

Prueba técnica – Ejercicios en Python

Ejercicio 1)

Encontrar el número máximo en un array de números aleatorios

Descripción:

Desarrolla un programa en Python que genere un array de 10 números aleatorios enteros entre un rango especificado (por ejemplo, entre 1 y 100). Y el programa debe de imprimir la lista de números enteros, y después indicar cual es el número máximo del ejercicio e indicar su posición en el array.

Prohibido:

- Usar la función **max()**

Consejos:

- Puedes utilizar la función **random.randint()** para generar un número aleatorio dentro de un rango específico.

Ejemplo de ejecución:

Array generado: [4, 2, 3, 6, 7, 5, 5, 10, 8, 10]

El número más alto de la lista es: 10 y está en la posición 7 del array

Ejercicio 2)

Calculadora de IMC

Descripción:

El índice de masa corporal (IMC) es una medida que se utiliza para calcular la cantidad de grasa corporal de una persona. Se calcula dividiendo el peso de una persona en kilogramos por su altura en metros al cuadrado.

Objetivo:

Escribir un programa en Python que calcule el IMC de una persona a partir de su peso y altura introducidos por el usuario.

Requisitos:

- El programa debe solicitar al usuario que introduzca su peso en kilogramos.
- El programa debe solicitar al usuario que introduzca su altura en metros.
- El programa debe calcular el IMC utilizando la siguiente fórmula:

```
imc = peso / (altura ** 2)
```

- El programa debe mostrar el IMC calculado al usuario.

- El programa debe mostrar un mensaje que interprete el valor del IMC según la siguiente tabla:

IMC	Interpretación
Menos de 18.5	Bajo peso
Entre 18.5 y 24.9	Peso normal
Entre 25 y 29.9	Sobrepeso
30 o más	Obesidad

Consejos:

- Puedes utilizar la función **float()** para convertir las entradas del usuario de cadenas a números.
- Puedes utilizar la función **round()** para redondear el valor del IMC a dos decimales.
- Puedes utilizar la función **input()** para pedir una entrada de valor al usuario

Ejemplo de ejecución:

```
Introduzca su peso en kg: 70
Introduzca su altura en metros: 1.7
Su IMC es: 23.82
Su peso está dentro del rango de peso normal.
```

Ejercicio 3)

Juego de Adivinanzas

Descripción:

Escribir un programa en Python que implemente un juego de adivinanzas clásico. El programa debe pensar en un número aleatorio entre un rango especificado por el usuario y luego pedirle al usuario que adivine el número. El programa debe dar pistas al usuario si su suposición es demasiado alta o baja hasta que el usuario adivine correctamente el número.

Objetivo:

Practicar el uso de bucles, condiciones e interacción con el usuario en Python.

Requisitos:

- El programa debe solicitar al usuario que introduzca un límite inferior para el rango de números aleatorios.
- El programa debe solicitar al usuario que introduzca un límite superior para el rango de números aleatorios.
- El programa debe generar un número aleatorio dentro del rango especificado.
- El programa debe solicitar al usuario que introduzca una suposición.
- El programa debe comparar la suposición del usuario con el número aleatorio:

- Si la suposición es demasiado alta, el programa debe mostrar un mensaje que indique que el usuario debe adivinar un número más bajo.
- Si la suposición es demasiado baja, el programa debe mostrar un mensaje que indique que el usuario debe adivinar un número más alto.
- Si la suposición es correcta, el programa debe mostrar un mensaje de felicitación al usuario y finalizar.
- El programa debe repetir el proceso de solicitar suposiciones y dar pistas hasta que el usuario adivine correctamente el número.

Consejo:

- Puedes utilizar la función **random.randint()** para generar un número aleatorio dentro de un rango específico.

Ejemplo de ejecución:

Bienvenido al juego de adivinanzas!

Introduzca un límite inferior para el rango: 10

Introduzca un límite superior para el rango: 100

¡He pensado en un número! Adivina cuál es: 50

¡Tu suposición es demasiado alta! Intenta con un número más bajo.

Adivina cuál es: 25

¡Tu suposición es demasiado baja! Intenta con un número más alto.

Adivina cuál es: 38

¡Felicidades! Has adivinado el número correctamente. ¡Era el 38!

Ejercicio 4)

Juego de Adivinanzas con palabras

Descripción:

Desarrollar un programa en Python que implemente un juego de adivinanzas con palabras. El programa debe seleccionar una palabra aleatoria de un diccionario y luego pedirle al jugador que adivine la palabra letra por letra. El jugador tiene un número limitado de intentos para adivinar la palabra correctamente.

Objetivo:

Practicar el uso de cadenas de texto, bucles y condiciones en Python.

Requisitos:

- El programa debe guardar como variable el siguiente diccionario de palabras:

```
palabras = ['periquito', 'comida', 'jamon', 'cocina',  
'programacion', 'trabajador', 'examen', 'deportista']
```

- El programa debe seleccionar una palabra aleatoria del diccionario.
- El programa debe mostrar la palabra al jugador con las letras ocultas por guiones bajos (_).
- El programa debe solicitar al jugador que ingrese una letra.
- El programa debe verificar si la letra ingresada está en la palabra oculta.
- Si la letra está en la palabra, el programa debe mostrar la palabra actualizada con la letra revelada.
- Si la letra no está en la palabra, el programa debe disminuir el número de intentos restantes.
- El juego termina cuando el jugador adivina la palabra correctamente o cuando se agotan los intentos.
- El programa debe mostrar un mensaje de felicitación o derrota al final del juego.

