Día 4

Temas

1. Funciones
   1. ¿Para qué sirven? (intro.)
   2. Return

Actividades en la mañana

1. Receta de cocina
   1. Describir una receta de cocina en codigo
2. Telefono descompuesto de funciones
   1. Les das un input a las niñas
   2. Ellas dicen el output
   3. El grupo debera de adivinar la función
3. Ruta por medios de transporte
   1. Se le da un tablero a cada niña
   2. Se les da un presupuesto
   3. Se especifica cuanto puede avanzar cada transporte y su costo
   4. Se especifica el inicio y el destino final

Actividades de programación

**Funciones**

Mi primera función

Crea un archivo y nómbralo Funciones.py

Crea una función llamada funcion sin parámetros: def funcion():

El objetivo de esta función será imprimir “esta es tu primera función”.

Manda llamar la función. Para esto solo escribe su nombre y los paréntesis vacíos (porque la función no requiere parámetros.)

Crea una nueva función llamada estatura con el parámetro “altura”.

Esta función debe imprimir la frase “Mides {altura} metros".

Llama la función pero dentro de los paréntesis escribe tu altura, ejemplo: estatura(1.60)

Observa que se sustituyó el valor de altura que estaba dentro de la función.

Ahora haz una nueva función llamada “persona” con los parámetros “nombre”, “edad” y “altura”. Deben estar separados por comas.

Esta función deberá imprimir tus datos en una frase, como lo hemos hecho antes.

Llama la función y completa con tus datos en el paréntesis, en el orden que la función lo indica y separados por una coma.

Ahora juega con eso, cambia los parámetros cuando llames las funciones y familiarízate con como funciona.

Solución: 1\_Funciones

Los cuadarados

1. Crea un archivo y nómbralo Cuadrado.py
2. Crea una función de nombre cuadrado.

Esta función debe recibir el número (como parámetro, claro) y sacar su cuadrado.

No crees variables nuevas, reasigna el valor a la que ya existe (número)

¿Cómo lo harías?

1. Al final, la variable debe devolver el resultado. Para hacer esto escribe al final

return numero

1. Crea una nueva variable (fuera de la función) y nómbrala “respuesta”. Cuando le asignes el valor, llama la función “cuadrado”, pero como parámetro ingresa un input con la clase que le corresponda (¿Que tipo de variable es: str, float, int?) y ese input le pide al usuario que ingrese un número.
2. Finalmente, imprime una frase que indique al usuario cuál es el cuadrado del número que ingresó.

Tip: la respuesta está contenida dentro de la variable “respuesta”

Solución: 2\_Return

Mi primera función

1. Crea un archivo y nómbralo FuncionRandom.py

Será el juego de adivinar los números, pero ahora usando funciones.

1. Escribe import random

Esto habilita que puedas usar la opción de elegir un número random.

1. Crea una variable llamada random y cuando le asignes el valor, que sea un número random entre 1 y 100.
2. Indícale al usuario de qué se trata el juego

print("Adivina el numero que estoy pensando")

1. Crea una función llamada atinale con el parámetro “num”
2. Dentro de esa función hay 3 opciones
   1. Si num es igual a random le indicas al usuario que ganó el juego y sales del mismo con “return”
   2. Si num es mayor o menor que random, crea una variable nueva, puede ser num1, y asígnale como valor un input que le pida al usuario un número mayor o menor según sea el caso.

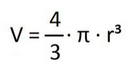
Vuelve a llamar la función “atinale” pero con el nuevo parámetro num1.

1. Finalmente, para que la función comienze, pídele al usuario el primer número, y luego llama la función con ese número como parámetro.

Solución: 3\_Random\_Funciones

Volumen esfera (Nivel fácil)

Esta función calcula el volumen de una esfera dado su radio.

* Crea un archivo y nómbralo VolumenEsfera.py
* Define la función llamada calcularVolumenEsfera
* La función recibe un parámetro llamado r, que corresponde al radio de la esfera.
* Calcula el volumen mediante la fórmula:
  + 
* Retorna el valor del volumen

Solución: PH2\_dia4\_facil\_1.py

Múltiplos comunes (Nivel intermedio)

Esta función recibe 3 números. Los dos primeros corresponden a los números de los cuáles buscas sus múltiplos en común. El tercer número es un límite. Por ejemplo, si recibe 6, 10 y 100, la función deberá encontrar los múltiplos comunes de 6 y 10 que son menores a 100, ponerlos en una lista y regresar esta lista como resultado de la función.

