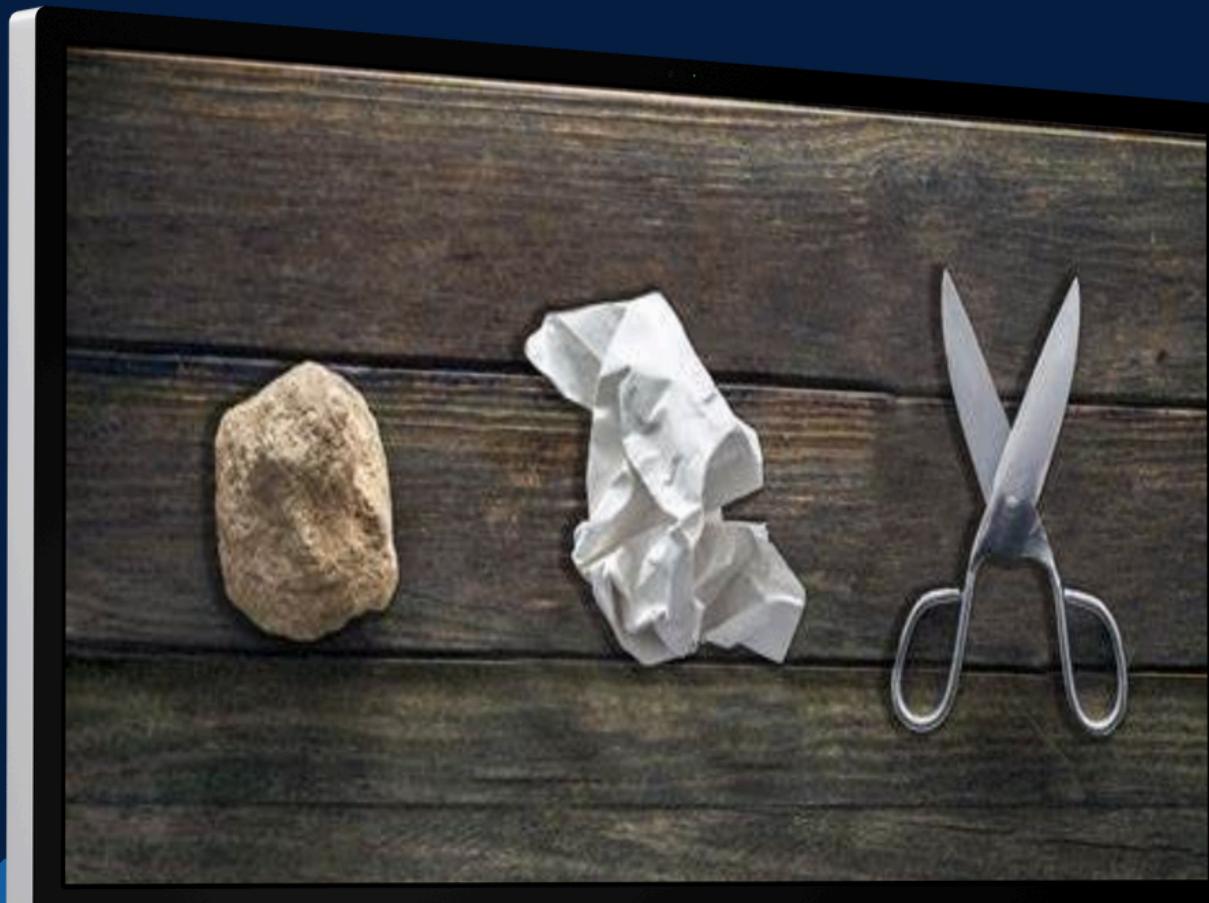


PROGRAMACIÓN EN PYTHON



*DESARROLLO DEL
JUEGO DE PIEDRA,
PAPEL, TIJERA O
DINAMITA*

Por: Bryan D. Ponce B.

A blurred screenshot of a computer screen showing a large amount of Python code. The code appears to be related to game development, specifically for a game like Rock, Paper, Scissors, or Dynamite. It includes functions for marking activities and comparing them.

