

## Taller de electrónica y multimedia

### Sesión 6

#### Instalación ambientes Python

```
sudo apt-get install git python-pip make build-essential libssl-dev zlib1g-dev  
libbz2-dev libreadline-dev libsqlite3-dev  
sudo pip install virtualenvwrapper
```

```
git clone https://github.com/yyuu/pyenv.git ~/.pyenv  
git clone https://github.com/yyuu/pyenv-virtualenvwrapper.git  
~/.pyenv/plugins/pyenv-virtualenvwrapper
```

```
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc  
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc  
echo 'eval "$(pyenv init -)"' >> ~/.bashrc  
echo 'pyenv virtualenvwrapper' >> ~/.bashrc
```

#### Reinicia la consola, los cambios ser verán reflejados al reiniciar consola

pyenv install --list // traerá las versiones de Python y gestores de paquetes como conda , anaconda y miniconda, útiles para computer visión

```
pyenv install 3.6.2 // de acuerdo a la versión que se necesite  
pyenv install 2.7
```

#### Instalaremos un ambiente virtual con Python 3.6.2

```
git clone https://github.com/yyuu/pyenv-virtualenv.git ~/.pyenv/plugins/pyenv-  
virtualenv  
source ~/.bashrc
```

```
mkdir ambientes_python
```

```
cd ambientes_python
```

```
pyenv virtualenv 3.6.2 mi_ambiente36 //instalo una version de Python 3.6.2
```

#### Revisamos la creación del ambiente

```
pyenv versions
```

```
pyenv activate mi_ambiente36 //activamos el ambiente, debería ver algo como
```

```
(ambiente36)pi@raspberrypi:~/ambientes_python
```

```
pip install --upgrade pip
```

**Desde acá será fácil instalar mediante comando pip, ¡allá vamos!**

```
source deactivate
```

## Utilización librería pygame

Pygame viene instalada en RPI3, pero puedes instalarlas para cualquier sistema utilizando el comando, como hemos creado un ambiente toca instalarla.

```
sudo apt-get build-dep python-pygame
sudo apt-get install python-dev

sudo apt-get install mercurial
```

```
sudo -H pip3 install gitsome
```

```
sudo apt-get install python3-dev python3-numpy libSDL-dev libSDL-image1.2-
dev libSDL-mixer1.2-dev libSDL-ttf2.0-dev libsmpeg-dev libportmidi-
dev libavformat-dev libswscale-dev libjpeg-dev libfreetype6-dev
```

**No importa que diga que hay paquetes que ya están instalados, no corremos riesgos**

```
sudo apt-get install libSDL-dev libSDL-image1.2-dev libSDL-mixer1.2-dev libSDL-
ttf2.0-dev
sudo apt-get install libsmpeg-dev libportmidi-dev libavformat-dev libswscale-dev
sudo apt-get install python3-dev python3-numpy
```

```
sudo apt-get build-dep python-pygame

sudo apt-get install libv4l-dev
```

**Activas tu ambiente, crea una carpeta para ese ambiente**

```
mkdir amb36

cd amb36

hg clone https://bitbucket.org/pygame/pygame
```

```
cd pygame
python3 setup.py build
sudo python3 setup.py install
```

**sale de la carpeta**

```
C ../
```

```
pip install pygame
```

ejemplo.py //los archivos no deben llamarse como la librería, entra en conflicto

```
import pygame #se importa pygame
```

```
pygame.init()# se inicializa el modulo pygame
```

```
blanco_fondo = [255, 255, 255]
```

```
ventana = pygame.display.set_mode((400, 300)) # se setea tamaño
```

```
pygame.display.set_caption("Mi primer ejercicio")# setea título de la ventana
```

```
ventana.fill(blanco_fondo)# color de fondo
```

```
ver_ventana = False
```

```
while not ver_ventana:
```

