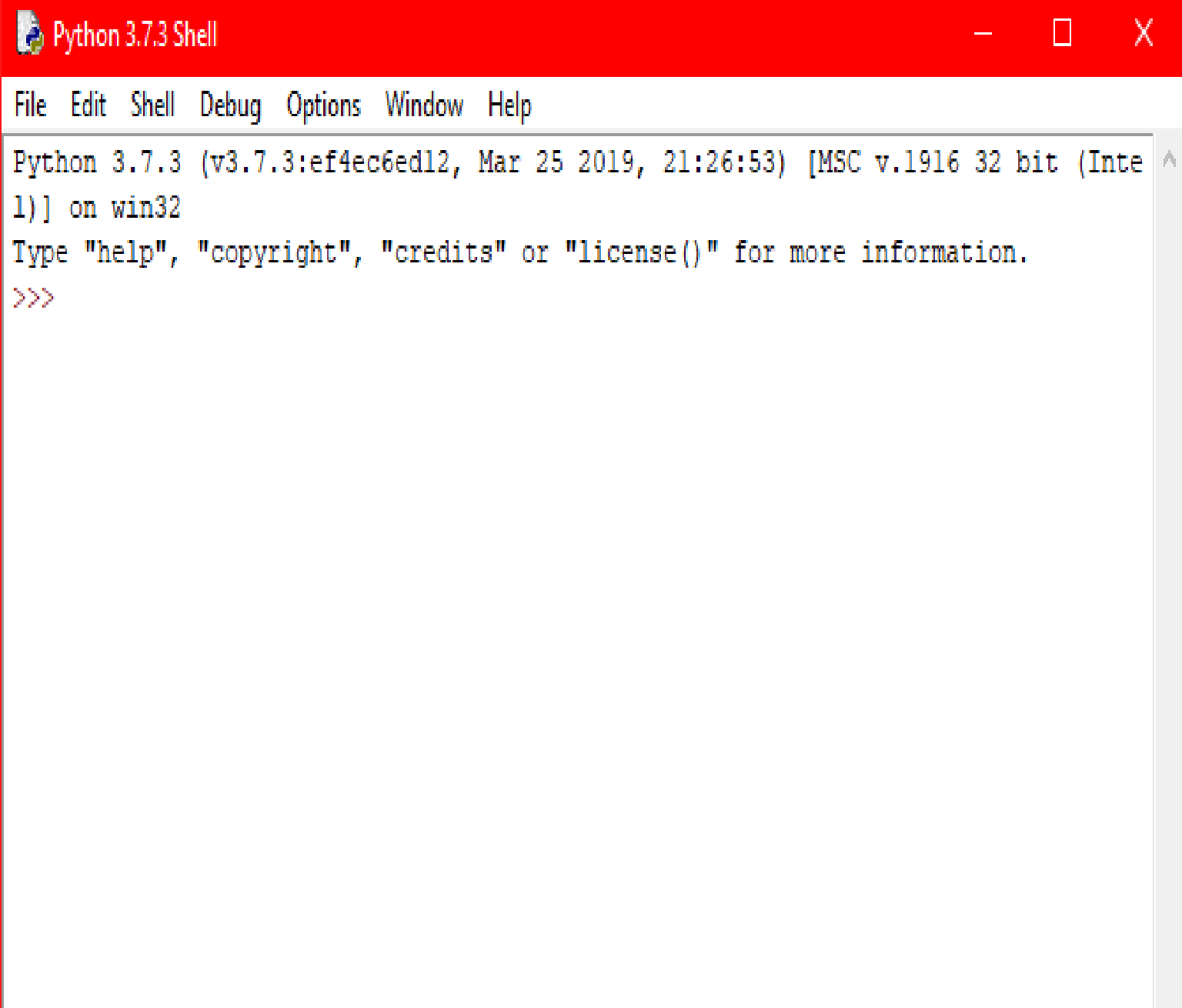
    while(True):
        if((greater % x == 0) and (greater % y == 0)):
            lcm = greater
            break
        greater += 1

