



ALGORITMOS Y ESTRUCTURAS DE DATOS (TSDS)

ASIGNATURA:

ALGORITMOS Y ESTRUCTURAS DE DATOS

PROFESOR:

Ing. Lorena Chulde MSc.

PERÍODO ACADÉMICO:

2023-B

TAREA 7 Grupal

TÍTULO:

FUNCIONES Y CORRECCIÓN DEL EXAMEN

Nombre de los estudiantes:

Guerra Lovato Josué Eduard

Pérez Orosco Carlos David

Soria Asanza Richard Mauricio

PYTHON
funciones:
parámetros y
argumentos

2023-B

1. Realice una consulta sobre las funciones en Python.

FUNCIONES

DEFINICION

Una función en Python, así como en cualquier otro lenguaje de programación, se define como un bloque de líneas de código o un conjunto de instrucciones diseñado para llevar a cabo una tarea específica. Su principal ventaja radica en la capacidad de reutilizar este bloque de código según sea necesario para repetir la tarea deseada. La creación y comprensión de funciones contribuyen significativamente a hacer el código más legible y comprensible. Este conocimiento también refleja la habilidad de abstracción de un programador, una destreza técnica fundamental y desafiante en el desarrollo de software. La siguiente guía proporciona una introducción básica sobre cómo definir funciones, marcando el inicio de la adquisición de estas habilidades en programación [1].

COMO DECLARAR UNA FUNCION

Las funciones en Python desempeñan un papel crucial al hacer que el código sea más comprensible y fácil de leer. La capacidad de un programador para generar funciones también refleja su habilidad en la abstracción, una destreza técnica desafiante en el desarrollo de software. Para comenzar en este camino, se proporciona una guía básica sobre cómo definir funciones [1].

Para definir una función en Python, se utiliza la palabra reservada "def", seguida del nombre deseado para la función. Luego, se colocan paréntesis con los parámetros de entrada, seguidos de dos puntos. A continuación, se agrega un bloque de instrucciones que se ejecutará cada vez que se invoque la función.

Dentro de este bloque, opcionalmente, se coloca una sentencia de retorno utilizando la palabra reservada "return", indicando el valor que la función debería devolver. Es importante destacar que una función en Python siempre devolverá un valor, predeterminadamente "None" si no se especifica un retorno.

Se ilustra un ejemplo de definición de función que suma dos valores:

```
def suma(a, b):  
    return a + b
```

```
result = suma(8, 5)  
# result = 13
```

Además, es posible definir valores predeterminados para los parámetros de entrada, permitiendo que estos sean opcionales al invocar la función. Si no se proporciona un valor, se utilizarán los predeterminados, como se muestra en el siguiente ejemplo:

```
def suma(a, b=8):  
    return a + b
```

```
result = suma(5)  
# result = 13
```

Esta capacidad de definir valores predeterminados mejora la flexibilidad y usabilidad de las funciones en Python [1].

CONSIDERACIONES ADICIONALES ACERCA DE LAS FUNCIONES EN PYTHON

- Al crear y emplear funciones personalizadas en Python, es fundamental considerar varias pautas:
- Garantizar el retorno de un valor: Es esencial que todas las funciones devuelvan un valor como respuesta, incluso si se trata del valor predeterminado "None".
- Establecer valores predeterminados para parámetros: Se recomienda definir valores por defecto para todos los parámetros de entrada siempre que sea posible, mejorando así la flexibilidad de la función.
- Verificar la correspondencia de parámetros: Al invocar una función, es crucial asegurarse de que la cantidad de parámetros utilizados coincida con la cantidad definida durante la creación de la función.
- Considerar la indentación: Dado que Python utiliza la indentación para delimitar bloques de código, es necesario verificar que las líneas destinadas a la función estén correctamente indentadas.
- Limitar el uso de variables globales: Se aconseja utilizar exclusivamente variables locales dentro de las funciones y evitar el uso de variables globales para prevenir comportamientos inesperados en el código.
- Atender a errores durante la interpretación: Dado que Python es un lenguaje interpretado, es posible que los errores en la definición de funciones no sean señalados hasta el momento en que la función es invocada, lo que resalta la importancia de revisar la sintaxis con precaución [2].

2. Realice la corrección del examen bimestral

Como tal las correcciones de cada uno las realizamos y subimos en un enlace de GitHub donde se encuentran los códigos escritos.

- Corrección de Guerra Josue, enlace GitHub:

https://github.com/JosueGuerra2023B/EstructurasDatos2023B/tree/master/Correcci%C3%B3n_del_examen_GuerraJosue

- Corrección de Pérez Carlos, enlace GitHub:

https://github.com/CarlosPerez19/Algoritmos-y-Estructuras-de-datos--Carlos-Perez/tree/1463b171afe2905c6d2303564a1cd6907dbe346b/Programa%20Examen%20Bimestral_Correccion

- Corrección de Soria Richard, enlace GitHub:

<https://github.com/RichardSoria/Estructura-Datos-2023-B/tree/28caef0b24b339fb63f69cb448ffce1c35145884/CORRECCION%20EXAMEN%20BIMESTRAL%20RICHARD%20SORIA>

ENLACE GITHUB:

<https://github.com/JosueGuerra2023B/Estructuras-Datos2023B/tree/master/Deberes/Deber%206>

ENLACE DIAPOSITIVAS:

https://www.canva.com/design/DAF5n4vCChs/X3XiOlwT4zFTSQrug7sx2Q/view?utm_content=DAF5n4vCChs&utm_campaign=designshare&utm_medium=link&utm_source=editor

- **DENTRO DEL ENLACE DE GITHUB SE ENCUENTRAN LAS DIAPOSITIVAS, LOS EJERCICIOS DE EJEMPLO, Y LOS DOCUMENTOS TANTO EN WORD COMO EN PDF.**

ENTREGABLES:

- Una vez culminada tu tarea, captura las pantallas de la ejecución del problema con tus datos y súbela en el apartado del aula virtual "S6-Tarea-6:
- Sube los ejercicios al git o al drive y entrega la url de los archivos .py
- Recuerda el nombre del archivo deberá ser: **Tarea7_Algoritmos_2023B_NApellido.**

RECURSOS NECESARIOS

- Acceso a Internet.
- Imaginación.
- VSC

1 Bibliografía

[1] hubspot, «hubspot.es,» 08 10 2022. [En línea]. Available: <https://blog.hubspot.es/website/funciones-python>. [Último acceso: 10 01 2024].

[2] C. MEDINA Salgado, junio 2012. [En línea]. Available: <http://goo.gl/0fuQU>.