






¡Comenzando con Python!

Sergio García Prado

28 de septiembre de 2020

Sobre mi



- ¡Hola! Soy Sergio García Prado
- Estudios:
 -  Graduado en Ingeniería Informática
 -  Graduado en Estadística
- Experiencia Laboral:
 -  Trabajando en Unlimiteck
- Sígueme:
 - Website: garciparedes.me
 - GitHub: [@garciparedes](https://github.com/garciparedes)
 - Email: sergio@garciparedes.me

¿Qué es programar?

“La programación es el proceso utilizado para idear y ordenar las acciones necesarias para realizar un proyecto, preparar ciertas máquinas o aparatos para que empiecen a funcionar en el momento y en la forma deseados o elaborar programas para su empleo en computadoras”

— R.A.E. (2014)



Similitud con recetas de cocina



- La idea de programar sobre un ordenador no es muy diferente de la tareas como la de seguir una receta.
- Receta de Bizcocho:
 - 1 Mezclar ingredientes secos.
 - 2 Mezclar ingredientes húmedos.
 - 3 Combinar todos los ingredientes.
 - 4 Encender el horno.
 - 5 Amasar hasta conseguir una masa homogénea.
 - 6 Hornear durante 30 minutos.
 - 7 Dejar reposar.

Pensamiento jerárquico: desde arriba hasta abajo

- La idea de la programación se basa en la construcción de abstracciones sobre conceptos básicos, que de manera conjunta forman una base de conocimiento más rica.
- Ejemplo:
 - Hacer bizcocho
 - Mezclar ingredientes secos, ...
 - Mezclar harina, azúcar, levadura, ...
 - Introducir harina en el recipiente, introducir azúcar en el recipiente, ...
 - Abrir paquete de harina, calcular cantidad de harina, ...
 - ...

¡Hola, Mundo!

- El término *Hello, World* consiste en la ejemplificación sobre las sentencias necesarias para un lenguaje de programación determinado, de tal manera que este imprima por consola el texto Hello, World!.
- Hola mundo en *Python*: `print("Hello, World!")`



¿Por qué Python es popular?



- *Python* es un lenguaje de programación interpretado cuya filosofía hace hincapié en la *legibilidad de su código*. bu
- En los últimos años ha sufrido un crecimiento exponencial, entre otros, por el ecosistema de tan heterogéneo que se ha ido construyendo a su alrededor: *Data Science*, *Web Development*, *Embedded Systems*, etc.

Caso Práctico: Recordatorios (I)

- Para ejemplificar cómo usar *Python* así como sus instrucciones básicas se propone construir una aplicación básica de recordatorios.
- Por motivos de simplicidad, por el momento la aplicación permitirá únicamente las siguientes historias de usuario:
 - Añadir nuevos recordatorios.
 - Listar recordatorios existentes.



Caso Práctico: Recordatorios (II)

This is a text in the first frame.

```
1 message = input("What to reminder? --> ")
2 seconds = int(input("How many seconds? --> "))
3 reminder = [False, seconds, message]
4 print(f"Created a reminder: {reminder}")
```

Caso Práctico: Recordatorios (II)

This is a text in the first frame.

```
1 reminders = list()  
2 reminders.append(reminder)
```

Caso Práctico: Recordatorios (II)

This is a text in the first frame.

```
1 print("Reminders:")
2 print("\tCompleted\tRemaining\tMessage")
3 for reminder in reminders:
4     print(f"\t{reminder[0]:>8}\t{reminder[1]:>9}\t{reminder[2]}")
```

Caso Práctico: Recordatorios (II)

This is a text in the first frame.

```
1 while True:
2     command = input()
3     if command == "q":
4         break
5     elif command == "a":
6         ...
7     elif command == "s":
8         ...
```

Python Sample

This is a text in the first frame.

```
1  reminders = list()
2  while True:
3      command = input()
4      for reminder in reminders:
5          if reminder[1] - time() >= 0:
6              continue
7          reminder[0] = True
8      reminders.sort()
9      if command == "q":
10         break
11     elif command == "a":
12         message = input("What to reminder? --> ")
13         seconds = round(time()) + int(input("How many seconds? --> "))
14         reminder = [False, seconds, message]
15         reminders.append(reminder)
16         print(f"Created a reminder: {reminder}")
17         print()
18     elif command == "s":
19         print("Remainders:")
20         print("\tCompleted\tRemaining\tMessage")
21         for reminder in reminders:
22             print(f"\t{reminder[0]}\t\t\t{reminder[1] - round(time()):>9}\t\t{reminder[2]}")
23         print()
24
25     sleep(0.5)
```

¿Preguntas?