

KoutLab

PROGRAMACIÓN EN PYTHON AVANZADA



By Aerobot

Para una mayor visualización se recomienda crear un perfil de alumno para el docente.

OBJETIVO GENERAL

El alumno comprenderá los fundamentos de la programación en Python incorporándose desde un escenario elemental. Dando a conocer conceptos básicos, características y lenguajes de desarrollo de forma que el estudiante pueda ir involucrándose en procesos de la programación en Python.

RUTA DE APRENDIZAJE

El alumno contará con una ruta de aprendizaje con cápsulas cada una de ellas con diferentes temas y actividades.



- **Introduccion:** Presentacion de inicio a cada tema principal.
- **Cápsula teórica:** Se dara una introduccion del tema,
- **Cápsula práctica:** contara con una practica a realizar, así como un video de apoyo.
- **Cápsula juego:** verá un juego relacionado al tema.
- **Cápsula evaluativa:** Una vez acumulados varios temas aprendidos se evaluara al alumno para poder avanzar a otra seccion.

TEMARIO

CAPITULO 1

- TEMA 1: Variables 2.0
- TEMA 2: Diccionarios 2.0
- TEMA 3: Funciones 2.0
 - TEMA 3.5: Creando funciones propias
- TEMA 4: Funciones integradas
 - TEMA 4.5: Funciones Lambda
- TEMA 5: Trabajando con archivos
- TEMA 6: Trabajando con gráficos
- TEMA 7: Expresiones regulares
 - TEMA 7.5: Analizando Expresiones regulares

CAPITULO 2

- TEMA 1: Eliminar strings vacíos
- TEMA 2: Asignar múltiples valores a múltiples variables
- TEMA 3: Asignar un mismo valor a múltiples variables
- TEMA 4: Asignación de valores a múltiples listas y tuplas
- TEMA 5: Convertir tipos datos
 - TEMA 5.5: convertir listas, tuplas y sets
- TEMA 6: listas anidadas
 - TEMA 6.5: Decoradores
- TEMA 7: Atributos y métodos
 - TEMA 7.5: Metaclases

TITULOS Y DIVISORES

Cápsula Teórica

Pregunta:
¿Para que sirven las variables?

Respuesta:
Guardar información para usarla mas tarde en un programa.

Cápsula práctica

Intrucción:
Crea una función que tome un número como argumento y devuelva True si el número es par o False si el número es impar.

Respuesta:

```
1\n2\n3\n4\n5\ndef es_par(numero):\n    if numero % 2 == 0:\n        return True\n    else:\n        return False
```

Cápsula juegos

Juego:
Copiar código

Respuesta:

```
edad = 9\nnombre = "Hanna"\nprint("Hola, mi nombre es " + nombre + " y tengo " + edad + " años")
```

FORMATO DE TEXTO

Cápsula Teórica

Pregunta:

¿Para que sirven las variables?

Respuesta:

Estructura de datos que permite almacenar elementos en pares clave-valor.

Cápsula práctica

Intrucción:

Agregar un párrafo con
Crea una función que tome una frase como argumento y devuelva el número de letras en la frase (busca como usar len(), es una función que ayuda).

Respuesta:

```
1\n2\n3\n4\ndef contar_letras(frase):\n frase_sin_espacios = frase.replace(" ",\n    "")\n numero_de_letras =\n len(frase_sin_espacios)\n return\n numero_de_letras
```

Cápsula juegos

Juego:
Memorama

Respuesta:
N/A

LISTAS ORDENADAS Y NO ORDENADAS

Cápsula Teórica

Pregunta:
¿Qué es una función en programación?

Respuesta:
Bloque de código reutilizable que realiza una tarea específica.

Cápsula práctica

Intrucción:
Crea una función que tome dos números como argumentos y devuelva el mayor de los dos.

Respuesta:

```
1\n2\n3\n4\n5\ndef mayor(numero1, numero2):\n    if numero1 > numero2:\n        return numero1\n    else:\n        return numero2
```

Cápsula juegos

Juego:
Seleccionar la respuesta correcta
(Preguntas aleatorias)

Respuesta:
P: _____('Maria')
R: saludar

P: Para crear una función, es...
R: def

P: Una función puede ejecutarse dos veces o más...
R: ¡Verdadero!

