

Colección de actividades Aprender Conectados
Nivel Secundario

Programación

Pixelart - del papel a la pantalla



```
5 {  
6   aprender.a.programar;  
7   si situacion.problematika = verdadero  
8     repetir hasta que problema.resuelto = verdadero  
9     pienso.estrategia  
10    diseño.algoritmo  
11    busco.error (errores)  
12    si errores <> 0  
13      decir "Corrigiendo errores..."  
14      encuentro. error  
15      corrijo.error  
16    fin si  
17  fin repetir  
fin si  
}
```

Actividad N° 1

Autoridades

Presidente de la Nación

Mauricio Macri

Jefe de Gabinete de Ministros

Marcos Peña

Ministro de Educación, Cultura, Ciencia y Tecnología

Alejandro Finocchiaro

Secretario de Gobierno de Cultura

Pablo Avelluto

Secretario de Gobierno de Ciencia, Tecnología e Innovación Productiva

Lino Barañao

Titular de la Unidad de Coordinación General del Ministerio de Educación, Cultura, Ciencia y Tecnología

Manuel Vidal

Secretaria de Innovación y Calidad Educativa

Mercedes Miguel

Subsecretario de Coordinación Administrativa

Javier Mezzamico

Directora Nacional de Innovación Educativa

María Florencia Ripani

ISBN en trámite

Este contenido fue producido por el Ministerio de Educación, Cultura, Ciencia y Tecnología de la Nación en el marco del Plan Aprender Conectados



Introducción

El Plan Aprender Conectados es la primera iniciativa en la historia de la política educativa nacional que se propone implementar un programa integral de alfabetización digital, con una clara definición sobre los contenidos indispensables para toda la Argentina.

En el marco de esta política pública, el Consejo Federal de Educación aprobó, en 2018, los Núcleos de Aprendizajes Prioritarios (NAP) de Educación Digital, Programación y Robótica (EDPR) para toda la educación obligatoria, es decir, desde la sala de 4 años hasta el fin de la secundaria. Abarcan un campo de saberes interconectados y articulados, orientados a promover el desarrollo de competencias y capacidades necesarias para que los estudiantes puedan integrarse plenamente en la cultura digital, tanto en la socialización, en la continuidad de los estudios y el ejercicio de la ciudadanía, como en el mundo del trabajo.

La incorporación de Aprender Conectados en la Educación Secundaria permite poner a disposición de estudiantes y docentes, tecnología y contenidos digitales que generan nuevas oportunidades para reconocer y construir la realidad: abre una ventana al mundo, facilita la comunicación y la iniciación a la producción digital.

La sociedad está cambiando a un ritmo más acelerado que nuestro sistema educativo y la brecha entre las propuestas pedagógicas que presentan las escuelas y la vida de los estudiantes se amplía cada vez más. Garantizar el derecho a aprender en el siglo XXI implica que todos los estudiantes puedan desarrollar las capacidades necesarias para actuar, desenvolverse y participar como ciudadanos en esta sociedad cada vez más compleja, con plena autonomía y libertad.

En este marco, Aprender Conectados presenta actividades, proyectos y una amplia variedad de recursos educativos para orientar la alfabetización digital en la educación obligatoria en todo el país. La actividad que se presenta a continuación y el resto de los recursos del Plan son un punto de partida sobre el cual cada docente podrá construir propuestas y desafíos que inviten a los estudiantes a disfrutar y construir la aventura del aprender.

María Florencia Ripani
Directora Nacional de Innovación Educativa

Objetivos generales

Núcleos de Aprendizajes Prioritarios

Educación Digital, Programación y Robótica – Educación secundaria

Ofrecer situaciones de aprendizaje que promuevan en los alumnas y alumnos:

- El desarrollo de proyectos creativos que involucren la selección y la utilización de múltiples aplicaciones, en una variedad de dispositivos, para alcanzar desafíos propuestos, que incluyan la recopilación y el análisis de información.
- La resolución de problemas a partir de su descomposición en partes pequeñas, aplicando diferentes estrategias, utilizando entornos de programación tanto textuales como icónicos, con distintos propósitos, incluyendo el control, la automatización y la simulación de sistemas físicos.

Objetivos de aprendizaje

Esta actividad permitirá introducir al lenguaje de programación Python y está orientada a desarrollar conocimientos iniciales vinculados con los siguientes objetivos de aprendizaje:

- Conocer el lenguaje de programación Python.
- Familiarizarse con IDLE, su entorno gráfico oficial.
- Conocer buenas prácticas para programar de forma textual, para producir código claro y bien documentado.
- Realizar un proyecto que imprima en pantalla una imagen pixel art previamente diseñada en papel por los alumnos.

Materiales y recursos

- Computadora.
- Python 2.x instalado.
- Hoja cuadriculada.
- Lápiz negro.



Desafío

Los estudiantes deberán realizar un dibujo estilo pixel art utilizando lápiz y papel para luego imprimirlo en pantalla. Trabajarán con el lenguaje Python y su entorno de programación gráfico. El desafío está orientado a que los alumnos se familiaricen con el lenguaje y su entorno, mientras utilizan la tecnología de forma creativa y como un nexo entre el mundo físico y lo virtual.

¿Qué es Python?

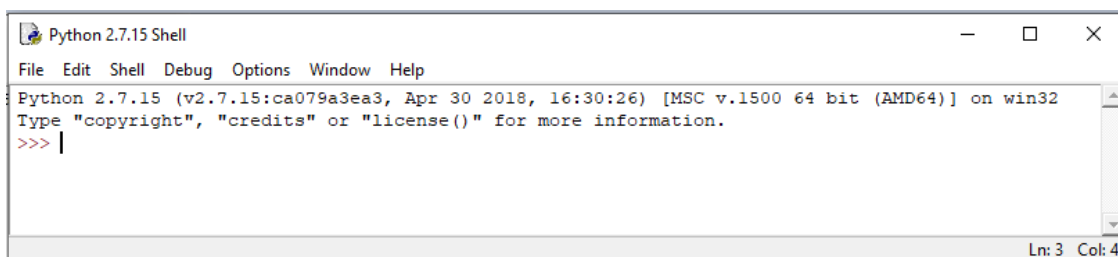
Python es un lenguaje de programación textual libre, fácil de aprender y muy poderoso, creado por la Python Software Foundation. Es un lenguaje ideal para dar los primeros pasos de programación con código escrito.