    return lcm
```

**No te asustes, con calma.
¡Iremos viendo esto poco a
poco!**

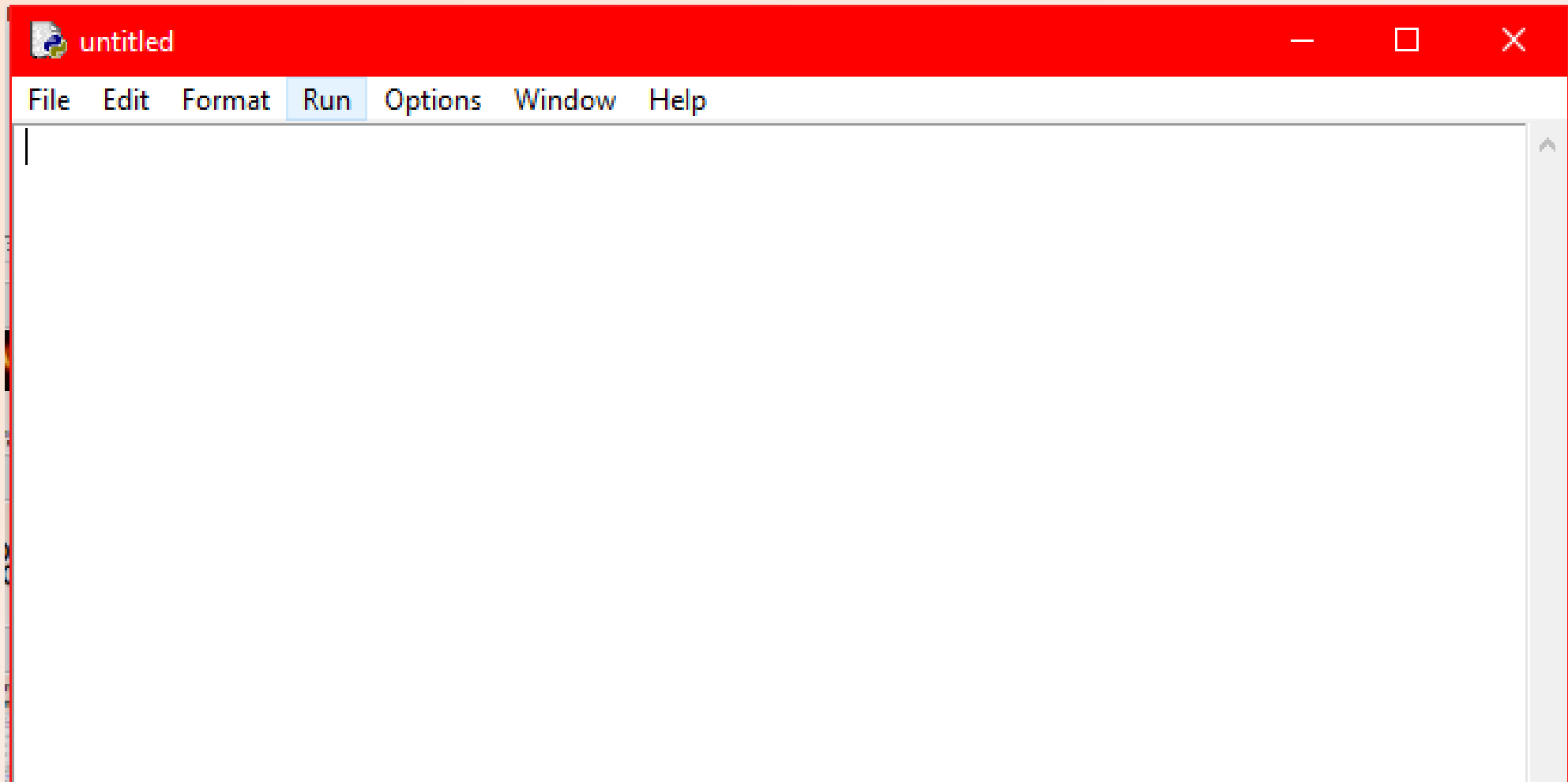
Primero necesitaremos un editor de Python.

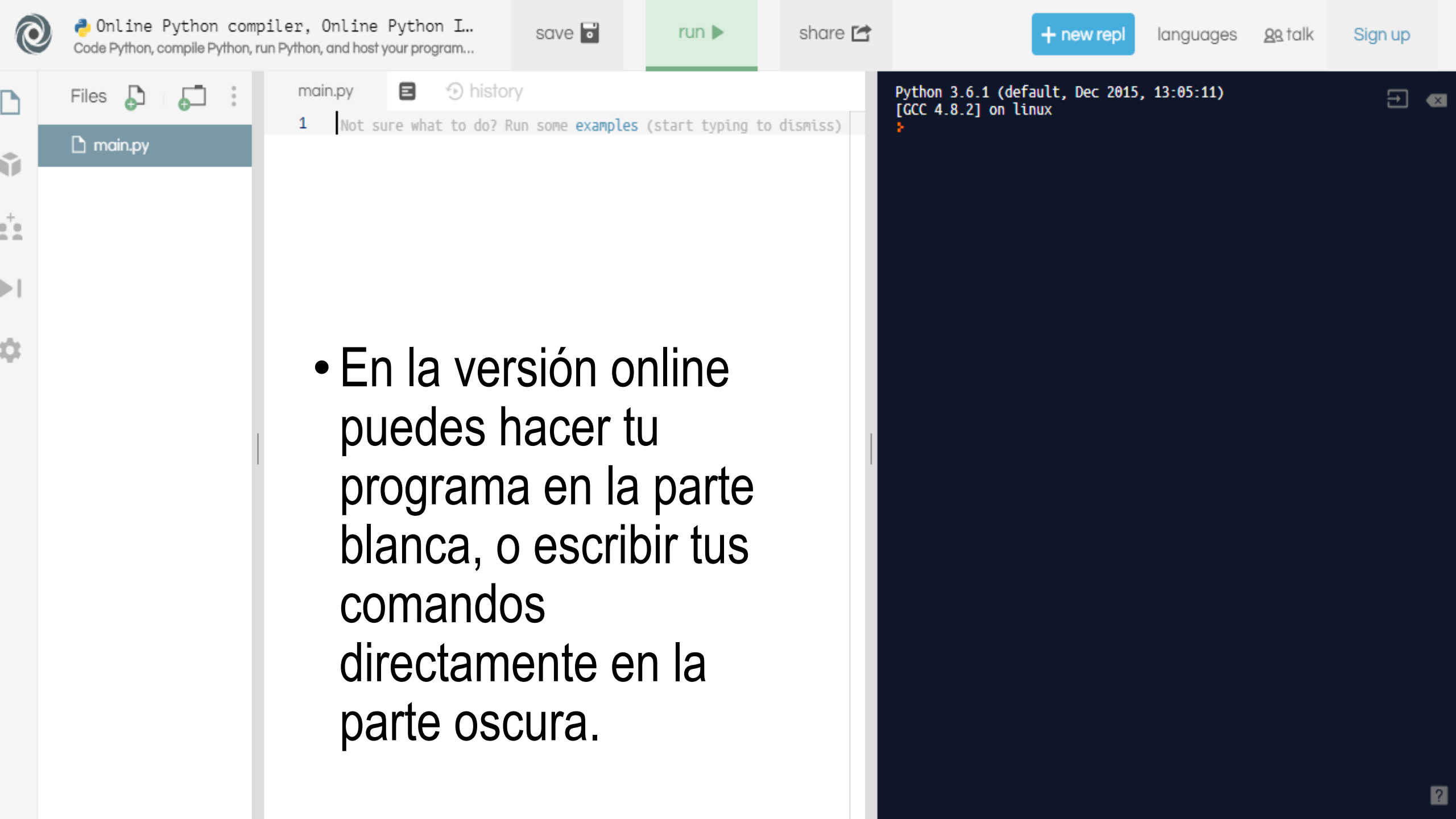
- Si estas en tu casa con tu computador, puedes bajar e instalar Python 3 desde aquí: <https://www.python.org/>
- ¡Aquí en la sala de clases Python ya esta instalado! (O lo van a instalar si es que todavía no). Busca un programa llamado IDLE.
- ¡Si no tienes Python instalado, puedes ir a esta pagina para programar online! <https://repl.it/languages/python3>