P: Un bloque de código que inicia...
R: una función

P: Puedo mandar a llamar una función que...
R: ¡imposible!

LINEA DE TIEMPO

Cápsula Teórica

Pregunta:
¿Para qué sirven las funciones específicas?

Respuesta:
Son funciones que se crean para realizar tareas específicas.

Cápsula práctica

Intrucción:
Crea otra función llamada "es_par" que tome un argumento (n) y regrese True si n es par y False si es impar.

Respuesta:

```
1\n2\n3\n4\n5\n6\n# Programación avanzada con Python\ndef es_par(n):\n    if n % 2 == 0:\n        return True\n    else:\n        return False
```

Cápsula juegos

Juego:
Copiar código

Respuesta:

```
palabra = "hola"\nlongitud = len(palabra)\nprint("La palabra", palabra, "tiene", longitud, "letras.")
```

IMAGENES

Cápsula Teórica

Pregunta:

¿Para qué sirven las funciones integradas?

Respuesta:

Sirven para realizar diversas tareas, como operaciones matemáticas, manipulación de cadenas de texto, entrada y salida de datos, manejo de

Cápsula práctica

Intrucción:

Utiliza la función len() para obtener la cantidad de elementos que hay en una lista.

Respuesta:

```
1\n2\nlista = ["Hola",  
"Mundo"]\nprint(len(lista))
```

Cápsula juegos

Juego:
Laberinto

Respuesta:
N/A

HIPERVÍNCULOS

Cápsula Teórica

Pregunta:

¿Cuándo se utiliza la función lambda?

Respuesta:

Se utilizan cuando se necesita tener una operacion simple o relaizar una terea rapida.

Cápsula práctica

Instrucción:

1.Crea una lista de números llamada numeros con los valores [1, 2, 3, 4, 5].

2. Usa una función lambda para calcular el cuadrado de cada número en la lista `numeros` y almacénalos en una nueva lista llamada `cuadrados`.

3. Imprime la

Respuesta:

```

n2\n3\nnúmeros = [1, 2, 3, 4,
5]\ncuadrados = list(map(lambda x:
x**2, números))\nprint(cuadrados)

```

Cápsula juegos

Juego:

Respuestas a enunciados

Respuesta:

1. matplotlib
2. import matplotlib.pyplot as plt
3. bar() y show()

MENU

Cápsula Teórica

Pregunta:
¿Qué línea de código se ocupa para cerrar el archivo?

Respuesta:
`archivo.close()`

Cápsula práctica

Intrucción:
1.Crea un archivo de texto llamado "mi_archivo.txt" y escribe en él algunas líneas de texto.
2.Abre el archivo y lee su contenido línea por línea, imprimiendo cada línea en la consola.

Respuesta:

```
1\n2\n3\n4\nfh =  
open("mi_archivo.txt")\nfor linea in  
fh.readlines():\n    print(linea)\nfh.close()
```

Cápsula juegos

Juego:
Copiar código

Respuesta:

```
import re  
  
texto = "Ana tiene una manzana y un  
abanico"  
patron = r"/ba/w*"  
resultado = re.findall(patron, texto)  
  
print(resultado)
```

VIDEO

Cápsula Teórica

Pregunta:

Para que sirve el modulo matplotlib

Respuesta:

Para crear graficos mas complejos, como graficos de barras, lineas o de dispersión.

Cápsula práctica

Intrucción:

Grafica los siguientes datos:

```
# Datos a graficar
x = ['Manzana', 'Banana', 'Naranja', 'Pera', 'Mango']
y = [20, 15, 25, 10, 30]
```

Respuesta:

```
1\n2\n3\n4\n5\nimport matplotlib.pyplot as plt\nx = ["Manzana", "Banana", "Naranja", "Pera", "Mango"]\ny = [20, 15, 25, 10, 30]\nplt.bar(x,y)\nplt.show()
```

Cápsula juegos

Juego:
Rompecabezas (Premium 1)

Respuesta:

N/A

PDF

Cápsula Teórica

Pregunta:
¿Qué son las expresiones regulares en programación?

Respuesta:
Son secuencias de caracteres que permiten buscar y manipular patrones de texto de manera eficiente.

