Desc.

Enfocados en el uso de variables deberemos ser capaces de conocer el uso de ellas y como operar entre datos mediante **Asignaciones**, **Tipos de datos**, **Constantes**.

Tipos de datos

La mayoría de los datos siguen el mismo tipado aunque depende del lenguaje que utilicemos pueden haber algunas variaciones como en Py.

Números

- **Int:** números enteros **sin decimales**, pueden ser **positivos** o **negativos** (-10, 0 10, 12, 92...)
- **Float:** números **con decimales** representados con un punto decimal (1.5, 10.1, 3.14, 2.5...)

Cadenas | Strings

- **string:** cadenas de caracteres definidas por comillas simples, dobles o invertidas ("hola", 'esto', `es`)

Decisiones | Booleanos

- **bool:** solo pueden tener dos valores (true | false)

Conjuntos | Listas, Arrays etc..

- **Listas:**
  - Se definen por corchetes **[ ]**
  - Pueden contener elementos de diferentes tipos
  - Permiten duplicados
  - Se accede a los elementos con índice
  - Ordenación y mutación constante de datos

```python
# Lista de nombres
nombres = ['Juan', 'María', 'Pedro', 'Ana', 'Luisa']

# Acceder a los elementos de la lista
print("Primer nombre:", nombres[0])
print("Segundo nombre:", nombres[1])
print("Último nombre:", nombres[-1])

# Modificar un elemento de la lista
nombres[2] = 'Carlos'

# Agregar un nuevo nombre a la lista
nombres.append('Elena')

# Mostrar la lista actualizada
print("Lista de nombres actualizada:", nombres)
```

- **Tuplas:**
  - Se definen por paréntesis **( )**
  - Inmutables
  - Pueden contener elementos de diferentes tipos
  - Se accede a los elementos con índice
  - Colección de datos que no cambian

```python
# Tupla para representar coordenadas
coord = (3, 5)

# Acceder a los elementos de la tupla
print("Coordenada x:", coord[0])
print("Coordenada y:", coord[1])
```

- **Conjuntos:**
  - Colección **NO** ordenada
  - Sin elementos duplicados
  - Pueden contener elementos de diferentes tipos
  - **NO** se puede acceder con índice
  - Colección de datos que no cambian
  - Operaciones de conjuntos, unión, borrado de duplicación de datos para listas, tuplas y otros.

```python
# Conjunto de números
numeros = {1, 2, 3, 4, 5}

# Mostrar el conjunto
print("Conjunto de números:", numeros)

# Agregar un nuevo número al conjunto
numeros.add(6)

# Intentar agregar un número que ya está en el conjunto (ERROR)
numeros.add(5)

print("Conjunto de números actualizado:", numeros)
```

- **Diccionarios:**
  - Se crea mediante **clave-valor**
  - Se definen con llaves **{ }**
  - claves únicas e induplicables
  - Se accede mediante claves
  - Representa datos estructurados

```python
# Diccionario: personaje
personaje = {
    'nombre': 'Aragorn',
    'clase': 'Guerrero',
    'nivel': 50,
    'puntos de vida': 1000,
    'puntos de magia': 200,
    'arma': 'Espada Andúril',
    'armadura': 'Coraza de Mithril',
    'habilidades': ['Corte Mortal', 'Grito de Guerra', 'Defensa Impenetrable']
}

# Acceder a los valores
print("Nombre del personaje:", personaje['nombre'])
print("Clase del personaje:", personaje['clase'])
print("Nivel del personaje:", personaje['nivel'])
print("Puntos de vida del personaje:", personaje['puntos de vida'])
print("Puntos de magia del personaje:", personaje['puntos de magia'])
print("Arma del personaje:", personaje['arma'])
print("Armadura del personaje:", personaje['armadura'])
print("Habilidades del personaje:", personaje['habilidades'])
```

Calculadora de poder de ataque (PA)

Disponemos de un personaje con diferentes atributos su PA o poder de ataque será calculado dependiendo de los atributos que posea este.

Procedimiento

- Hacer uso de una constante en el programa
- Mostrar el resultado en terminal
- Hacer uso de la función `input()`

Ejercicio

```
# Solicitar al usuario que ingrese el daño base, nivel del personaje y bonificación

# Operación

# Mostrado en pantalla
```

Conversor de puntuación de experiencia (XP)

En nuestro juego disponemos de 3 clases de enemigos, dependiendo del tipo de enemigo que matemos recibiremos más o menos experiencia, además también tenemos que tener en cuenta que cuanto mayor sea nuestro nivel menos experiencia recibiremos (factor de reducción).

Formula

XP Total=(XP Ligero+XP Medio+XP Pesado) × Multiplicador de Niv

Enemigos

- Ligero: 5
- Medio: 10
- Pesado: 30

Ejercicio

```
# Definir las cantidades de experiencia para cada tipo de enemigo, nivel del
jugador

# Operación

# Resultado por pantalla
```

Resultado

Ejercicio 1

```
# Solicitar al usuario que ingrese el daño base, nivel del personaje y bonificación
DAÑO_BASE = 25
nivel_personaje = int(input("Por favor, ingrese el nivel del personaje: "))
bonificacion = float(input("Por favor, ingrese la bonificación del personaje: "))

# Operación con el PA
pa = (DAÑO_BASE + nivel_personaje) * bonificacion

# Mostramos el resultado
print("El Poder de Ataque (PA) del personaje es:", pa)
```

Ejercicio 2

```
# Experiencia de los enemigos
XP_LIGERO = 5
XP_MEDIO = 10
XP_PESADO = 30

# Nivel del jugador
nivel = int(input("ingresa el nivel del jugador: "))

# Multiplicador de nivel (suponiendo un 10% de reducción por nivel)
multiplicador_nivel = (1 - nivel * 0.1)

# Calcular la experiencia total
XP_TOTAL = (XP_LIGERO + XP_MEDIO + XP_PESADO) * multiplicador_nivel

# Mostrar el resultado
```

```
print("La experiencia total obtenida al matar a los enemigos es:", XP_TOTAL)
```