A continuación se listan algunas de sus ventajas:

- Sencillo de utilizar: los alumnos podrán hacer programas sencillos en poco tiempo.
- Código claro: la forma en que se escribe el código lleva a una lectura clara del mismo.
- Multiplataforma: Python puede correr en toda clase de dispositivos, desde grandes servidores hasta micro controladoras.
- Variedad de bibliotecas: existe una gran variedad de bibliotecas disponibles que amplían la funcionalidad del recurso.
- Comunidad de usuarios: existe una gran comunidad que aporta código disponible para ser descargado y utilizado de forma libre, además de foros activos que proporcionan ayuda en forma inmediata.

Reconocimiento del entorno de trabajo de Python

El entorno de trabajo oficial de Python se llama IDLE. Para ejecutarlo debes buscarlo entre los programas del menú inicio de tu sistema operativo. Recomendamos que el docente realice el reconocimiento inicial de la interfaz, junto a sus alumnos.



Al ejecutar Python IDLE, se crea una ventana como esta en el escritorio de tu sistema operativo.

La interfaz cuenta con una barra de menús al igual que en muchos otros entornos de trabajo visual. Desde el menú "File" podemos crear un nuevo archivo, abrir un archivo existente y salvar nuestra sesión actual, entre otras opciones.



Ventana ejemplo.

Buenas prácticas

Al momento de escribir un código en forma textual -como es el caso de Python- se deben tener en cuenta varios aspectos vinculados con las buenas prácticas, para que sea legible y operativo. A continuación enumeramos algunas, que comprenderán y aplicarán a lo largo de las diversas actividades que se desarrollen.

- Escribir los programas en forma simple y clara.
- Utilizar comentarios para explicar el propósito, funcionamiento completo y el resultado esperado en las líneas más importantes de código.
- Definir los nombres de funciones, variables y constantes en forma representativa del contenido, dato y función que realizarán o almacenarán.
- Declarar variables en líneas separadas, posibilitando agregar una descripción de cada variable mediante comentarios.
- Evitar la incorporación de más de una instrucción por línea.
- Establecer valores iniciales para las variables y contadores.
- Respetar los modos de sintaxis de las instrucciones (cualquier detalle que no se ajusta al formato será marcado como error al momento de ejecutar el programa).

Los programas suelen estar organizados en las siguientes secciones:

- Encabezado, con datos de su creador y breve descripción del programa.
- Definición de librerías utilizadas.
- Declaración de constantes.
- Declaración de variables globales.
- Declaración de funciones.
- Declaración de bloque de inicialización (si existiera).
- Declaración de bloque principal.
- Cierre del programa, al salir del bloque principal (si existiera).

< Inicio >

Disparador

Se presenta en clase el concepto de pixel art como forma de arte digital. ¿Lo conocen? ¿Conocen su origen? ¿Se animan a crear el propio?

A continuación se invita a los alumnos a crear su propio pixel art diseñado en papel para luego imprimirlo en pantalla.

En este documento las líneas de código son presentadas con el siguiente formato, para su fácil identificación y copiado:

```
# Soy un comentario en el código  
print 'Soy una línea de código'  
Soy una línea de código
```

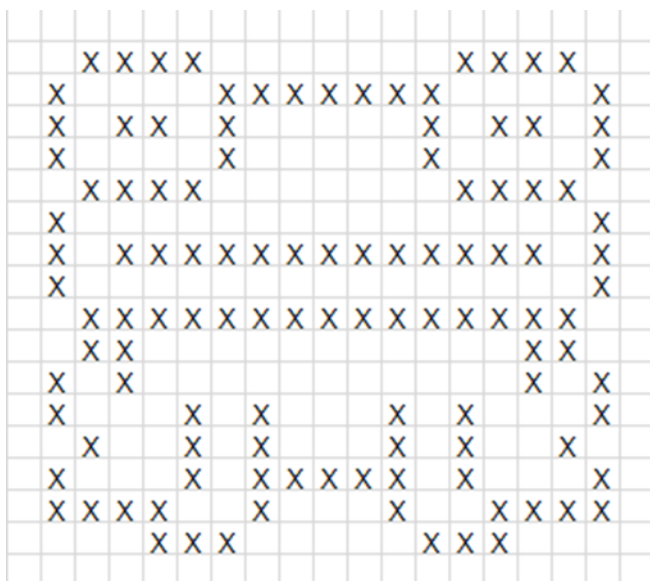
Las líneas de color azul son la respuesta al código introducido en las líneas anteriores y se presentan como un ejemplo del resultado a obtener. No deben ser copiadas y ni ejecutadas en IDLE.

También se incluyen comentarios en gris, precedidos por el símbolo numeral. Estos comentarios son notas que dan claridad al código, pero que Python ignora y no son ejecutados.

