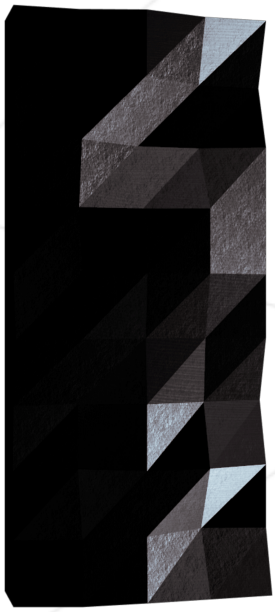

Discovery Piscine

Módulo7 - Python

Resumen: En este Módulo7, vemos cómo usar arrays y sus funciones asociadas.

Versión: 1.00

Índice general

Capítulo I

Unas palabras sobre esta Discovery Piscine

¡Bienvenido!
Comenzarás un módulo de esta Discovery Piscine en programación informática. Nuestro objetivo es introducirte al código que se esconde detrás del software que usas a diario y sumergirte por completo en el aprendizaje entre pares, el modelo educativo de 42.

Programar trata sobre lógica, no sobre matemáticas. Te brinda bloques de construcción básicos que puedes combinar de innumerables formas. No existe una única solución "correcta" para un problema; tu solución será única, al igual que la de cada uno de tus compañeros.

Rápida o lenta, elegante o desordenada, mientras funcione, ¡eso es lo que importa! Estos bloques de construcción formarán una secuencia de instrucciones (para cálculos, visualizaciones, etc.) que el ordenador ejecutará en el orden que diseñes.

En lugar de ofrecerte un curso donde cada problema tiene una única solución, te colocamos en un entorno de aprendizaje entre pares. Buscarás elementos que te ayuden a afrontar el desafío, los refinarás mediante pruebas y experimentación, y finalmente crearás tu propio programa. Habla con otros, comparte tus perspectivas, genera nuevas ideas en conjunto y prueba todo por ti mismo para asegurarte de que funcione.

La evaluación entre pares es una gran oportunidad para descubrir enfoques alternativos y detectar posibles problemas en tu programa que podrías haber pasado por alto (piensa en lo frustrante que puede ser que un programa se bloquee). Cada revisor analizará tu trabajo de manera diferente, como clientes con expectativas variadas, brindándote nuevas perspectivas. Incluso podrías establecer conexiones para futuras colaboraciones.

Al final de esta Piscine, tu recorrido será único. Habrás enfrentado distintos desafíos, validado diferentes proyectos y elegido caminos distintos a los de los demás, ¡y eso está perfectamente bien! Esta es una experiencia tanto colectiva como individual, y todos sacarán algo valioso de ella.

Buena suerte a todos; esperamos que disfruten este viaje de descubrimiento.

Capítulo II

Introducción

Lo que aprenderás en este módulo:

- Aprenderás a usar arrays y sus funciones asociadas.

Capítulo III

Instrucciones generales

A menos que se indique lo contrario, las siguientes reglas se aplican todos los días de esta Piscine.

- Este documento es la única fuente confiable. No te fíes de rumores.
- Este documento puede actualizarse hasta una hora antes del plazo de entrega.
- Las tareas deben completarse en el orden especificado. No se evaluarán tareas posteriores si las anteriores no están correctamente finalizadas.
- Presta mucha atención a los permisos de acceso de tus archivos y carpetas.
- Tus tareas serán evaluadas por tus compañeros de la Piscine.
- Todos los ejercicios de terminal deben ejecutarse con `/bin/bash`.
- No debes dejar ningún archivo en tu espacio de entrega salvo aquellos explícitamente solicitados en las instrucciones.
- ¿Tienes una duda? Pregunta a tu compañero de la izquierda. Si no, prueba con el de la derecha.
- Toda respuesta técnica que necesites se encuentra en las páginas `man` o en línea.
- Recuerda usar el foro de la Piscine en tu intranet y Slack!
- Lee los ejemplos detenidamente, pueden contener requisitos que no sean evidentes en la descripción del ejercicio.
- ¡Por Thor, por Odín! ¡Usa tu cerebro!

Capítulo IV

Ejercicio 00: parameter_matching

	Ejercicio: 00
Identificar un Parámetro	
Directorio de entrega: *ex*00/	
Archivos a entregar: `parameter_matching.py`	
Funciones autorizadas: `All`	

- Crea un programa llamado `parameter_matching.py`.
- Asegúrate de que este programa sea ejecutable.
- El programa debe:
  - Tomar un string como parámetro.
  - Pedir al usuario que ingrese una palabra que coincida con el parámetro pasado como argumento, como se muestra en el ejemplo a continuación.
  - Mostrar `"Good job!"` seguido de un salto de línea si la palabra ingresada por el usuario coincide con el parámetro pasado.
  - Mostrar `"Nope, sorry..."` seguido de un salto de línea si la palabra ingresada por el usuario no coincide con el parámetro pasado.
  - Mostrar `"none"` seguido de un salto de línea si el número de parámetros pasados es diferente de 1.

```
?> ./parameter_matching.py
none
?> ./parameter_matching.py "Hello"
What was the parameter? Bonjour
Nope, sorry...
?> ./parameter_matching.py "Hello"
What was the parameter? Hello
Good job!
?>
```

Usa una o más condicionales if-else.