Cápsula práctica

Intrucción:
1.Pedir al usuario que ingrese una contraseña.
2.Verificar si la contraseña cumple con los requisitos utilizando expresiones regulares.
3.Si la contraseña cumple con los requisitos, imprimir un mensaje indicando que la contraseña es válida.
4.Si la contraseña

Respuesta:

1

Cápsula juegos

Juego:
Preguntas ágiles (Premium 2) (Preguntas aleatorias)

Respuesta:

P: ¿Para que sirve la palabra lambda...?

R: Para declara funciones en la...

P: ¿Cuál es la forma correcta de...?

R: suma = lambda a, b: a + b

P: Las funciones lambda se pueden mandar...

R: ¡Falso!

P: suma = ____ a, b: a + b

R: lambda

P: Puedo mandar a llamar una función lambda varias....

R: ¡Verdadero!

BOTONES

Cápsula Teórica

Pregunta:

¿Para qué se utilizan las expresiones regulares?

Respuesta:

Se utilizan para buscar, manipular y procesar patrones de texto de manera eficiente.

Cápsula práctica

Intrucción:

Crea un programa que lea una cadena de texto ingresada por el usuario y verifique si contiene un número de teléfono con el formato "(123) 456-7890".

Respuesta:

```
1\n2\n3\n4\n5\ntexto = input("Ingresa una cadena de texto: ")\nif "hola" in texto.lower():\n    print("La cadena de texto contiene la palabra hola.")\nelse:\n    print("La cadena de texto no contiene la palabra hola.")
```

Cápsula juegos

Juego:
Crucigrama
(Premium 3)

Respuesta:

Horizontal:

1. Python
2. Regular
3. Control

Vertical:

1. findall

Pregunta 1:

Una página con extensión htm o html comienza por html (entre signos mayor y menor)

Respuesta:
Verdadero

Pregunta 2:

¿En donde se encuentra el title en una pagina web?

Respuesta:
Body

Pregunta 3:

¿Qué etiqueta utilizamos para insertar una línea horizontal?

Respuesta:
<hr>

Pregunta 4:

¿Cuál es la forma correcta de insertar un comentario?

Respuesta:
<!--...-->

Pregunta 5:

Elija la forma correcta de ingresar una imagen.

Respuesta:

Pregunta 6:

¿Qué etiqueta define un salto de línea?

Respuesta:

Pregunta 7:

Elija la etiqueta que nos da el título más grande

Respuesta:
<h1></h1>

Pregunta 8:

Para definir el fondo de la página se utiliza:

Respuesta:
<body bg color="#00ff00">

Pregunta 9:

En el body se puede:

Respuesta:
Insertar imágenes, texto, tablas y fondos

Pregunta 10:

¿En la etiqueta center que se permite?

Respuesta:
Permite centrar el texto.

CONECTAR HTML Y CSS

Cápsula Teórica

Pregunta:

¿Qué método es el correcto para eliminar los strings vacíos?

Respuesta:

`remove()`

Cápsula práctica

Instrucción:

Crea una lista llamada `frutas` que contiene algunos elementos vacíos (`""`). Luego utiliza la función `filter()` para eliminar todos los elementos vacíos. La función `filter()` toma dos argumentos: una función y una lista

Respuesta:

```
frutas = ["Manzana", "", "Pera", ""]  
frutas = filter(None, frutas)  
print(list(frutas))
```

Cápsula juegos

Juego:

Seleccionar la respuesta correcta (Preguntas aleatorias)

Respuesta:

P: ¿Qué método podemos ocupar para...?

R: `remove()`

P: ¿Cuál de las siguientes respuestas es un arreglo?

R: `numeros = [1, 4, 5, 7]`

P: ¿Podemos borrar espacios vacíos...?

R: ¡Verdadero!

P: ¿Puedo eliminar espacios vacíos de un arreglo...?

R: ¡Verdadero!

P: ¿`remove()` es la única forma de eliminar...?

R: ¡No! existen muchos más

SINTAXIS DE CSS

Cápsula Teórica

Pregunta:
Selecciona la opción correcta al guardar información.

Respuesta:
`nombre,edad,ciudad="Juan",25,"Bogotá"`

Cápsula práctica

Instrucción:
Asigna tres valores a tres variables diferentes en una sola línea de código. Luego, utiliza la función `print()` para imprimir el valor de cada variable.