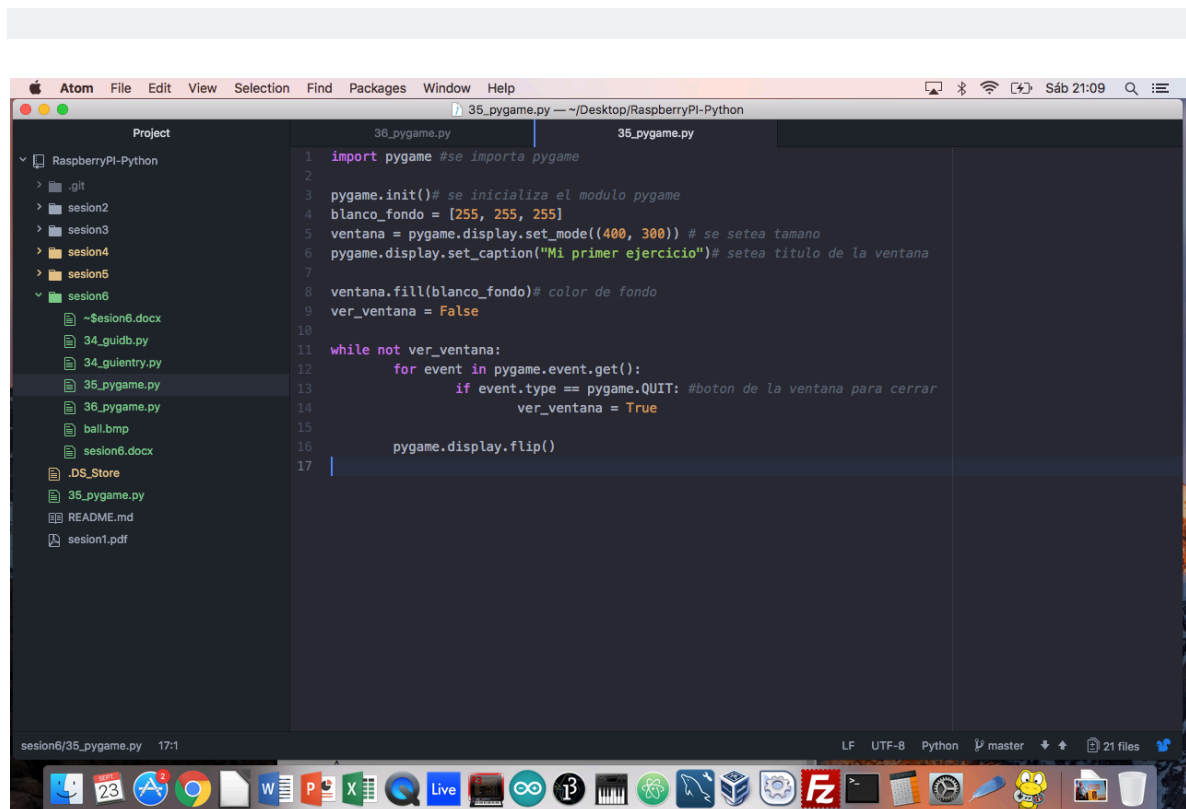
```
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT: #boton de la ventana para cerrar
```

```
            ver_ventana = True
```

```
    pygame.draw.rect(ventana, (0, 128, 255),(30, 30, 60, 60))
```

```
    pygame.display.flip()
```