* Crea un nuevo archivo y nómbralo MultiplosComunes.py
* Define la función llamada multiplos\_comunes
* La función recibe 3 parámetros, indícalos como: num1, num2 y lim.
* Dentro de la función crea una lista vacía llamada multiplos
* Crea un ciclo for en el que recorras desde 0 hasta lim, de la siguiente manera:
  + for indice in range(0, lim)
* Crea dos condicionales anidadas. En una debes asegurar que se cumpla: indice % num1 == 0. En la otra que índice % num2 == 0
* Si se cumplen ambas condiciones, debes añadir índice a la lista de múltiplos.
* Al final retorna la lista de múltiplos.

Solución: PH2\_dia4\_intermedio\_1.py

Promociones (Nivel intermedio)

Un servicio de internet tiene un sistema de recompensas que funciona de la siguiente manera:

* Si cargas menos de 10 dólares, no obtienes nada extra.
* Si cargas entre 10 y 25 dólares (inclusive) obtienes un 3% extra.
* Con una carga entre 25 y 50 (inclusive), obtienes 8% extra.
* Con una carga mayor a 50 dólares, obtienes 20% extra.

Procedimiento:

* Crea un archivo nuevo y nómbralo Promociones.py
* Define una función llamada ofertas
* La función recibe un parámetro correspondiente al monto a cargar y retorna el monto total (incluyendo la bonificación correspondiente).
* El cuerpo de la función consta de una estructura del tipo: if-elif-else. (Ver el código en la solución).

Solución: PH2\_dia4\_intermedio\_2.py

Barra de progreso (Nivel intermedio)

Esta función despliega una barra de progreso en la terminal de Python.

Hará esto mediante la impresión de una X cada “n” cantidad de segundos hasta llegar a un límite de tiempo establecido. Por ejemplo: Si el tiempo es 50 segundos, y n = 5, a los 5 segundos se imprime una X, a los 10 se verá XX y así hasta que se llegue a los 50 segundos: XXXXXXXXXX.

* Crea un nuevo archivo y nómbralo BarraProgreso.py
* Escribe: import time en la primera línea del programa. Esa librería es la que va a permitir a la función contar los segundos en tiempo real.
* Define la función llamada barra\_progreso.
* La función recibe 2 parámetros. El primero se llama segundos. Que representa la cantidad total de tiempo. El segundo se llama n, que es la cantidad de tiempo que pasará entre cada impresión de una X.
* Crea una variable llamada impresiones, que corresponde a la cantidad total de “X” que se van a imprimir. Y se calcula mediante la división entera entre el parámetro impresiones y 5 (Recuerda que se va a imprimir una X cada “n” segundos).
* Crea un ciclo for que irá desde 0 hasta la cantidad de impresiones calculada en el paso anterior.
* Dentro del ciclo usa la operación: time.sleep(n). Lo que hace esta operación es hacer una pausa de n segundos en tiempo real antes de ejecutar la siguiente línea de código.
* Realiza la impresión de “X” mediante un print con end=””
* Esta función no retorna nada.

Solución: PH2\_dia4\_intermedio\_3.py

Búsqueda binaria (Reto)

La búsqueda binaria es un algoritmo muy eficiente para realizar búsquedas en listas que se encuentran ordenadas. Funcionan a través de dividir el problema, descartando la mitad de los elementos de la lista en cada iteración. Los pasos para realizar una búsqueda binaria son:

* Ver el elemento que se encuentra a la mitad de la lista, si el elemento que busco es mayor, descarto la primera mitad de la lista y busco en la segunda mitad. (Nuevamente en el elemento que se encuentre a la mitad de ese segmento).

Si el elemento es menor, descarto la segunda mitad del segmento y busco en el elemento que se encuentra en la mitad que me interesa. Continúo de esa manera hasta encontrar el elemento o descubrir que no se encuentra en la lista.

* Con el algoritmo anterior, se minimiza el tiempo que toma la búsqueda. Para poder implementarlo siempre debemos mantener dos índices que nos marquen la posición inicial y final del segmento de la lista en donde estamos buscando, de manera que siempre podamos encontrar el elemento central.

Instrucciones:

* Crea un nuevo archivo llamado BusquedaBinaria.py
* Define una función llamada busqueda\_binaria.
* La función recibe dos parámetros. El primero es la lista ordenada de elementos y el segundo es el elemento que busco.
* Retorna el índice en el que se encontró el elemento (en caso de encontrarlo)
* Retorna -1 si no se encuentra el elemento en la lista.

Solución: PH2\_dia4\_reto\_1.py

Empiezan con: (Reto)

Diseña una función que recibe una lista de strings y una letra, y regresa cuántas palabras de la lista empiezan con la letra indicada.

Solución: PH2\_dia4\_reto\_2.py