Respuesta:
`1\n2\nvariable1, variable2, variable3 = 10, "Hola", True\nprint("El valor 1: {variable1}, El valor 2: {variable2}, El valor 3: {variable3}")`

Cápsula juegos

Juego:
Copiar código

Respuesta:
`nombre, edad, ciudad = "Luis", 12, "Puebla"\nnombre, edad, ciudad = "Marcela", 11, "Puebla"`

BOX MODEL

Cápsula Teórica

Pregunta:

¿Qué ocurre al asignar el mismo valor a una variable en Python?

Respuesta:

La variable simplemente conserva el valor anterior y no se produce ningún cambio en su contenido.

Cápsula práctica

Intrucción:

1. Asignar un mismo valor a múltiples variables

2. Cambiar el valor de una variable

3. Mostrar los valores actualizados de todas las variables

Respuesta:

```
1\n2\n3\nvariable1 = variable2 =  
variable3 = "Hola"\nvariable2 =  
0\nprint(variable1, variable2, variable3)
```

Cápsula juegos

Juego:

Repuestas a enunciados

Respuesta:

1. a = b = c = 12
2. ¡Imposible!
3. ¡Verdadero!

VARIABLES DE CSS

Cápsula Teórica

Pregunta:

¿Cuál es la diferencia entre la asignación de valores múltiples en listas y tuplas?

Respuesta:

Tanto las listas como las tuplas permiten la asignación de valores múltiples

Cápsula práctica

Instrucción:

1.Crea dos listas vacías: una para nombres y otra para edades.

2.Crea dos tuplas con los mismos valores para los nombres y las edades que quieras agregar a las listas.

3.Asigna los valores de las tuplas a las listas con la función

Respuesta:

```
1\n2\n3\n4\n5\n6\n7\n8\nnombres = []\nedades = []\ntupla_nombres = ("Juan", "María", "Carlos")\ntupla_edades = (25, 30, 35)\nnombres.extend(tupla_nombres)\nedades.extend(tupla_edades)\nprint("Lista de nombres: ", nombres)\nprint("Lista de edades: ", edades)
```

Cápsula juegos

Juego:
Memorama

Respuesta:
N/A

SELECTORES CSS

Cápsula Teórica

Pregunta:

¿Qué es la conversión de tipos de datos en Python?

Respuesta:

Es una operación importante que permite convertir un tipo de dato a otro tipo.

Cápsula práctica

Instrucción:

Convierte lista_numeros en una tupla llamada tupla_numeros.

Respuesta:

```
1\n2\n3\nlista_numeros = [1, 2, 3, 4, 5]\ntupla_numeros = tuple(lista_numeros)\nprint(tupla_numeros)
```

Cápsula juegos

Juego:
Copiar código

Respuesta:

```
numero_string = "10"\nnumero_entero = int(numero_string)\nprint(numero_entero)
```

SELECTOR ETIQUETA

Cápsula Teórica

Pregunta:
¿Qué es una lista en Python?

Respuesta:
Es una estructura de datos inmutable que no puede ser modificada una vez creada.

Cápsula práctica

Instrucción:
Convertir una tupla en una lista:

Respuesta:

```
l\n2\n3\ncolores = ("Morado",  
"Purpura", "Azul",  
"Verde")\nlista_colores =  
list(colores)\nprint(type(lista_colores))
```

Cápsula juegos

Juego:
Laberinto

Respuesta:
N/A

SELECTOR CLASE

Cápsula Teórica

Pregunta:

¿Para qué se utilizan las listas anidadas en Python?

Respuesta:

Se utilizan para almacenar y organizar datos de manera jerárquica en una estructura de lista.

Cápsula práctica

Intrucción:

Crea una lista anidada que contenga tres listas, cada una con tres elementos:

Respuesta:

```
lista_anidada = [[1, 2, 3], ["A", "B", "C"], [True, False, True]]\nprint(lista_anidada)
```

Cápsula juegos

Juego:
Copiar código

Respuesta:

```
# Atributos\nperro.nombre
```

```
# Método\nperro.correr()
```

SELECTOR ID

Cápsula Teórica

Pregunta:
¿Qué son los decoradores en Python?

