

Introducción a la programación con Python

Proyecto

Alexis Rodríguez

Marcel Morán C

¿Qué es un módulo?

- Instrucciones guardadas en un archivo
- Instrucciones son importadas dentro de otro programa
- La collection de estos módulos se conocen librerías
- **OS y datetime son un ejemplo de estos**
- **Clases y funciones**

from **datetime** import **date**

print("El día de hoy es ", **date.today()**)

>>> El día de hoy es 2022-03-11

```
"""Concrete date/time and related types.

See http://www.iana.org/time-zones/repository/tz-link.html for
time zone and DST data sources.
"""

__all__ = ("date", "datetime", "time", "timedelta", "timezone", "tzinfo",
           "MINYEAR", "MAXYEAR", "UTC")

import time as _time
import math as _math
import sys
from operator import index as _index

def _cmp(x, y):
    return 0 if x == y else 1 if x > y else -1

MINYEAR = 1
MAXYEAR = 9999
_MAXORDINAL = 3652059 # date.max.toordinal()

# Utility functions, adapted from Python's Demo/classes/Dates.py, which
# also assumes the current Gregorian calendar indefinitely extended in
# both directions. Difference: Dates.py calls January 1 of year 0 day
# number 1. The code here calls January 1 of year 1 day number 1. This is
# to match the definition of the "proleptic Gregorian" calendar in
# and Reinhold's "Calendrical Calculations", where it's the base calendar
# for all computations. See the book for algorithms for converting between
# proleptic Gregorian ordinals and many other calendar systems.

# -1 is a placeholder for indexing purposes.
_DAYS_IN_MONTH = [-1, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

_DAYS_BEFORE_MONTH = [-1] # -1 is a placeholder for indexing purposes.
dbm = 0
for dim in _DAYS_IN_MONTH[1:]:
    _DAYS_BEFORE_MONTH.append(dbm)
    dbm += dim
del dbm, dim

def _is_leap(year):
    "year -> 1 if leap year, else 0."
    return year % 4 == 0 and (year % 100 != 0 or year % 400 == 0)

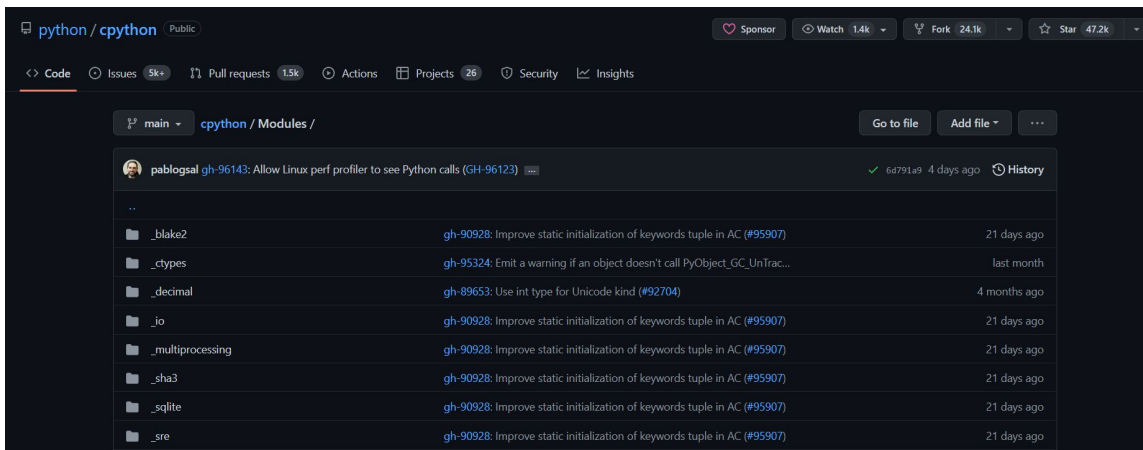
def _days_before_year(year):
    "year -> number of days before January 1st of year."
    y = year - 1
    return y*365 + y//4 - y//100 + y//400

def _days_in_month(year, month):
    "year, month -> number of days in that month in that year."
    assert 1 <= month <= 12, month
    # The following table gives the number of days in each month for
```

¿Qué es una librería?

