



Discovery Piscine

Módulo9 - Python

Resumen: En este módulo, veremos cómo manipular diccionarios. ¡Como arrays, pero más cómodo!

Versión: 3.0

Índice general

I.	Unas palabras sobre esta Discovery Piscine	2
II.	Introducción	3
III.	Instrucciones generales	4
IV.	Ejercicio 00: your_namebook	5
V.	Ejercicio 01: family_affairs	7
VI.	Ejercicio 02: help_your_professor	9
VII.	Ejercicio 03: persons_of_interest	10
VIII.	Entrega y Evaluación entre Pares	12

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 cómo manejar 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: your_namebook

	Ejercicio: 00
¡Organiza esta lista de nombres para mí!	
Directorio de entrega: <i>ex00/</i>	
Archivos a entregar: <code>your_namebook.py</code>	
Funciones autorizadas: All	

- Crea un script llamado `your_namebook.py`.
- Este script debe contener un método llamado `array_of_names`.
- Este método debe tomar un diccionario que asocia nombres con apellidos como parámetro.
- Debe construir un array que contenga los nombres completos de las personas, con la primera letra de cada nombre en mayúsculas. El método debe devolver este array. Consulta el ejemplo.
- Por ejemplo, el siguiente script:

```
# your method definition here

persons = {
    "jean": "valjean",
    "grace": "hopper",
    "xavier": "niel",
    "fifi": "brindacier"
}

print(array_of_names(persons))
```

Producirá la siguiente salida:


```
?> ./your_namebook.py
['Jean Valjean', 'Grace Hopper', 'Xavier Niel', 'Fifi Brindacier']
?>
```



Puedes usar el método `capitalize` en Python.

Capítulo V

Ejercicio 01: family_affairs

	Ejercicio: 01
Asuntos familiares	
Directorio de entrega: <i>ex01/</i>	
Archivos a entregar: family_affairs.py	
Funciones autorizadas: All	

- Crea un script llamado **family_affairs.py**.
- Este script debe contener un método llamado **find_the_redheads**.
- Este método tomará un diccionario como parámetro, donde las claves son los nombres de los miembros de la familia y los valores son los colores de su cabello.
- El método debe usar la función **filter** para recopilar los nombres de las personas con cabello rojo en una nueva lista, que debe devolver.
- El resultado debe ser convertido en una lista antes de ser devuelto usando la función **tolist()**.
- Por ejemplo, el siguiente script:

```
# your method definition here
dupont_family = {
    "florian": "red",
    "marie": "blond",
    "virginie": "brunette",
    "david": "red",
    "franck": "red"
}

print(find_the_redheads(dupont_family))
```

Producirá la siguiente salida:


```
?> ./family_affairs.py
['florian', 'david', 'franck']
?>
```




Busca Python dictionary `filter`, `keys()`, `tolist()`

Capítulo VI

Ejercicio 02: help_your_professor

	Ejercicio: 02
Dale una mano a tu profesor	
Directorio de entrega: <i>ex02/</i>	
Archivos a entregar: help_your_professor.py	
Funciones autorizadas: All	

- Crea un script llamado `help_your_professor.py`.
- El script debe contener un método llamado `average`.
- Este método debe tomar un diccionario como parámetro, donde las claves son los nombres de los estudiantes y los valores son sus puntuaciones en una tarea. El método debe calcular el promedio de la clase para esa tarea.
- Por ejemplo, el siguiente script:

```
# your method definition here
class_3B = {
    "marine": 18,
    "jean": 15,
    "coline": 8,
    "luc": 9
}
class_3C = {
    "quentin": 17,
    "julie": 15,
    "marc": 8,
    "stephanie": 13
}


print(f"Average for class 3B: {average(class_3B)}.")
print(f"Average for class 3C: {average(class_3C)}.")
```

Producirá la siguiente salida:

```
?> ./help_your_professor.py
Average for class 3B: 12.5.
Average for class 3C: 13.25.
?>
```

Capítulo VII

Ejercicio 03: persons_of_interest

	Ejercicio: 03
Personas que vale la pena conocer	
Directorio de entrega: <i>ex03/</i>	
Archivos a entregar: persons_of_interest.py	
Funciones autorizadas: All	

- Crea un script llamado `persons_of_interest.py`.
- El script debe contener un método llamado `famous_births`.
- Este método debe tomar un diccionario como parámetro, representando figuras históricas. Cada entrada en el diccionario es, a su vez, un diccionario con las claves: `name` y `date_of_birth`.
- El método debe ordenar el diccionario por fechas de nacimiento y luego mostrar cada entrada (vea el ejemplo a continuación).
- Por ejemplo, el siguiente script:

```
# your method definition here

women_scientists = {
    "ada": { "name": "Ada Lovelace", "date_of_birth": "1815" },
    "cecilia": { "name": "Cecila Payne", "date_of_birth": "1900" },
    "lise": { "name": "Lise Meitner", "date_of_birth": "1878" },
    "grace": { "name": "Grace Hopper", "date_of_birth": "1906" }
}

famous_births(women_scientists)
```

Producirá la siguiente salida:

```
?> ./persons_of_interest.py
Ada Lovelace is a great scientist born in 1815.
Lise Meitner is a great scientist born in 1878.
Cecila Payne is a great scientist born in 1900.
Grace Hopper is a great scientist born in 1906.
?>
```



Google python dictionary y sorted



También puedes buscar los nombres mencionados anteriormente,
¡realmente vale la pena aprender sobre ellos!



¡Python es una herramienta poderosa, pero también puede ser muy
divertido! `print(chr(sum(range(ord(min(str(not()))))))))`

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 `module9`.
- Dentro de `module9`, 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.