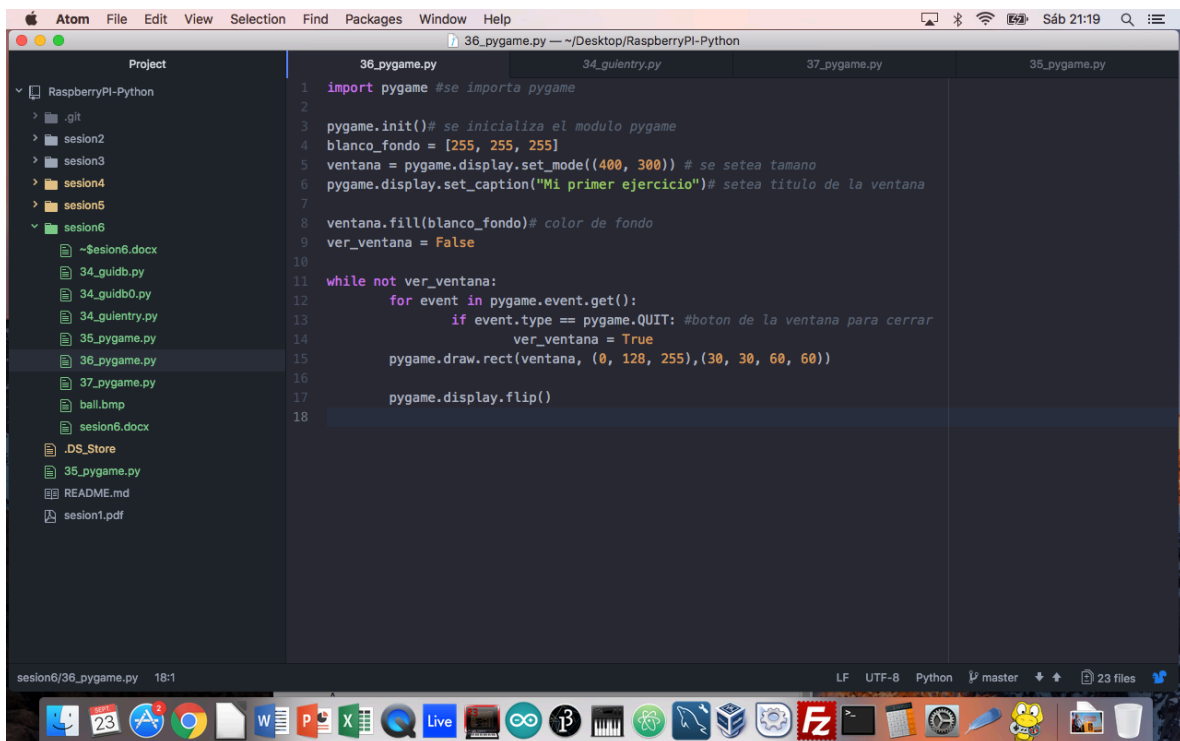
ejecutar

```
python ejemplo.py
```

## Primitivas

<u><code>pygame.draw.rect</code></u>	— draw a rectangle shape
<u><code>pygame.draw.polygon</code></u>	— draw a shape with any number of sides
<u><code>pygame.draw.circle</code></u>	— draw a circle around a point
<u><code>pygame.draw.ellipse</code></u>	— draw a round shape inside a rectangle
<u><code>pygame.draw.arc</code></u>	— draw a partial section of an ellipse
<u><code>pygame.draw.line</code></u>	— draw a straight line segment
<u><code>pygame.draw.lines</code></u>	— draw multiple contiguous line segments
<u><code>pygame.draw.aaline</code></u>	— draw fine antialiased lines
<u><code>pygame.draw.aalines</code></u>	— draw a connected sequence of antialiased lines