- Collection de modulus o codigo
- Modulus pueden ser escritos en c o en python

```
import math
```

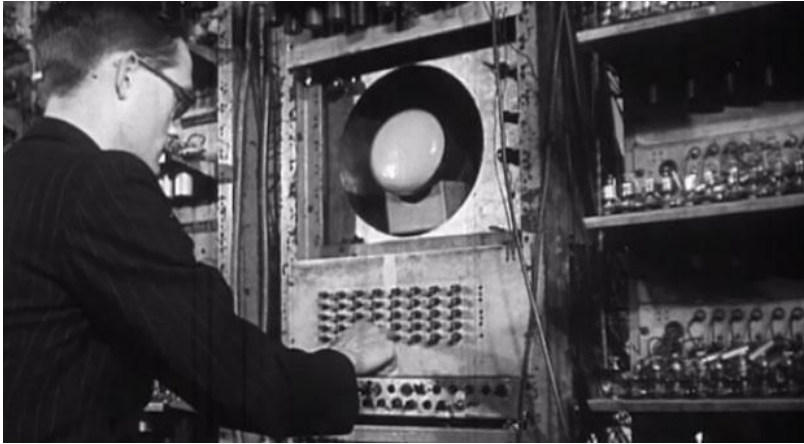


Computadoras

- Baby, la primera computadora, programación realizada a través de switches
- IBM tarjetas perforadas

Computadoras

- Baby, la primera computadora, programación realizada a través de switches
- IBM tarjetas perforadas

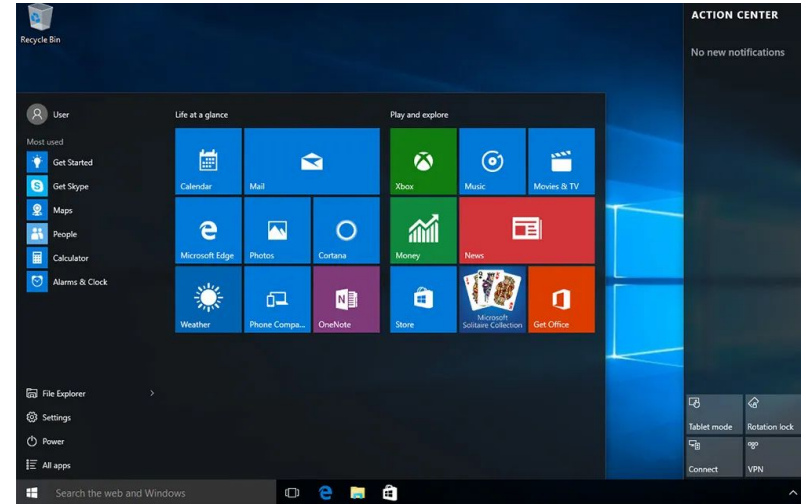


Computadoras

- Usando comandos a través de una terminal
- Mouse, iconos gráficos de aplicaciones

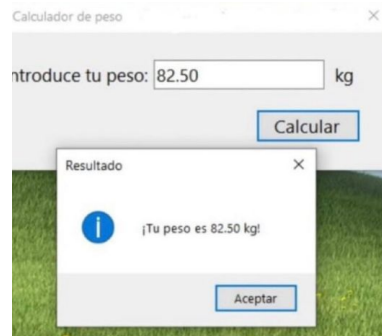


Computadoras



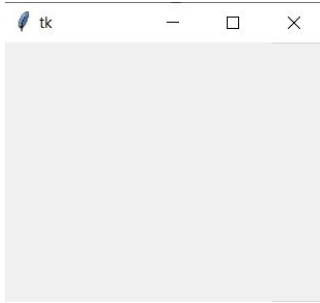
¿Qué es una interfaz gráfica de usuario?