RESEÑA DEL JUEGO

Este juego es utilizado con mucha frecuencia para decidir quién de dos personas hará algo o solo por rivalidad para ver quien es el ganador y quien el perdedor, es un juego de manos en el que existen tres elementos: la piedra, que vence a la tijera rompiéndola, la tijera, que vence al papel cortándolo, y el papel, que vence a la piedra envolviéndola, dando lugar a un círculo o ciclo cerrado, en la actualidad varias generaciones han implementado nuevas palabras al juego como en este ejemplo utilizariamos la palabra dinamita la cual solo perderia con tijera ya que esta cortaria la mecha.



EN LA ACTUALIDAD

Este juego lo pueden practicar los niños hasta los adultos ya que es juego que ha pasado por generaciones, en la actualidad se lo puede jugar mediante dispositivos electronicos o manual mente



CODIGOS IMPLEMENTADOS

Libreria random

Utilizamos esta librería ya que nos permitirá escoger aleatoriamente por medio de su elemento (random.choice) una secuencia de datos proveida como en este caso lo detallamos a continuación con una lista.

Listas

Utilizamos el metodo (lista.append) para agregar una nueva variable al juego original en esta caso se implemento (dinamita)

```
Users > ASUS > Desktop > UIDE > 5 Logica de la programacion > Piedra Papel Tijera Dinamita.py
1 import random
2
3 lista = ["piedra", "papel", "tijera"]
4 print(lista)
5 lista.append ("dinamita")
6 print(lista)
7
8 lista = ["piedra", "papel", "tijera", "dinamita"]
9
10 while True:
11     pc = random.choice(lista)
12     jugador = None
13
```

CODIGOS IMPLEMENTADOS

Bucle while

Utilizamos el bucle while para ejecutar un código de manera repetitiva mientras se cumple una especificación señalada

Condicional if

El condicional if, funciona verificando la condición ingresada y de acuerdo a su valor de verdadero lleva a cabo o no las instrucciones que hayamos indicado.

```
while jugador not in lista:  
    jugador = input("piedra, papel, tijera o dinamita?: "). lower()  
  
if jugador == pc:  
    print("pc:", pc)  
    print("jugador:", jugador)  
    print("empate!")  
  
elif jugador == "piedra":  
    if pc == "papel":  
        print("pc:", pc)  
        print("jugador:", jugador)  
        print("perdiste!")  
    if pc == "tijera":  
        print("pc:", pc)  
        print("jugador:", jugador)  
        print("ganaste!")
```



CODIGOS IMPLEMENTADOS

```
while jugador not in lista:  
    | jugador = input("piedra, papel, tijera o dinamita?: "). lower()  
  
if jugador == pc:  
    | print("pc:", pc)  
    | print("jugador:", jugador)  
    | print("empate!")  
  
elif jugador == "piedra":  
    | if pc == "papel":  
    |     | print("pc:", pc)  
    |     | print("jugador:", jugador)  
    |     | print("perdiste!")
```

Condicional elif

El condicional elif se utiliza para evaluar múltiples condiciones después de que una condición if resulta falsa, esto permitira escribir y ejecutar diferentes bloques de código según cada condición.

Método lower

El método lower lo utilizaremos para convertir todos los caracteres que el usuario escriba en mayúsculas en minúsculas.



CRONOGRAMA

*JUEGO DE PIEDRA, PAPEL, TIJERA
O DINAMITA*

DESARROLLO DEL JUEGO DE PIEDRA, PAPEL, TIJERA O DINAMITA

Unidad 1

TEMA 1
Empezamos a investigar sobre los problemas , tipos de problemas y sus manera de resolverlo mediante el análisis del mismo.

TEMA 2
Procedemos a escoger el software que vamos a desarrollar y realizar un análisis macro del mismo.

Unidad 2

TEMA 3
Conocimos diferentes lenguajes de programación en especial el de python con el cual vamos a desarrollar nuestro software.

TEMA 4
Una vez analizado el contenido de la clase se configuro el Github y se realizo los diagramas de flujo de nuestro software, junto con el avance del 25% de nuestro código

Unidad 3

TEMA 5
Se estudio herramienta de solución de problema mediante programación, algoritmos y condicionales que se va a implementar en el desarrollo del software

TEMA 6
Se incorporo funcionalidades en estructuras lógicas, y codificación de estructuras repetitivas..

Unidad 4

TEMA 7
En esta semana se participo en un foro para complementar los conocimientos con otros compañeros y dar críticas constructivas, además de obtener conocimientos de las listas, tuplas y diccionario

TEMA 8
Se estudio las funciones para poder completar el desarrollo del software para su posterior verificación final.

GRACIAS A TODOS

