



Discovery Piscine

Módulo3 - Python

Resumen: En este Módulo3 veremos cómo usar bucles.

Versión: 1.00

Índice general

I.	Unas palabras sobre esta Discovery Piscine	2
II.	Introducción	3
III.	Instrucciones generales	4
IV.	Ejercicio 00: Hasta el 25	5
V.	Ejercicio 01: multiplication_table	7
VI.	Ejercicio 02: i_got_that	8
VII.	Exercise 03: advanced_mult	9
VIII.	Entrega y Evaluación entre Pares	10

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 hacer algunos bucles.

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: Hasta el 25

	Ejercicio: 00
¡Vamos hasta el 25!	
Directorio de entrega: <i>ex00/</i>	
Archivos a entregar: to25.py	
Funciones autorizadas: All	

- Crea un programa llamado `to25.py`.
- Asegúrate de que el programa sea ejecutable.
- El programa debe:
 - Aceptar una entrada del usuario, que se almacenará en una variable numérica.
 - Usar un bucle para mostrar todos los números desde el número ingresado hasta el 25.
 - Si el número ingresado es mayor que 25, mostrar «Error» seguido de un salto de línea.


```
?> ./to25.py
Enter a number less than 25
45
Error
?> ./to25.py
Enter a number less than 25
20
Inside the loop, my variable is 20
Inside the loop, my variable is 21
Inside the loop, my variable is 22
Inside the loop, my variable is 23
Inside the loop, my variable is 24
Inside the loop, my variable is 25
?>
```



Usa un bucle `while`.

Capítulo V

Ejercicio 01: multiplication_table


	Ejercicio: 01
El regreso de las tablas de multiplicar	
Directorio de entrega: <i>ex01/</i>	
Archivos a entregar: <code>multiplication_table.py</code>	
Funciones autorizadas: All	

- Crea un programa llamado `multiplication_table.py`.
- Asegúrate de que el programa sea ejecutable.
- El programa debe:
 - Aceptar una entrada del usuario, que se almacenará en una variable numérica.
 - Mostrar la tabla de multiplicar de ese número (por ejemplo, si la entrada es 2, mostrar la tabla de multiplicar del 2).

```
?> ./multiplication_table.py
Enter a number
8
0 x 8 = 0
1 x 8 = 8
2 x 8 = 16
3 x 8 = 24
4 x 8 = 32
5 x 8 = 40
6 x 8 = 48
7 x 8 = 56
8 x 8 = 64
9 x 8 = 72
?>
```


Capítulo VI

Ejercicio 02: i_got_that

	Ejercicio: 02
¿Lo tienes?	
Directorio de entrega: <i>ex02/</i>	
Archivos a entregar: <i>i_got_that.py</i>	
Funciones autorizadas: All	

- Crea un programa llamado `i_got_that.py`.
- Asegúrate de que el programa sea ejecutable.
- El programa debe:
 - Utilizar un bucle `while` que acepte continuamente la entrada del usuario y responda con «I got that! Anything else?» despues de cada entrada.
 - El bucle solo debe detenerse cuando el usuario ingrese "STOP".


```
?> ./i_got_that.py
What you gotta say? : Hello
I got that! Anything else? : I like ponies
I got that! Anything else? : stop...
I got that! Anything else? : STOP
?>
```



While, break.

Capítulo VII

Exercise 03: advanced_mult

	Ejercicio: 03
El Regreso del Regreso de las Tablas de Multiplicar	
Directorio de entrega: <i>ex03/</i>	
Archivos a entregar: advanced_mult.py	
Funciones autorizadas: All	

- Crea un programa llamado `advanced_mult.py`.
- Asegúrate de que el programa sea ejecutable.
- El programa debe:
 - Muestra todas las tablas de multiplicar del 0 al 10 en el siguiente formato:

```
?> ./advanced_mult.py
Table of 0: 0 0 0 0 0 0 0 0 0 0 0
Table of 1: 0 1 2 3 4 5 6 7 8 9 10
Table of 2: 0 2 4 6 8 10 12 14 16 18 20
Table of 3: 0 3 6 9 12 15 18 21 24 27 30
Table of 4: 0 4 8 12 16 20 24 28 32 36 40
Table of 5: 0 5 10 15 20 25 30 35 40 45 50
Table of 6: 0 6 12 18 24 30 36 42 48 54 60
Table of 7: 0 7 14 21 28 35 42 49 56 63 70
Table of 8: 0 8 16 24 32 40 48 56 64 72 80
Table of 9: 0 9 18 27 36 45 54 63 72 81 90
Table of 10: 0 10 20 30 40 50 60 70 80 90 100
?>
```

- Solo se te permite usar dos bucles while.

Capítulo VIII

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 `module3`.
- Dentro de `module3`, 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.