< Desarrollo >

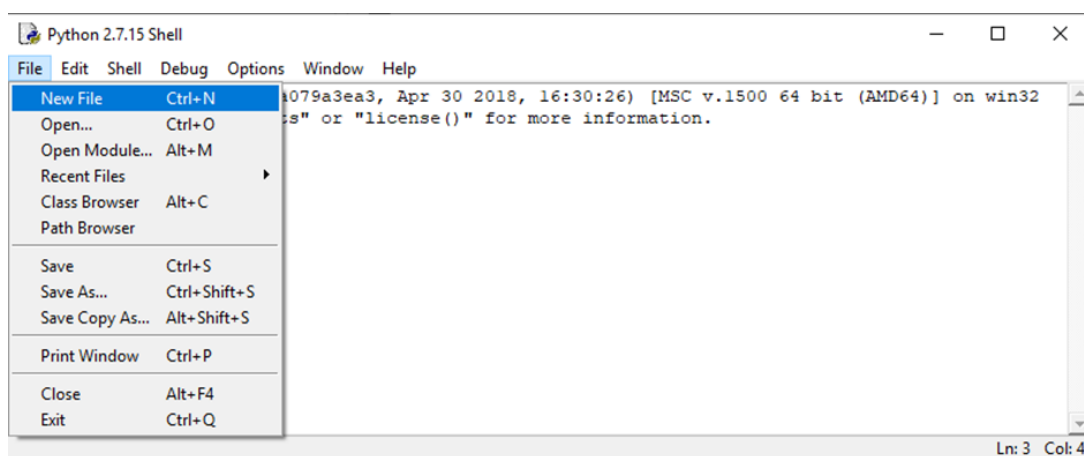
Para comenzar la actividad, el docente les pide a los alumnos que diseñen su propio pixel art, utilizando una hoja cuadrículada y un lápiz negro. Los alumnos deben marcar con una cruz las casillas que desean pintar y dejar vacías las que no.

A continuación se presenta el modelo que es utilizado como ejemplo para esta actividad.

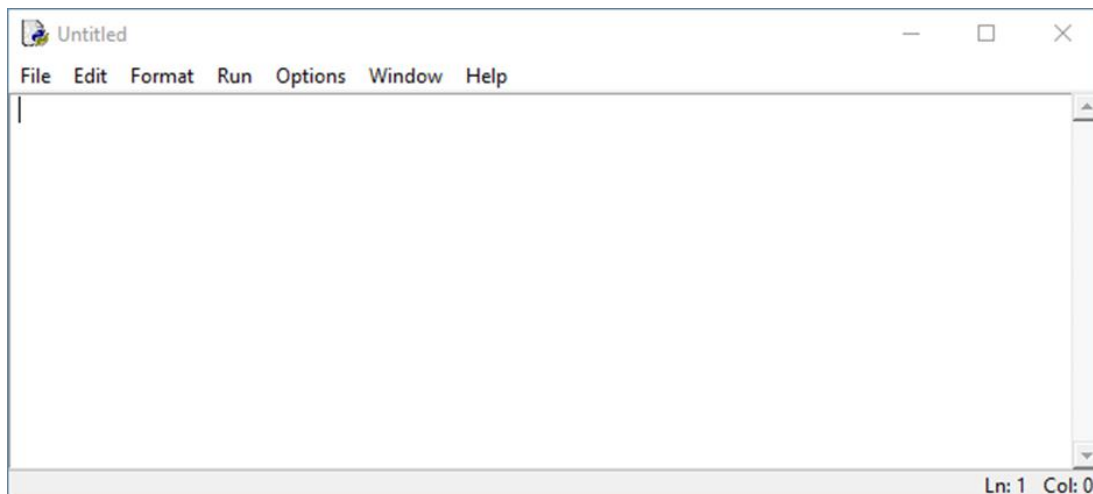


Ejemplo de pixel art en una hoja cuadriculada.

Se sugiere que luego de dibujar el pixel art, el docente guíe a los alumnos para identificar cómo acceder a IDLE Python en sus computadoras y abrir un nuevo archivo (New file).



Crear un nuevo archivo de trabajo en IDLE



Ventana ejemplo de un nuevo archivo de trabajo en IDLE

Pasos de la actividad

1. Incorporar un comentario que describa el nombre del creador y una descripción acerca del programa. Para incorporar un comentario se utiliza el símbolo numeral # y luego se escribe el texto.

```
# Creador Pedro Perez  
# Este programa imprime en pantalla un dibujo pixel art
```

Puedes incluir otra información en el encabezado, como la fecha de creación o el lenguaje de programación en el que está escrito.

2. Tomar el patrón que se realizó en papel cuadriculado al comienzo de la actividad y que será utilizado como guía para programar el dibujo. Con la ayuda de Python, se escribirá una línea de código por cada fila horizontal que abarque esta ilustración, de modo de construirla digitalmente. Para esto utilizamos la función `print()` de python, que es una función que imprime -es decir, escribe en la pantalla, cuando se ejecuta el programa- el contenido que se encuentra entre los paréntesis. Puede ser una cadena de caracteres, una variable o el resultado de una función.

Para imprimir nuestro pixel art, lo imprimimos fila por fila y utilizamos espacios para indicar las casillas vacías, y el símbolo numeral (#) para indicar las casillas coloreadas.

Para imprimir la primera fila de nuestro pixel art que utilizamos como ejemplo, debemos imprimir 2 espacios en blanco, seguidos de 4 símbolos numeral, seguidos de 7 espacios en blanco para terminar la línea con otros 4 símbolos numeral. No es necesario imprimir los espacios en blanco al final de cada línea.