- Permite a un usuario interactuar con dispositivos electrónicos
- Manipulación directa sobre los elementos gráficos
- Facilitan el uso de aplicación (en el mayor de los casos)

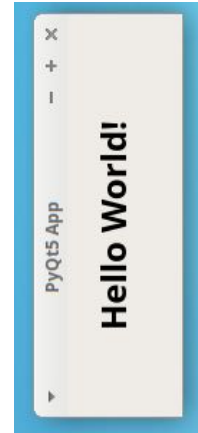
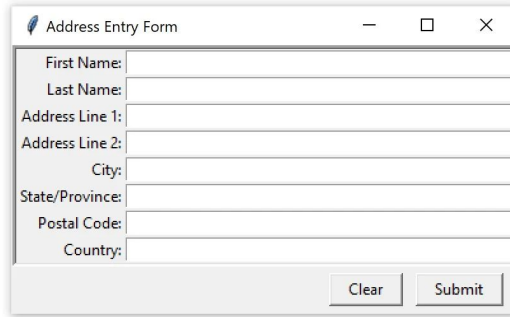
A screenshot of a "Create New Account" form. The form has a title "Create New Account" in bold. It contains several input fields: "First Name" with the value "Patricia R Reed", "Last Name" with the value "Reed", "Email Address" with the value "tskono1542@posiklan.com", and "Password" with the value "Reed". There is also an "Enter OTP" field. To the right of the OTP field is a button labeled "verify email". At the bottom of the form are three buttons: "Clear", "Register", and "Exit".

¿Qué es una interfaz gráfica de usuario?

- Librerías y paquetes para crear interfaces



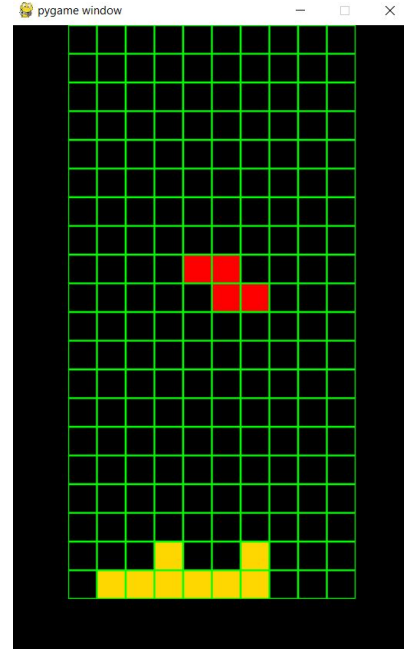
Tkinter



PyQt

PyGame

- Módulos de python, con librerías integradas de gráficos de computadora y sonidos
- Implementación de vectores, colisiones, y sprites
- Cross-platform (inclusive desarrollo en android)



Proyecto

- **Desarrollo de un video juego usando python (Parejas | individual)**
- **Puntos de proyecto 100 puntos, 20 por cada categoría**
 - **Planificación**
 - **Conceptos fundamentales**
 - **POO**
 - **Trabajo en equipo**
 - **Dificultad**

Proyecto

- Desde jueves 25 de Abril al sábado 4 de Mayo
- Desarrollo de un video juego usando python

APRIL 2024						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Proyecto

- Desde jueves 25 de Abril al sábado 4 de Mayo
- Desarrollo de un video juego usando python
- Presentacion (Turno aleatorio)

APRIL 2024						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Proyecto

- Desde jueves 25 de Abril al sábado 4 de Mayo
- Desarrollo de un video juego usando python
- **Presentacion (Turno aleatorio)**
- **Entrega de notas (Lunes 6 de Mayo)**