Capítulo V

Ejercicio 01: count_it

	Ejercicio: 01
Contar y Medir Parámetros	
Directorio de entrega: *ex*01/	
Archivos a entregar: `count_it.py`	
Funciones autorizadas: `All`	

- Crea un programa llamado `count_it.py`.
- Asegúrate de que este programa sea ejecutable.
- El programa debe:
  - Mostrar `"parameters:"` seguido del número total de parámetros pasados como argumentos.
  - Para cada parámetro, mostrar el parámetro en sí y su longitud, terminando con un salto de línea.
  - Si no se proporcionan parámetros, muestra `"none"` seguido de un salto de línea.

```
?> ./count_it.py | cat -e
none$
?> ./count_it.py "Game" "of" "Thrones" | cat -e
parameters: 3$
Game: 4$
of: 2$
Thrones: 7$
?>
```

Esta vez, debes usar un bucle ¨for¨ en lugar de un bucle ¨while¨.

Capítulo VI

Ejercicio 02: string_are_arrays

	Ejercicio: 02
Entendiendo las Strings como Arrays	
Directorio de entrega: *ex*02/	
Archivos a entregar: `string_are_arrays.py`	
Funciones autorizadas: `All`	

- Crea un programa llamado `string_are_arrays.py` que reciba una string como parámetro.
- Este programa debe ser ejecutable.
- Al ejecutarlo, el programa debe mostrar `"z"` por cada ocurrencia del carácter `"z"` en la string pasada como parámetro, seguida de un salto de línea.
- Si el número de parámetros es diferente de 1, o si no hay caracteres `"z"` en el string, debe mostrar `"none"` seguido de un salto de línea.

```
?> ./string_are_arrays.py | cat -e
none$
?> ./string_are_arrays.py "The character Z is not found in this string" | cat -e
none$
?> ./string_are_arrays.py "The character z is found in this string" | cat -e
z$
?> ./string_are_arrays.py "Zaz visits the zoo with Zazie" | cat -e
zzz$
?>
```

Los strings también están compuestos por caracteres individuales, al igual que los arrays. ¡Inténtalo!

Capítulo VII

Ejercicio 03: append_it

	Ejercicio: 03
Modificando Strings	
Directorio de entrega: *ex*03/	
Archivos a entregar: `append_it.py`	
Funciones autorizadas: `All`	

- Crea un programa llamado `append_it.py`.
- Asegúrate de que este programa sea ejecutable.
- El programa debe mostrar cada parámetro pasado como argumento, uno por uno, añadiéndole «ism».
- Si un parámetro ya termina con «ism», debe ser omitido y no mostrado.
- Si no se proporcionan parámetros, el programa debe mostrar `"none"` seguido de un salto de línea.

```
?> ./append_it.py | cat -e
none$
?> ./append_it.py "parallel" "egoism" "human" | cat -e
parallelism$
humanism$
?>
```

`Usa matching.`

Capítulo VIII

Ejercicio 04: free_range

	Ejercicio: 04
¡A un paso más!	
Directorio de entrega: *ex*04/	
Archivos a entregar: `free_range.py`	
Funciones autorizadas: `All`	

- Crea un programa llamado `free_range.py`.
- Asegúrate de que este programa sea ejecutable.
- Este programa debe recibir 2 números como parámetros.
- Debes construir un array que contenga todos los valores entre estos dos números.
- Debes mostrar el array usando la función `print`.
- Si el número de parámetros es diferente de 2, el programa debe mostrar `"none"` seguido de un salto de línea.

```
?> ./free_range.py | cat -e
none$
?> ./free_range.py 10 14 | cat -e
[10, 11, 12, 13, 14]$
?>
```

`Usa la función range.`

Capítulo IX

Entrega y Evaluación entre Pares

- Debes tener una carpeta llamada `discovery_piscine` en la raíz de tu directorio personal.
- Dentro de `discovery_piscine`, debe existir una carpeta llamada `module7`.
- Dentro de `module7`, debe haber una carpeta para cada ejercicio.
- El Ejercicio 00 debe estar en la carpeta `ex00`, el Ejercicio 01 en `ex01`, y así sucesivamente.
- Cada carpeta de ejercicio debe contener los archivos solicitados en el enunciado.

`Durante tu defensa, todo lo que no esté en la carpeta correspondiente para el día no será revisado.`