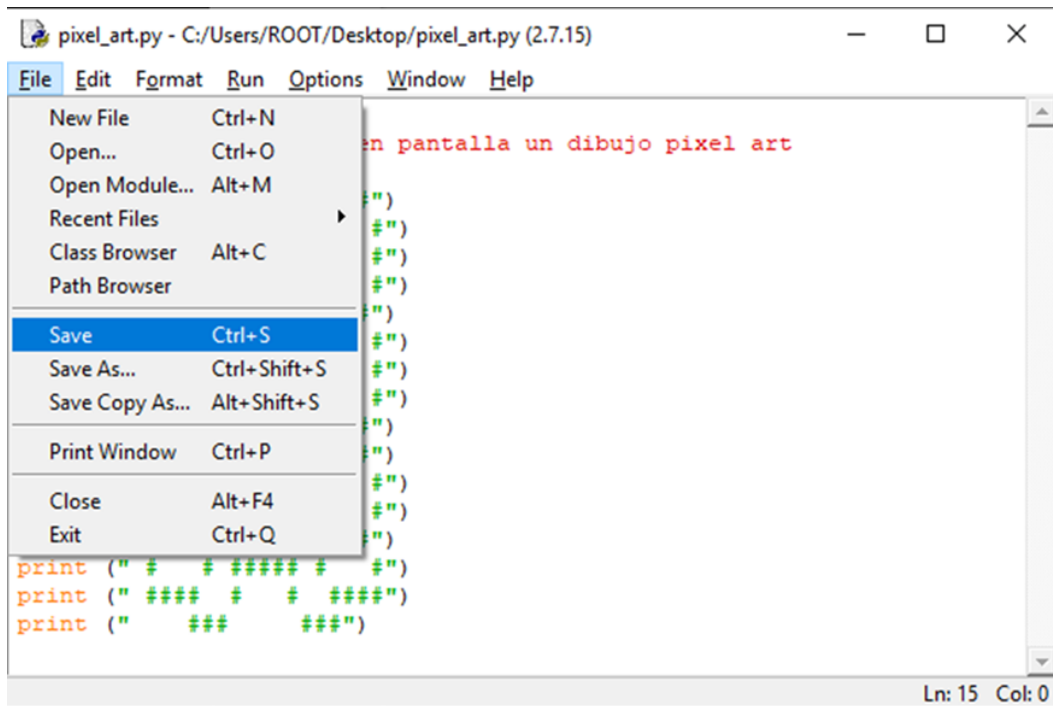
```
print ("  ####      ####")
```

Seguir esta misma regla y escribir el código para imprimir el resto de las filas de nuestro pixel art. A continuación se encuentra el código que dibuja en forma completa el pixel art de nuestro ejemplo:

```
# Creador Pedro Perez
# Este programa imprime en pantalla un dibujo pixel art

print ("  ####      ####")
print (" #      #####  #")
print (" #  #  #      #  #  #")
print (" #      #      #      #")
print ("  ####      ####")
print (" #                                     #")
print (" # #####          #")
print (" #                                     #")
print (" #####")
print ("  ##                                ##")
print (" #  #                                #  #")
print (" #      #  #      #  #      #")
print (" #      #  #      #  #      #")
print (" #      # #####  #      #")
print (" ####  #      #  ####")
print ("      ##      ##")
```

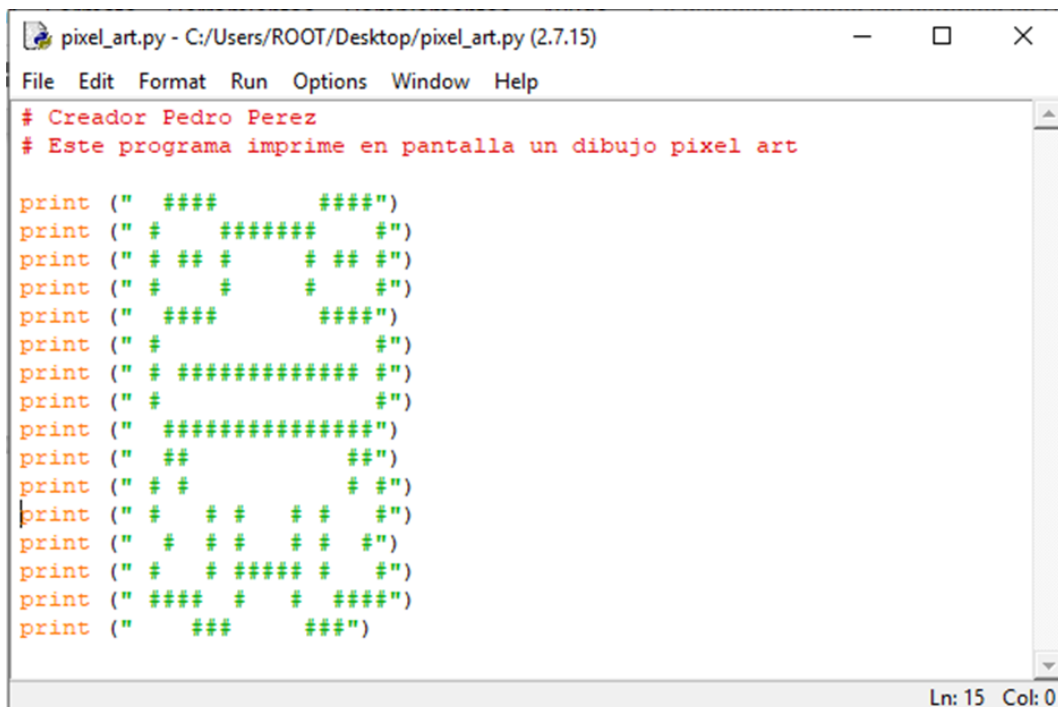
3 - Guardar el proyecto para ejecutarlo. Acceder desde File, opción Save y colocar un nombre específico y reconocible, como "pixel art".



Ventana de ejemplo

5 - Python funciona como un intérprete que traduce las instrucciones en un lenguaje entendible para la computadora, quien es la encargada de ejecutar las acciones del programa.

Para que la computadora ejecute el programa, se debe ir al menú Run y seleccionar la opción Run Module. Puedes obtener el mismo resultado al presionar la tecla F5.



Ventana de ejemplo

El programa ejecutado mostrará el pixel art creado, como se muestra abajo.