```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

- En IDLE puedes escribir directamente comandos en esta ventana, o si bien, si quieres crear un programa, ve a **File > New File**.





- En la versión online puedes hacer tu programa en la parte blanca, o escribir tus comandos directamente en la parte oscura.

¿Y ahora que?

¡Nuestro primer programa!

- Python es grande, abordemos lo mas básico: Hacer que Python nos salude.
- Para que Python nos diga “Hola mundo”, deberás escribir este comando en tu editor.

```
print(`¡Hola mundo!`)
```

Ahora dale a Play en tu editor (Run-> Run Module (F5) en IDLE, o el botón de Play en el editor online).

RESTA

¡Hola mundo!

>>>

```
Python 3.6.1 (default, Dec 2015, 13:05:11)  
[GCC 4.8.2] on linux  
¡Hola mundo!  
+ 
```


**Ahora pon mas “*prints*”
abajo, pero con otras
frases.**

```
print(";Hola mundo!")  
print("¿Como estas?")  
print("¿Falta mucho para el break?")  
print("Voy a hacer un salto de linea\nEsto es un salto de linea")|
```

```
¡Hola mundo!  
¿Como estas?  
¿Falta mucho para el break?  
Voy a hacer un salto de linea  
Esto es un salto de linea  
>>>
```

En fondo, tu computador lee el código que acabas de hacer, y cuando encuentra un “Print”, muestra en pantalla lo que tiene dentro de print.

Estas cosas, se llaman comandos, y es el elemento principal de un programa.

¿Cómo así, entonces tu le das comandos y Python las ejecuta por ti?

- ¡Precisamente!
- Pero... No creo que hayan sacado la foto del agujero negro solo con puros “prints”. Hay muchos, muchos mas comandos que existen acá.

```
url = "http://ajax.googleapis.com/ajax/services/search/web?v=1.0&"
query = raw_input("What do you want to search for ? >> ")
query = urllib.urlencode( {'q' : query } )
response = urllib2.urlopen (url + query ).read()
data = json.loads ( response )
results = data [ 'responseData' ] [ 'results' ]

for result in results:
    title = result['title']
    url = result['url']
    print ( title + ' ; ' + url )
```

- Puedes hacer que Python te pregunte algo, como tu nombre, y que después lo use, como por ejemplo, para saludarte.

```
print("¡Hola! ¿Como te llamas?")  
nombre = input()  
print("Mucho gusto ", nombre)
```

¿Te acuerdas de las variables?

```
¡Hola! ¿Como te llamas?  
Juan Perez  
Mucho gusto Juan Perez  
~ ~ ~ |
```



```
print("¡Hola! ¿Como te llamas?")
```

```
nombre = input()
```

```
print("Mucho gusto ", nombre)
```

- Mira bien lo que se hace acá. Nos está preguntando nuestro nombre.
- Cuando contestamos (por ejemplo Juan Perez), nuestro nombre el computador lo guardará en una casilla que se llama **nombre**.
- Y al final, cuando dice “Mucho gusto”, usa esa casilla donde esta guardado nuestro nombre, y muestra **“Mucho gusto Juan Perez”**.
- Esas casillas, son las famosas variables que hablábamos las clases pasadas.

```
print("¿Cual es tu nombre?")
nombre = input()
print("¿y tu edad?")
edad = input()
print("¿y tu helado favorito?")
helado = input()
print("Tu nombre es ", nombre, ". Tienes ", edad, " años y te gusta el helado de " , helado)
|
```

- ¿Cual creen que será el resultado de este programa?

```
¿Cual es tu nombre?
```

```
Pedro Ulloa
```

```
¿y tu edad?
```

```
18
```

```
¿y tu helado favorito?
```

```
Frutilla
```

```
Tu nombre es  Pedro Ulloa . Tienes  18  años y te gusta el helado de  Frutilla
```

```
>>> |
```

```
print("Dime un color. Recuerdalo")
color = input()
print("Ahora dime otro color. Recuerdalo tambien")
color = input()
print("El color que recuerdo es ", color)
|
```

- ¿Y de este programa?

```
print("Dime un color. Recuerda")  
color = input()  
print("Ahora dime otro color. Recuerdalo")  
color = input()
```

```
| Dime un color. Recuerdalo  
| Cafe  
| Ahora dime otro color. Recuerdalo tambien  
| Naranja  
| El color que recuerdo es Naranja
```

- ¿Y de este programa?


```
print("Dime el nombre de tu mascota")
mascota = input()
print("Tu mascota se llama", mascota, "y es de raza ", raza)
|
```

- ¿Y de este otro?

```
print("Dime el nombre de tu mascota")
mascota = input()
print("Tu mascota se llama", mascota, "y es de raza ", raza)
|
```

- Opcion A: ¿Muestra “raza”?
- Opcion B: ¿El programa falla olímpicamente?
- Opcion C: ¿Aparece Ricardo Milos bailando en tu pantalla?