Ejemplo de ejecución:

```
¡Bienvenido al juego de adivinanzas!
```

```
He pensado en una palabra de 6 letras. Adivina cuál es:
```

```
_ _ _ _ _ _
```

```
Tienes 6 intentos restantes.
```

```
Introduce una letra: a
```

```
Palabra: _ a _ a _ a
```

```
Tienes 5 intentos restantes.
```

```
Introduce una letra: e
```

```
¡La letra e no forma parte de la palabra!
```

```
Palabra: _ a _ a _ a
```

```
Tienes 4 intentos restantes.
```

```
...
```

```
¡Felicidades! Adivinaste la palabra: "banana" en 4 intentos.
```

Ejercicio 5)

Algoritmo de conteo de palabras

Descripción

Desarrolla un programa en Python que realice un análisis de frecuencias de palabras en un texto.

Requisitos:

- El programa debe guardar como variable y leer el siguiente texto:

```
texto = ""Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ac diam ut justo aliquam iaculis id eu magna. Cras nulla lectus, tristique non imperdiet eget, aliquam quis sem. Aliquam suscipit, ex eu elementum efficitur, ante ipsum finibus metus, vitae blandit massa tellus et diam. Maecenas porta feugiat efficitur. Etiam ut diam quis velit malesuada vulputate. Quisque in risus gravida, finibus massa vel, ultrices nisl. Ut placerat semper ligula id aliquam. Donec ac commodo risus, a tincidunt dui. Phasellus at nisi nulla. Nulla facilisi. Fusce in tristique est, at convallis leo.
```

Nulla facilisi. Etiam ac convallis risus. Nulla faucibus rhoncus justo, eu scelerisque ex luctus sit amet. Duis imperdiet dui quis dui luctus pretium. Duis volutpat orci ultricies orci posuere ultrices. Donec eget turpis aliquet, varius enim quis, commodo nibh. Vivamus mi felis, semper a felis vitae, rhoncus sodales justo. In condimentum, justo quis venenatis congue, sem purus congue quam, in bibendum elit sem ut nibh. Nullam lorem sem, porta vitae nunc in, commodo semper magna. Praesent hendrerit egestas nulla a ultrices. Etiam sed eleifend nibh. Sed varius rutrum diam at vulputate. Etiam vulputate eu lacus quis varius.

Vestibulum egestas eleifend scelerisque. Aliquam erat volutpat. Aliquam posuere non erat eget mollis. Suspendisse et eleifend neque. Sed quis orci eu sem auctor ultrices sit amet sit amet leo. Nunc convallis, est id vulputate facilisis, felis ex tempor sem, a vestibulum ipsum nulla at mauris. Phasellus leo felis, interdum id enim sit amet, eleifend cursus urna.

Praesent urna leo, pretium in tellus in, venenatis volutpat justo. Maecenas est est, auctor quis dui et, porta volutpat libero. Quisque tincidunt fermentum lorem, at tempus sem viverra ac. Aliquam odio sapien, tincidunt ut magna quis, pellentesque dictum mi. Mauris a ligula congue, fringilla mi at, tincidunt lorem. In eget nibh et diam tempus tincidunt. Praesent facilisis, ligula in mattis sodales, ipsum massa auctor velit, sed maximus purus magna sit amet sapien. Cras dapibus neque tempor massa aliquet, id finibus ligula molestie. Nullam nec erat turpis. Integer a consequat mi, sed placerat leo.

Maecenas sit amet justo lacus. Praesent vitae libero velit. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Sed quis felis volutpat, sodales quam in, suscipit sapien. Vivamus pellentesque eget ligula nec blandit. Fusce rutrum id diam et pulvinar. Integer lobortis ante id congue aliquam. Integer fermentum feugiat ultrices. Morbi rhoncus vestibulum semper. Vivamus eu erat dapibus, ultricies diam nec, accumsan enim. Praesent purus ipsum, viverra et ipsum sit amet, vulputate laoreet tellus. Phasellus eu nibh fringilla, convallis leo id, aliquam diam. Nullam elit erat, imperdiet nec arcu convallis, sodales hendrerit est. Aenean eget consectetur libero."""

- Preprocesar el texto: El programa debe preprocesar el texto para eliminar caracteres especiales, puntuación y convertir todo a minúsculas.
- Tokenizar el texto: El programa debe tokenizar el texto, es decir, dividirlo en palabras individuales.
- Crear un diccionario de frecuencias: El programa debe crear un diccionario que almacene cada palabra única del texto y su correspondiente frecuencia de aparición.
- Ordenar el diccionario por frecuencias: El programa debe ordenar el diccionario por las frecuencias de las palabras, de mayor a menor.
- Presentar los resultados: El programa debe presentar los resultados del análisis de frecuencias, por ejemplo, mostrando las 10 palabras más frecuentes.

Prohibido:

- Usar la función **sorted()**

Resultado de ejecución:

Top 10 palabras más usadas:

1. quis: 10 usos
2. in: 10 usos
3. aliquam: 9 usos
4. sit: 8 usos
5. amet: 8 usos
6. diam: 8 usos
7. id: 8 usos
8. eu: 7 usos
9. nulla: 7 usos
10. sem: 7 usos