A screenshot of a Windows command prompt window titled "Python 2.7.15 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area shows the following output:

```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:/Users/ROOT/Desktop/pixel_art.py =====  
#####  
#           #  
#   #####   #  
#     #       #  
#         #   #  
#####  
#  
#   #####   #  
#     #       #  
#         #   #  
#####  
#  
#   #####   #  
#     #       #  
#         #   #  
#####  
#  
#   #####   #  
#     #       #  
#         #   #  
#####  
#  
#   #####   #  
#     #       #  
#         #   #  
#####
```


The output consists of several lines of hash symbols (#) arranged to form a pixelated character, which appears to be a stylized letter 'P' or a similar shape. The character is composed of multiple rows and columns of hashes, with some internal spaces creating its unique silhouette. The window's status bar at the bottom right indicates "Ln: 21 Col: 1".

Ventana de ejemplo

< Cierre >

Para finalizar la actividad, los estudiantes se agruparán en parejas y cada uno compartirá su producción con sus compañeros, mostrando el dibujo que utilizó como patrón para crearla.

Los integrantes de cada grupo harán un intercambio para evaluar juntos si se necesita hacer algún cambio para mejorar la producción. El trabajo en equipo sirve como base para desarrollar en forma creciente la autonomía de los estudiantes en la utilización de este recurso de programación.

Si se cuenta con proyector o una pantalla grande, se puede cerrar la actividad con una muestra de todas las producciones. De lo contrario, se pueden hacer rotar a los estudiantes para ver todos los trabajos de sus compañeros.



Evaluación

El docente puede evaluar el proyecto tanto a través de la observación, durante el desarrollo de las actividades, como en relación al programa final.

Tomará en cuenta la actividad de evaluación entre pares realizada, donde los estudiantes compararon la idea original de sus compañeros (dibujo patrón en papel) y el dibujo creado utilizando Python.

A continuación, se presentan preguntas orientadoras:

- a) ¿Logró realizar el desafío propuesto?
- b) Si el desafío le presentó dificultades, ¿logró superarlas a través del intercambio en equipo?
- c) ¿Construyó modelos alternativos al ejemplo de la actividad?

El proceso de evaluación de la evolución del aprendizaje podrá continuar con la actividad propuesta a continuación.

Para seguir aprendiendo

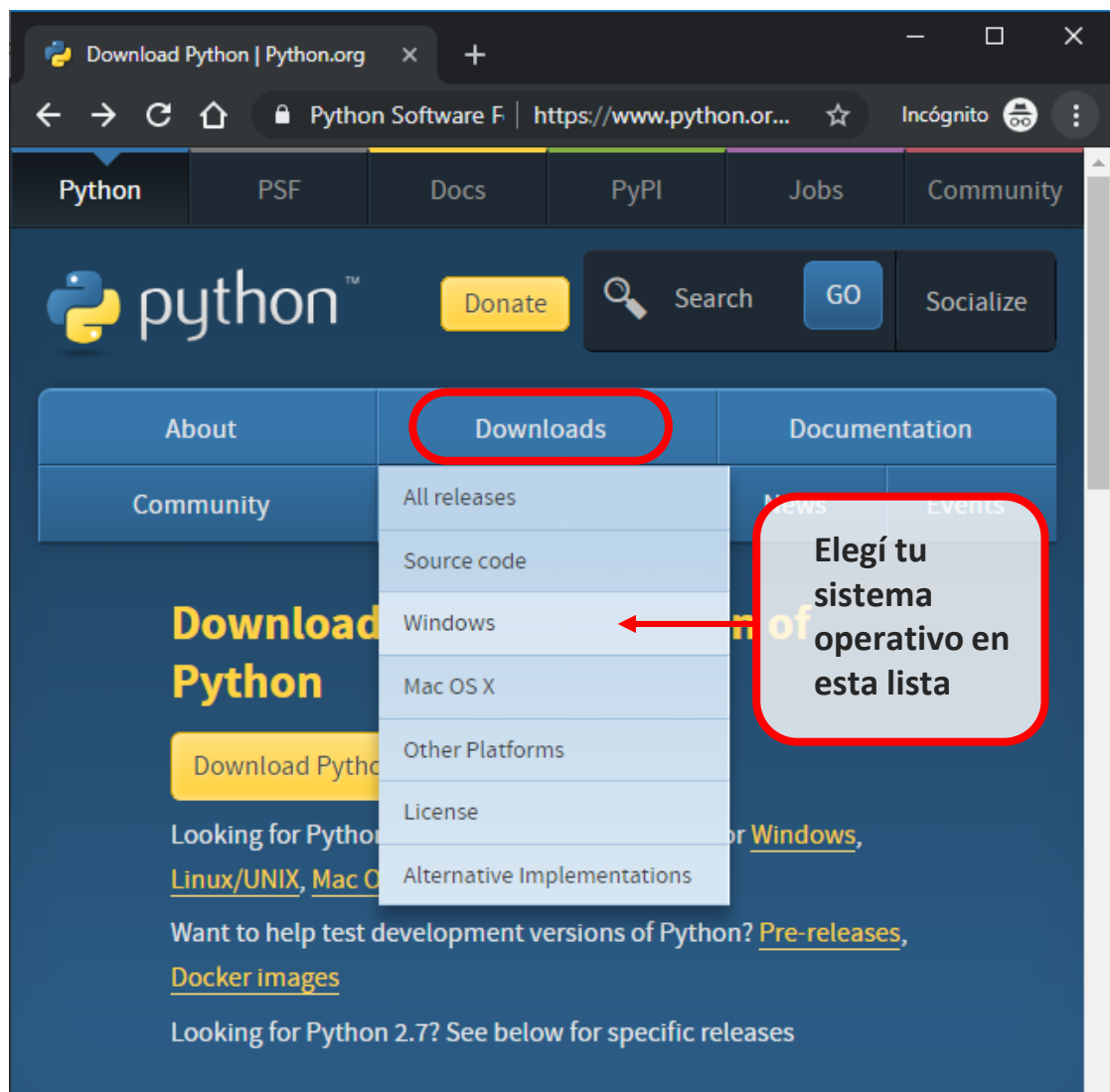
Cada uno de los estudiantes entrega su dibujo pixel art y el programa que lo representa, a un compañero (cediendo la computadora con la que venían trabajando o pasando el archivo del programa).