## Ejemplo, dibuja cuadrado azul

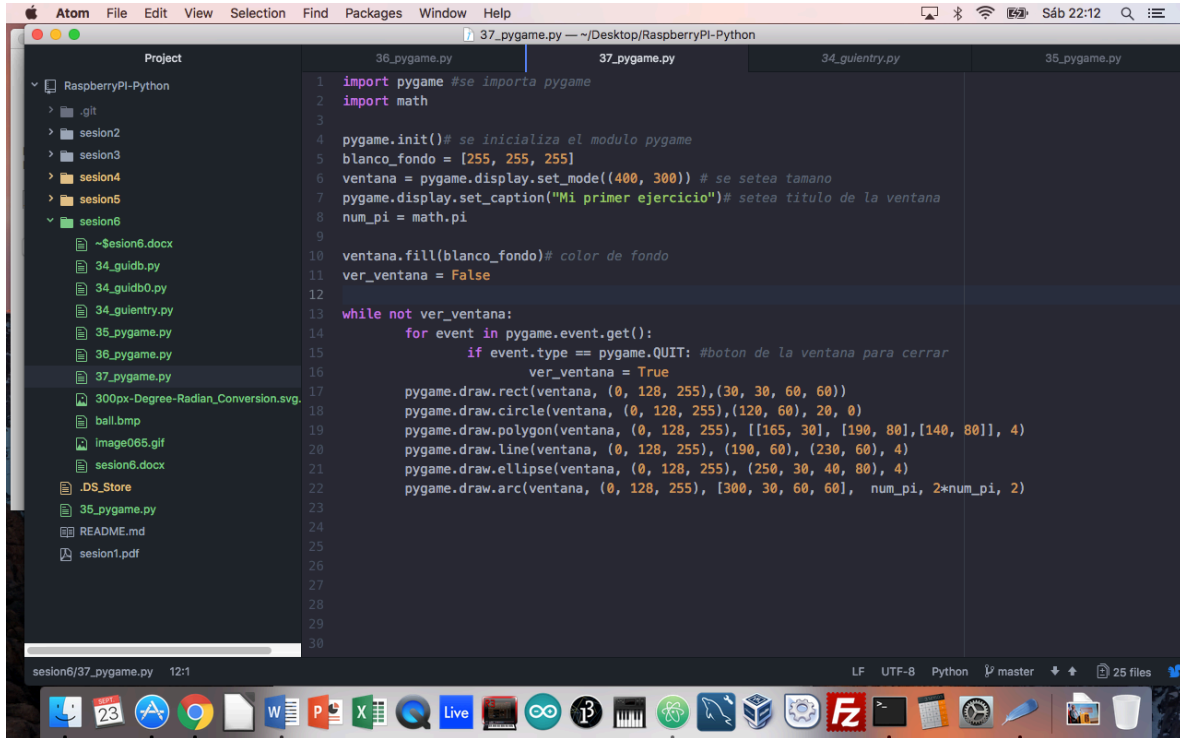


The screenshot shows the Atom code editor with a project named 'RaspberryPi-Python'. The file explorer on the left shows a directory structure with files like '34\_guidb.py', '34\_gulentry.py', '35\_pygame.py', '36\_pygame.py', '37\_pygame.py', 'ball.bmp', 'sesion6.docx', '.DS\_Store', '35\_pygame.py', 'README.md', and 'sesion1.pdf'. The main editor window displays the code for '36\_pygame.py'.

```
1 import pygame #se importa pygame
2
3 pygame.init()# se inicializa el modulo pygame
4 blanco_fondo = [255, 255, 255]
5 ventana = pygame.display.set_mode((400, 300)) # se setea tamaño
6 pygame.display.set_caption("Mi primer ejercicio")# setea título de la ventana
7
8 ventana.fill(blanco_fondo)# color de fondo
9 ver_ventana = False
10
11 while not ver_ventana:
12     for event in pygame.event.get():
13         if event.type == pygame.QUIT: #boton de la ventana para cerrar
14             ver_ventana = True
15     pygame.draw.rect(ventana, (0, 128, 255),(30, 30, 60, 60))
16
17     pygame.display.flip()
18
```

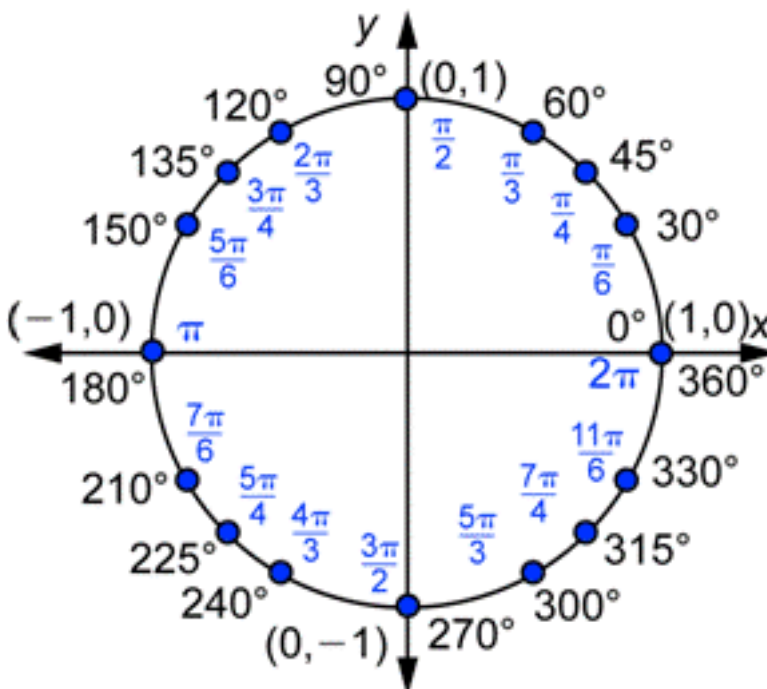
The status bar at the bottom indicates the file is 'sesion6/36\_pygame.py', the cursor is at line 18, column 1, and the encoding is UTF-8. The bottom of the screen shows a macOS dock with various application icons.

## Primitivas ejemplo



```
1 import pygame #se importa pygame
2 import math
3
4 pygame.init()# se inicializa el modulo pygame
5 blanco_fondo = [255, 255, 255]
6 ventana = pygame.display.set_mode((400, 300)) # se setea tamaño
7 pygame.display.set_caption("Mi primer ejercicio")# setea título de la ventana
8 num_pi = math.pi
9
10 ventana.fill(blanco_fondo)# color de fondo
11 ver_ventana = False
12
13 while not ver_ventana:
14     for event in pygame.event.get():
15         if event.type == pygame.QUIT: #boton de la ventana para cerrar
16             ver_ventana = True
17     pygame.draw.rect(ventana, (0, 128, 255), (30, 30, 60, 60))
18     pygame.draw.circle(ventana, (0, 128, 255), (120, 60), 20, 0)
19     pygame.draw.polygon(ventana, (0, 128, 255), [[165, 30], [190, 80], [140, 80]], 4)
20     pygame.draw.line(ventana, (0, 128, 255), (190, 60), (230, 60), 4)
21     pygame.draw.ellipse(ventana, (0, 128, 255), (250, 30, 40, 80), 4)
22     pygame.draw.arc(ventana, (0, 128, 255), [300, 30, 60, 60], num_pi, 2*num_pi, 2)
23
24
25
26
27
28
29
30
```

draw.arc está expresado en Radians.



## Instalar voice recognition

<https://pypi.python.org/pypi/SpeechRecognition/2.1.3>

```
brew install portaudio
```

```
sudo apt-get install python-pyaudio python3-pyaudio //para debían, Ubuntu, rpi
```

```
pip install portaudio
```

```
pip install pyaudio
```

```
pip install SpeechRecognition
```

```
pip install --upgrade google-api-python-client
```