Tu mascota se llama Firulais y es de raza {raza}

>>>

```
Dime el nombre de tu mascota
```

```
Firulaais
```

```
Traceback (most recent call last):
```

```
  File "C:/Users/elsem/studio/progra_colegios/test.py", line 3, in <module>
```

```
    print("Tu mascota se llama", mascota, "y es de raza ", raza)
```

```
NameError: name 'raza' is not defined
```

**¡Oh no! Tu programa falló (“se cayó”).
Intento buscar la variable “raza” que no existe.**

```
print("Dime un numero")
numero1 = input()
print("Dime ahora otro numero")
numero2 = input()
resultado = numero1 + numero2
print("La suma de los 2 numeros es ", resultado)
```

¿Si pregunto por 2 números, podré sumarlos?
¿Cuanto es el resultado de esto?

```
Dime un numero  
4  
Dime ahora otro numero  
2  
La suma de los 2 numeros es 42
```

• ¿Qué demonios?



Recuerda: Los computadores son tontos.

- Python cree que esos números son palabras. No sabe que son números.
- Entonces al sumar, Python cree que estas sumando 2 palabras (rompe + cabezas = rompecabezas).
- Tienes que decirle a Python especialmente que cada una de esas variables son números. ¿Como?

```
numero = int(palabra)
```

- Al igual que **print**, el comando **int** hace algo, y **transforma palabras en números**.
- Si le pasas una palabra que dice “3”, guardará en la variable **numero** un numero 3.
- ¿Porque se llama **int**? **int** viene del ingles **integer (entero)**.

```
print("Dime un numero")

#preguntamos por una palabra y lo transformamos a numero
palabral = input()
numero1 = int(palabral)

print("Dime ahora otro numero")

#lo mismo aca
palabra2 = input()
numero2 = int(palabra2)

resultado = numero1 + numero2
print("La suma de los 2 numeros es ", resultado)
```

¿Problemos ahora?

Dime un numero

4

Dime ahora otro numero

2

La suma de los 2 numeros es 6

...

```
print("Dime un numero")

#preguntamos por una palabra y lo transformamos a numero
palabral = input()
numero1 = int(palabral)

print("Dime ahora otro numero")

#lo mismo aca
palabra2 = input()
numero2 = int(palabra2)

resultado = numero1 + numero2
print("La suma de los 2 numeros es ", resultado)
```



El mismo código en Python y en Scratch.


```
print("Dime un numero")
numero1 = int(input())

print("Dime ahora otro numero")
numero2 = int(input())

resultado = numero1 + numero2
print("La suma de los 2 numeros es ", resultado)
```



Es lo mismo lo que hacíamos en scratch cuando metíamos un bloque dentro de otro.

Es el mismo código pero mas resumido. Es lo mismo.

```
print("Dime un numero")
numero1 = int(input())

print("Dime ahora otro numero")
numero2 = int(input())

print("La suma de los 2 numeros es ", numero1 + numero2)
```

Y es el mismo código, pero aun mas resumido.

```
print("La suma de los 2 numeros es ", numerol + numero2)
print("La resta de los 2 numeros es ", numerol - numero2)
print("La multiplicacion de los 2 numeros es ", numerol * numero2)
print("La division de los 2 numeros es ", numerol / numero2)
```

Y obviamente, no solo puedes sumar. También puedes restar, multiplicar, dividir...

¿BREAK?

**¿ES HORA DE
EJERCICIOS?**

- Escriba un programa que pida al usuario que escriba su nombre y el de su compañero, y que salude a ambos a la vez.
- Escriba un programa que calcule el promedio de 4 notas ingresadas por el usuario.
- Escriba un programa que convierta de centímetros a pulgadas. Una pulgada es igual a 2.54 centímetros.
- Escriba un programa que pregunte por teclado una cantidad en pesos, y imprima esa misma cantidad a dólares. Considere que el dólar está a \$677 pesos.

```
#repaso de lo que vimos hoy

print("Estoy diciendo algo")
variable = input()
print("La variable es ", variable)

print("Ingresa un numero")
numero = int(input())
numero = numero + 2
print("El numero mas 2 es ", numero)
```