Cada estudiante tendrá un proyecto que hizo otro estudiante y se propondrá intervenirlo. Pueden agregar lo que quieran al dibujo o modificarlo, lo cual deberán luego reflejar en el programa. De este modo, crearán un producto tecnológico digital, interviniendo uno preexistente.

Anexo

¿Cómo instalar Python?

Antes de instalar Python, es necesario descargar la última versión disponible de Python desde su sitio web oficial, <http://www.python.org>. Una vez en el sitio, seleccionar la opción “Downloads”, y elegir el sistema operativo entre las opciones de la lista desplegable:



En la siguiente pantalla, elegir la última versión de Python 2 para acceder a la página de descargas:

Download Python | Python.org

Python Software Foundation | [https://www.python.org/...](https://www.python.org/)

Python PSF Docs PyPI Jobs Community

python™ Donate Search GO Socialize

About Downloads Documentation
Community Success Stories News Events

Download the latest version of Python

Download Python 3.7.3

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more
Python 3.7.3	March 25, 2019	Download Release Notes
Python 3.4.10	March 18, 2019	Download Release Notes
Python 3.5.7	March 18, 2019	Download Release Notes
Python 2.7.16	March 4, 2019	Download Release Notes
Python 3.7.2	Dec. 24, 2018	Download Release Notes
Python 3.6.8	Dec. 24, 2018	Download Release Notes

Elegí la última versión de Python 2

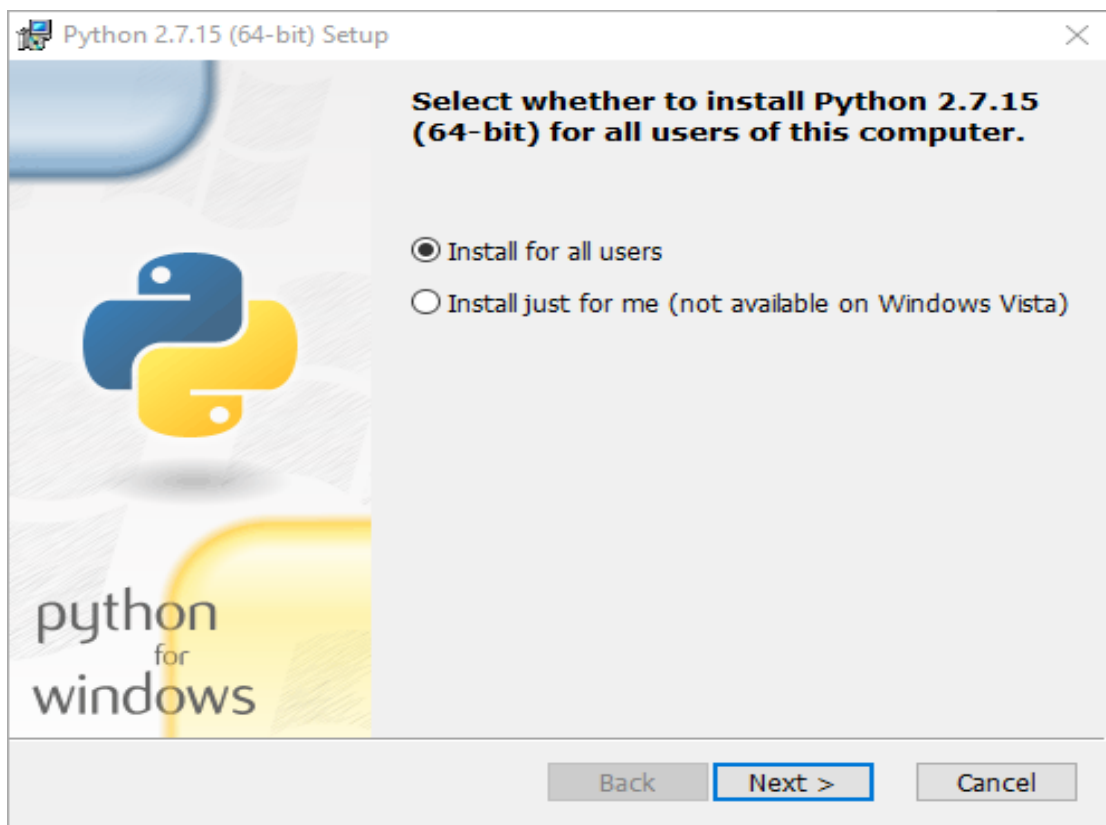
Desplazar el cursor hacia abajo en la página de descarga y elegir el archivo que corresponde al instalador para el sistema operativo de la máquina que se va a utilizar. El clic dará inicio a la descarga:

The screenshot shows the Python 2.7.16 release page. A table lists download links for various operating systems. Two red callouts with numbered circles provide instructions:

