



Discovery Piscine

Módulo6 - Python

Resumen: En este Módulo6, vemos cómo usar arrays y sus funciones asociadas.

Versión: 2.1

Índice general

| | | |
|-------|--|----|
| I. | Unas palabras sobre esta Discovery Piscine | 2 |
| II. | Introducción | 3 |
| III. | Instrucciones generales | 4 |
| IV. | Ejercicio 00: Mostrar el Primer Parámetro | 5 |
| V. | Ejercicio 01: <code>UPCASE_IT</code> | 6 |
| VI. | Ejercicio 02: <code>downcase_it</code> | 7 |
| VII. | Ejercicio 03: <code>aff_rev_params</code> | 8 |
| VIII. | Ejercicio 04: <code>scan_it</code> | 9 |
| IX. | 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 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: Mostrar el Primer Parámetro


| | |
|---|---------------|
|  | Ejercicio: 00 |
| Mostrar un parámetro | |
| Directorio de entrega: <i>ex00/</i> | |
| Archivos a entregar: aff_first_param.py | |
| Funciones autorizadas: All | |

- Crea un programa llamado **aff_first_param.py**.
- Asegúrate de que este programa sea ejecutable.
- El programa debe mostrar el primer parámetro de tipo string pasado, seguido de un salto de línea.
- Si no hay parámetros, muestra **"none"** seguido de un salto de línea.

```
?> ./aff_first_param.py | cat -e
none$
?> ./aff_first_param.py "Code Ninja" "Numeric" "42" | cat -e
Code Ninja$
?>
```

Capítulo V

Ejercicio 01: UPCASE_IT


| | |
|---|---------------|
|  | Ejercicio: 01 |
| ¡Hazlo Grande! | |
| Directorio de entrega: <i>ex01/</i> | |
| Archivos a entregar: upcase_it.py | |
| Funciones autorizadas: All | |

- Crea un programa llamado `upcase_it.py` que reciba un string como parámetro.
- Asegúrate de que este programa sea ejecutable.
- El programa debe mostrar el string en mayúsculas, seguido de un salto de línea.
- Si el número de parámetros es diferente de 1, muestra "**none**" seguido de un salto de línea.

```
?> ./upcase_it.py | cat -e
none$
?> ./upcase_it.py "initiation" | cat -e
INITIATION$
?> ./upcase_it.py 'This exercise is quite easy!' | cat -e
THIS EXERCISE IS QUITE EASY!$
?>
```

Capítulo VI

Ejercicio 02: lowercase_it

| | |
|---|---------------|
|  | Ejercicio: 02 |
| Hazlo Pequeño | |
| Directorio de entrega: <i>ex02/</i> | |
| Archivos a entregar: lowercase_it.py | |
| Funciones autorizadas: All | |

- Crea un programa llamado `lowercase_it.py`.
- Asegúrate de que este programa sea ejecutable.
- El programa debe recibir un string como parámetro.
- Debe mostrar el string en minúsculas, seguido de un salto de línea.
- Si el número de parámetros es diferente de 1, debe mostrar **"none"** seguido de un salto de línea.


```
?> ./lowercase_it.py | cat -e
none$
?> ./lowercase_it.py "FIREFLY" | cat -e
firefly$
?> ./lowercase_it.py 'This exercise is quite easy!' | cat -e
this exercise is quite easy!$
?>
```



Este ejercicio no debería tomarte más de 10 segundos.

Capítulo VII

Ejercicio 03: aff_rev_params


| | |
|---|---------------|
|  | Ejercicio: 03 |
| Mostrar un Array en Orden Inverso | |
| Directorio de entrega: <i>ex03/</i> | |
| Archivos a entregar: aff_rev_params.py | |
| Funciones autorizadas: All | |

- Crea un programa llamado `aff_rev_params.py`.
- Asegúrate de que este programa sea ejecutable.
- Cuando se ejecute, el programa debe mostrar todos los strings pasados como parámetros, seguidos de un salto de línea, en orden invertido.
- Si hay menos de dos parámetros, debe mostrar `"none"` seguido de un salto de línea.

```
?> ./aff_rev_params.py | cat -e
none$
?> ./aff_rev_params.py "coucou" | cat -e
none$
?> ./aff_rev_params.py "Python" "piscine" "hello" | cat -e
hello$
piscine$
Python$
?>
```

Capítulo VIII

Ejercicio 04: scan_it

| | |
|---|---------------|
|  | Ejercicio: 04 |
| Escanear un String | |
| Directorio de entrega: <i>ex04/</i> | |
| Archivos a entregar: scan_it.py | |
| Funciones autorizadas: All | |

- Crea un programa llamado **scan_it.py**.
- Asegúrate de que este programa sea ejecutable.
- Este programa debe tomar dos parámetros.
- El primer parámetro es una palabra clave para buscar en un string.
- El segundo parámetro es el string en el que se va a buscar.
- Al ejecutarse, el programa debe mostrar el número de veces que aparece la palabra clave en el string.
- Si el número de parámetros es diferente de 2 o si el primer string no aparece en el segundo, debe mostrar **"none"** seguido de un salto de línea.

```
?> ./scan_it.py | cat -e
none$
?> ./scan_it.py "the" | cat -e
none$
?> ./scan_it.py "the" "hello world" | cat -e
none$
?> ./scan_it.py "the" "the quick brown fox jumps over the lazy dog" | cat -e
2$
?>
```



Puedes usar el método `re.findall()` del módulo `re` de Python.

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 `module6`.
- Dentro de `module6`, 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.