



# Discovery Piscine

## Módulo2 - Python

*Resumen: En este Módulo2 veremos cómo usar condiciones.*

*Versión: 2.00*

# Índice general

<b>I.</b>	<b>Unas palabras sobre esta Discovery Piscine</b>	<b>2</b>
<b>II.</b>	<b>Introducción</b>	<b>3</b>
<b>III.</b>	<b>Instrucciones generales</b>	<b>4</b>
<b>IV.</b>	<b>Ejercicio 00: ¿Soy Null?</b>	<b>5</b>
<b>V.</b>	<b>Ejercicio 01: ¿Soy Negativo?</b>	<b>6</b>
<b>VI.</b>	<b>Ejercicio 02: Contraseñas</b>	<b>7</b>
<b>VII.</b>	<b>Ejercicio 03: Multiplicación</b>	<b>8</b>
<b>VIII.</b>	<b>Entrega y Evaluación entre Pares</b>	<b>10</b>

# 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 cómo usar condiciones.

# 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: ¿Soy Null?

	Ejercicio: 00
¿Soy Null?	
Directorio de entrega: <i>ex00/</i>	
Archivos a entregar: <b>iszero.py</b>	
Funciones autorizadas: All	

- Crea un programa llamado `iszero.py`.
- Asegúrate de que este programa sea ejecutable (presta atención a los permisos del archivo).
- Al ejecutarlo, el programa debe pedir al usuario que ingrese un número.
- Si el número es igual a cero, el programa debe mostrar: "This number is equal to zero."
- Si el número no es igual a cero, el programa debe mostrar: "This number is different from zero."


```
?> ./iszero.py
42
This number is different from zero.
?> ./iszero.py
0
This number is equal to zero.
?>
```



Hay una pequeña línea que debes agregar al principio de tu archivo para que el shell sepa cómo interpretar el script. Busca "Hacer un script de Python ejecutable".

# Capítulo V

## Ejercicio 01: ¿Soy Negativo?


	Ejercicio: 01
¿Soy Negativo?	
Directorio de entrega: <i>ex01/</i>	
Archivos a entregar: <b>isneg.py</b>	
Funciones autorizadas: All	

- Crea un programa llamado `isneg.py`.
- Asegúrate de que este programa sea ejecutable.
- Al ejecutarlo, el programa debe pedir al usuario que ingrese un número.
- Si el número es negativo, el programa debe mostrar: "This number is negative."
- Si el número es positivo, el programa debe mostrar: "This number is positive."
- Si el número es igual a cero, el programa debe mostrar: "This number is both positive and negative."

```
?> ./isneg.py
42
This number is positive.
?> ./isneg.py
-42
This number is negative.
?> ./isneg.py
0
This number is both positive and negative.
?>
```

# Capítulo VI

## Ejercicio 02: Contraseñas

	Ejercicio: 02
Acceso denegado	
Directorio de entrega: <i>ex02/</i>	
Archivos a entregar: <b>password.py</b>	
Funciones autorizadas: All	

- Crea un programa llamado `password.py`.
- Asegúrate de que este programa sea ejecutable.
- Define una variable que contenga una contraseña.

```
password = "Python is awesome"
```


- Al ejecutarlo, el programa debe pedir al usuario que ingrese una contraseña.
- Si la contraseña es correcta, el programa debe mostrar: “ACCESS GRANTED”.
- Si la contraseña es incorrecta, el programa debe mostrar: “ACCESS DENIED”.

```
?> ./password.py
1234
ACCESS DENIED
?> ./password.py
Python is awesome
ACCESS GRANTED
?>
```



# Capítulo VII

## Ejercicio 03: Multiplicación

	Ejercicio: 03
Multiplicación	
Directorio de entrega: <i>ex03/</i>	
Archivos a entregar: <i>mult.py</i>	
Funciones autorizadas: <i>All</i>	

- Crea un programa llamado `mult.py`.
- Asegúrate de que este programa sea ejecutable.
- Al ejecutarlo, el programa debe pedir al usuario que ingrese 2 números.
- El programa debe:
  - Mostrar si el resultado de multiplicar los dos números es positivo, negativo o cero.
  - Mostrar el resultado de la multiplicación.

```
$ ./mult.py
Enter the first number:
42
Enter the second number:
42
42 x 42 = 1764
The result is positive.
```

```
$ ./mult.py
Enter the first number:
78
Enter the second number:
-1
78 x -1 = -78
The result is negative.
```

```
$ ./mult.py
Enter the first number:
72
Enter the second number:
0
```

```
72 x 0 = 0  
The result is positive and negative.
```

# 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 `module2`.
- Dentro de `module2`, 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.