APRIL 2024						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

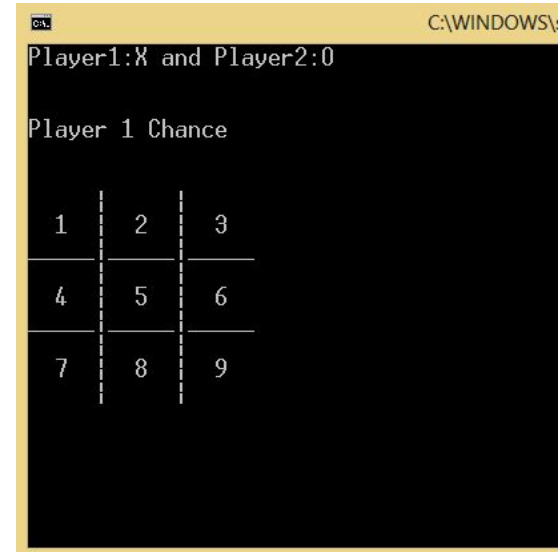
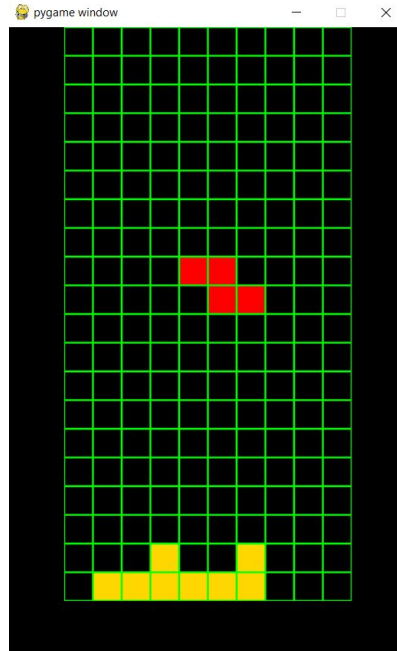
Proyecto

- Que estamos esperando



Proyecto

- Que estamos esperando



Proyecto

- Que estamos esperando



```
C:\Windows\System32\cmd.exe - python3 main.py

D:\primerproyecto\proyecto>python3 main.py
*****
*****Piedra, Papel o Tijera*****
*****
*****

Seleccione una de las tres opciones:
(debe escribir el número correspondiente: 1, 2 o 3)
1. Piedra
2. Papel
3. Tijera
1

Tu elección fue: Piedra
Esperando a que el computador elija....
La elección del ordenador fue: Piedra
Comparando resultados .....
EMPATASTE esta ronda
*****

Seleccione una de las tres opciones:
(debe escribir el número correspondiente: 1, 2 o 3)
1. Piedra
2. Papel
3. Tijera
2
```

Planificación - Esquema

- Description de Juego
- Gameplay basico
- Diseño de código
- Librerias
- Otras cosas
- División y planificación de tiempo
- Recomendamos enviar esta hasta el 27/04/2024

Group (Número): (Alexis Rodriguez, Marcel Moran)

Título: Tetris

1. Description de Juego:

El juego se juega en un canvas en el cual debemos encajar figuras geometricas compuestas de cuadrados unidos de forma ortogonal para formar una fila o filas, estas figuras caen desde el tope del canvas y se detienen cuando chocan con la parte baja del canvas o chocan con otras piezas, el jugador puede rotar la pieza en 90 o -90 grados, las piezas que caen del canvas son aleatorias y por tanto el jugador no puede saber exactamente qué pieza le tocará jugar, normalmente el jugador toca la parte baja del canvas.

2. Gameplay basico:

El usuario puede mover las piezas con las teclas izquierda y derecha, con la tecla arriba, la pieza caerá inmediatamente a la base, piezas pueden rotar presionando el boton numerico 0, el juego puede ser pausado con X

3. Diseño de código:

Es programa requiere:

Clase celda subclase de pygame.Rect	
Atributos	Métodos
<code>__free</code> <code>__left</code> <code>__right</code> <code>__up</code> <code>__down</code>	<code>clear</code> <code>set</code>

División y planificación de tiempo

Número de horas esperadas: 14 horas

Alexis	Marcel	Dia
Planificacion	Planificacion	Lunes
Aprender acerca de pygame	Aprender acerca de pygame	Martes
Desarrollo de Celda	Desarrollo de Juego	Miércoles
Desarrollo de Contenedor	Desarrollo de Player	Jueves
Integracion	Integracion	Viernes

Alexis	Marcel	Dia
Depuracion	Depuracion	Lunes
Preparar presentación	Preparar presentación	Martes
		Miércoles
Presentacion	Presentacion	Jueves
		Viernes

Retroalimentación

- Para retroalimentación dirigirse al siguiente enlace <https://forms.gle/7VBKvNqd9kAEG8am8> .
- Déjanos saber qué podemos hacer para mejorar el curso