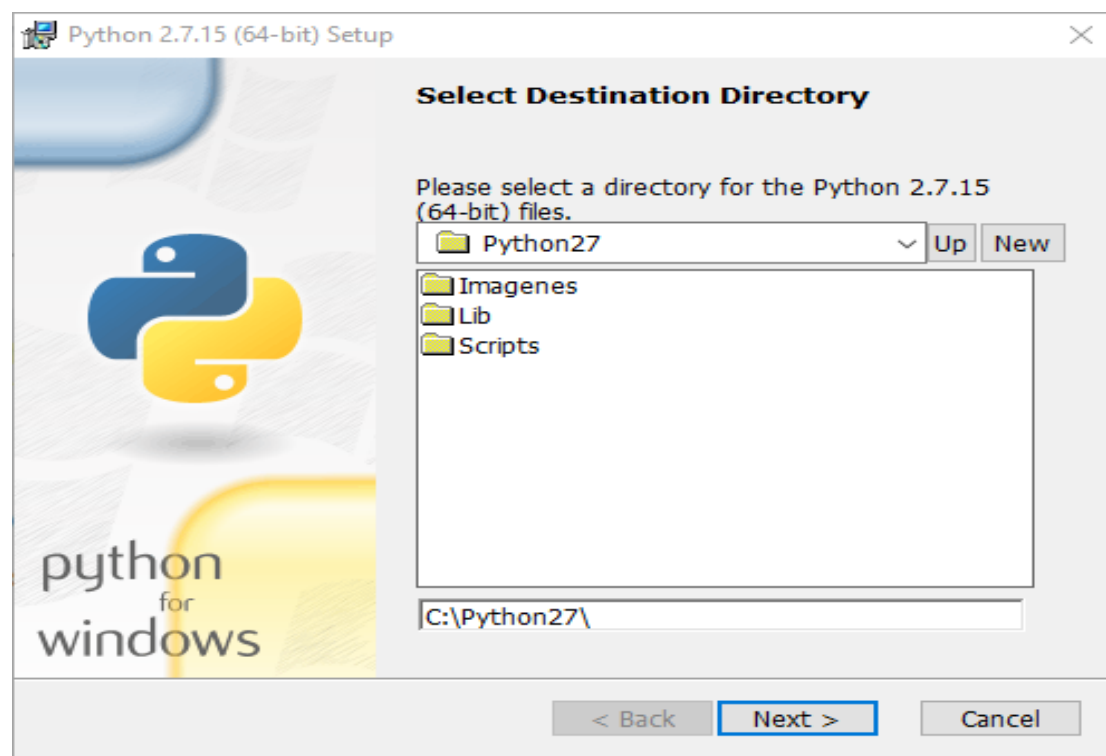
- 1** Hacé clic en el instalador. Esto iniciará la descarga. (Click on the installer. This will start the download.) - Points to the 'Windows x86-64 MSI installer' link.
- 2** Una vez descargado podrás instalarlo. (Once downloaded you will be able to install it.) - Points to the downloaded file 'python-2.7.16.amd64.msi' in the taskbar.

Platform	Architecture	Notes	SHA256 Hash	Size
tarball				
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	70b0f58eba7b78b174056369b076c085	3025
Windows debug information files	Windows		f94690edbbf58b10bfd718badc08b1f8	2508
Windows debug information files for 64-bit binaries	Windows		4292c4db30c27fedbbec8544967b6452	2589
Windows help file	Windows		3bbf29b6712b231d2dff9211fc7b21e2	6263
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64	2fe83194bb4027be75b29852027f1a79	2036
Windows x86 MSI installer	Windows		91228345b7e0428544ec4edcdf70286	1941

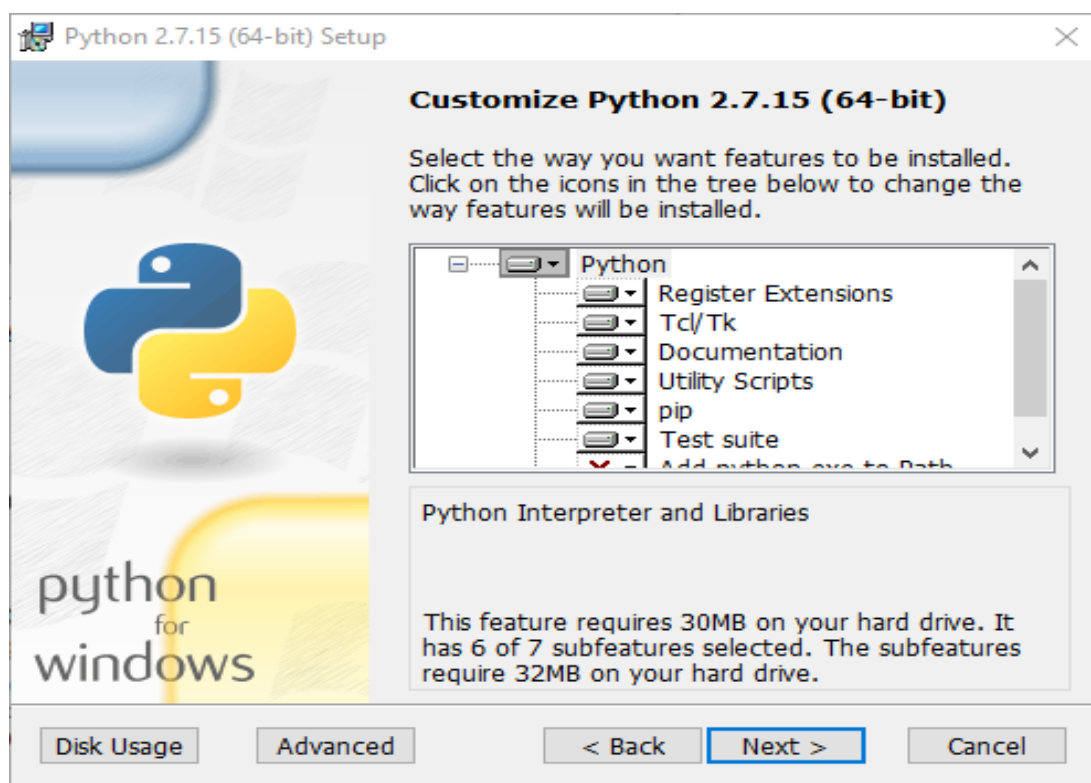
Ejecutar el instalador descargado. Se puede elegir algunas opciones durante la instalación, o simplemente aceptar la configuración por defecto:



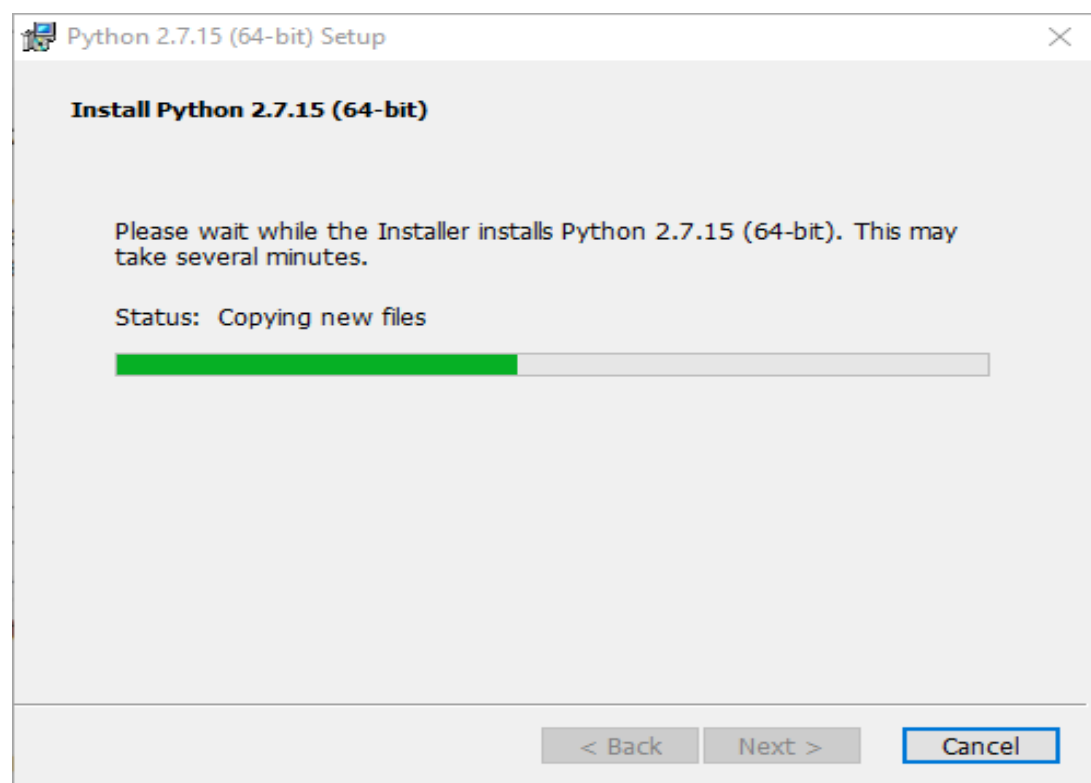
Paso 1. Se puede instalar para todos los usuarios de la computadora, o solo para el propio.



Paso 2. Elegir dónde se instalará Python.



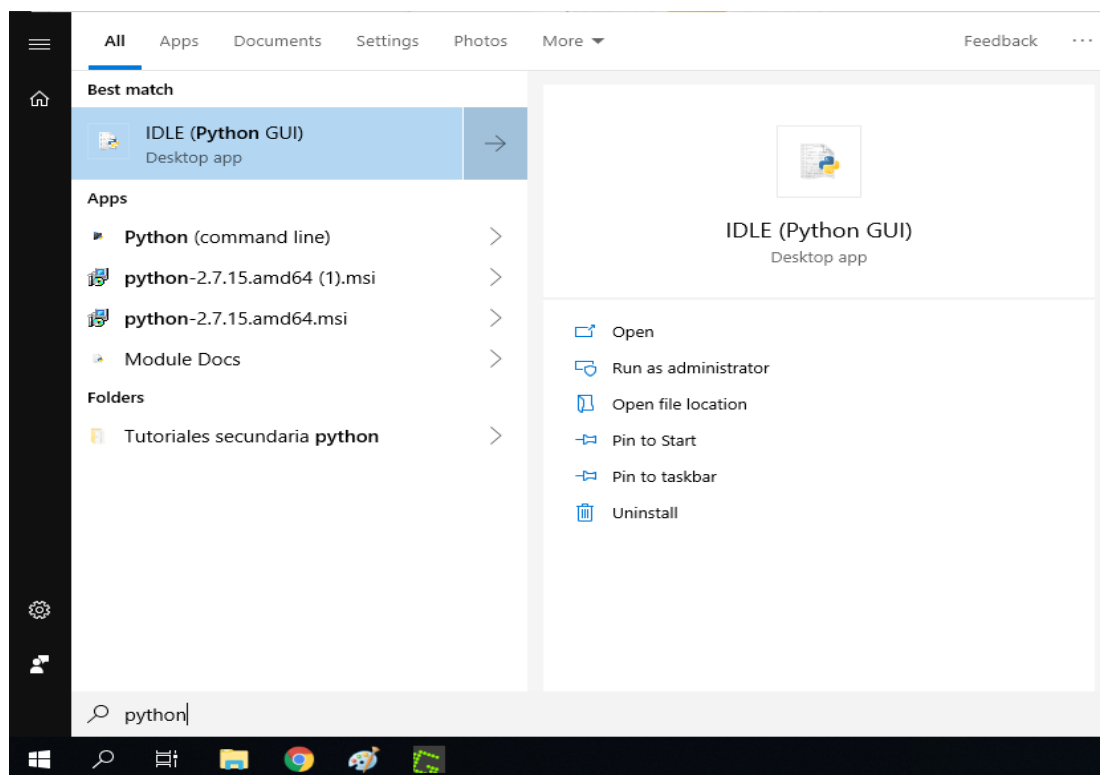
Paso 3. Elegir qué componentes de Python se desea instalar. Si no estás seguro, aceptar la configuración por defecto y dar clic a “Next >”.



Paso 4. Esperar a que Python sea instalado en la computadora.



Paso 5. Python ya se encuentra instalado en tu computadora. Cerrar el instalador.



Ejecutar IDLE (Python GUI) desde el menú "Inicio" de tu sistema operativo.