Respuesta:
Son una característica avanzada en Python que permite modificar o extender la funcionalidad.

Cápsula práctica

Instrucción:
Define una clase llamada "Persona" que tenga los siguientes atributos: nombre, edad y género. Define un método llamado "saludar" que imprima un mensaje de saludo personalizado con el nombre de la persona.

Respuesta:

```
1\n2\n3\n4\n5\nclass Persona:\n    def __init__(self, nombre):\n        self.nombre = nombre\n    def saludar(self):\n        print(f'Hola, {self.nombre}')
```

Cápsula juegos

Juego:
Rompecabezas (Premium 4)

Respuesta:
N/A

BORDES

Cápsula Teórica

Pregunta:

¿Qué son los atributos y métodos en Python?

Respuesta:

Los atributos son variables asociadas a un objeto y los métodos son funciones definidas en una clase que pueden ser invocadas en los objetos de esa clase.

Cápsula práctica

Instrucción:

Crea una función llamada saludar que tome un argumento nombre y devuelva un saludo con el nombre.

Ejemplo: "¡Hola,
[nombre]!"

Respuesta:

```
1n2n3n4ndef saludar(nombre):\n
print(";Hola " + nombre + "!")\n return
0\nsaludar("Jaime")
```

Cápsula juegos

Juego:

Preguntas ágiles
(Premium 5) (Preguntas
aleatorias)

Respuesta:

P: Los _____ son una característica...

R: decoradores

P: Prácticamente, un decorador es una función...

R: ¡Verdadero!

P: Los decoradores permiten ejecutar código...

R: ¡Verdadero!

P: Los decoradores pueden ayudarnos...

R: ¡Verdadero!

P: ¿Tienes hambre?

R: ¡Tengo hambre!

SOMBRAS

Cápsula Teórica

Pregunta:

¿Qué son las metaclasses en Python?

Respuesta:

Es una clase especial que define el comportamiento de otras clases.

Cápsula práctica

Intrucción:

1. Crea una clase llamada MiClase y especifica MiMetaclase como su metaclass utilizando la sintaxis de class
MiClase(metaclass=MiMetaclase):
2. En la definición de la clase MiMetaclase, agrega un método llamado nuevo_metodo que

Respuesta:

```
1\n2\n3\n4\n5\n6\n7\n8\n9\nclass\n    MiMetaclase(type):\n    def\n        nuevo_metodo(self):\n            print("¡Hola,\n                soy un nuevo método!")\n\nclass\n    MiClase(metaclass=MiMetaclase):\n        pass\n\nobjeto =\n    MiClase()\nobjeto.nuevo_metodo()
```

Cápsula juegos

Juego:
Crucigrama
(Premium 6)

Respuesta:

Horizontales:

1. Metaclasses
2. Objeto
3. Variable

Verticales:

1. Python
2. Clases

RESPUESTAS CÁPSULA EVALUATIVA PYTHON

Pregunta 1:

¿Cuál es la forma más eficiente de aplicar estilos CSS en un documento HTML?

Respuesta:

Hojas de estilo externas

Pregunta 2:

¿Cuáles de las siguientes declaraciones son propiedades o atributos?

Respuesta:

background-color

Pregunta 3:

¿Cuáles de las siguientes declaraciones son selectores?

Respuesta:

p

Pregunta 4:

¿Cómo se define un color de fondo para todas las etiquetas "p" en CSS?

Respuesta:

p {background-color:red;}

Pregunta 5:

¿Con qué propiedad se define el espacio entre el borde de un elemento y los elementos que lo rodean?

Respuesta:

margin

Pregunta 6:

¿Cómo se cambia el valor del margen derecho de un elemento?

Respuesta:

margin-right

Pregunta 7:

¿Qué significa CSS?

Respuesta:

Cascading Style Sheets

Pregunta 8:

¿Cómo se indica que el color del texto es de color rojo?

Respuesta:

Todas las respuestas son correctas

Pregunta 9:

¿Cómo se indica que todos los elementos "span" tienen el texto en negrita?

Respuesta:

span {font-weight:bold}

Pregunta 10:

¿Qué propiedad se emplea para cambiar el tipo de letra de un elemento?

Respuesta:

font-family