

Tarea 18 - Errores y excepciones en Python

Curso de Python

Ejercicio 1

Dada la función `divisors()`, asegura con `assert` que el parámetro `n` se trata de un número entero positivo. De lo contrario, muestra como mensaje que "n debe ser de tipo int y debe ser mayor que 0"

```
def divisors(n):  
    """  
    Calcula los divisores de un número entero positivo.  
    Args:  
        n: Número entero positivo  
    Returns:  
        divisors: Lista de divisores de n  
    """  
    divisors = []  
    for i in range(1, n + 1):  
        if n % i == 0: divisors.append(i)  
    return divisors
```

Ejercicio 2

Dada la función `divisors()`, muestra `TypeError` con el mensaje correspondiente si el parámetro `n` no se trata de un número entero y muestra `ValueError` con el mensaje correspondiente si `n` no se trata de un número positivo.

```
def divisors(n):  
    """  
    Calcula los divisores de un número entero positivo.  
    Args:  
        n: Número entero positivo  
    Returns:  
        divisors: Lista de divisores de n  
    """  
    divisors = []  
    for i in range(1, n + 1):  
        if n % i == 0: divisors.append(i)  
    return divisors
```

Ejercicio 3

Dada la función `is_palindrome()`, asegura con `assert` que el parámetro `word` se trata de una variable de tipo string. De lo contrario, muestra como mensaje que "word debe ser de tipo string".

```
def is_palindrome(word):
    """
    Devuelve si la palabra word es palíndroma.
    Args:
        word: Palabra
    Returns:
        isPalindrome: Booleano
    """
    word = word.lower()
    l = []
    isPalindrome = True
    for c in word:
        l.append(c)
    n = len(l)
    for i in range(int(n / 2)):
        if l[i] != l[n - (i + 1)]: isPalindrome = False
    return isPalindrome
```

Ejercicio 4

Dada la función `is_palindrome()`, muestra `TypeError` con el mensaje correspondiente si `word` no se trata de una variable de tipo string.

```
def is_palindrome(word):
    """
    Devuelve si la palabra word es palíndroma.
    Args:
        word: Palabra
    Returns:
        isPalindrome: Booleano
    """
    word = word.lower()
    l = []
    isPalindrome = True
    for c in word:
        l.append(c)
    n = len(l)
    for i in range(int(n / 2)):
        if l[i] != l[n - (i + 1)]: isPalindrome = False
    return isPalindrome
```

Ejercicio 5

Dada la función `is_palindrome()`, muestra `ValueError` con el mensaje correspondiente si `word` no se trata de una palabra. Es decir, si una vez ha pasado la comprobación de ser una variable de tipo string, comprueba que no tiene espacios.

```
def is_palindrome(word):
    """
    Devuelve si la palabra word es palíndroma.
    Args:
```

```

    word: Palabra
Returns:
    isPalindrome: Booleano
"""
word = word.lower()
l = []
isPalindrome = True
for c in word:
    l.append(c)
n = len(l)
for i in range(int(n / 2)):
    if l[i] != l[n - (i + 1)]: isPalindrome = False
return isPalindrome

```

Ejercicio 6

Dada la función `is_palindrome()` del resultado del ejercicio anterior, modifícala de modo que acepte no solo palabras sino también frases y devuelva si el string introducido es palíndromo o no. Recuerda tener en cuenta la excepción `TypeError` junto al mensaje correspondiente.

Ejercicio 7

Utiliza `try / except` para dado un objeto de Python guardado en la variable `x`, aplicarle el método `.index()` y localizar el elemento "c". En el bloque `try`, guarda el resultado en la variable `result`. En el bloque `except`, en la variable `result`, si el objeto `x` es una lista, guarda el mensaje "La lista no tiene el elemento "c". En caso contrario, guarda el mensaje "El objeto {tipo del objeto que sea x} no tiene el método `.index()`". Finalmente, imprime por pantalla el resultado.

Ejercicio 8

Crea una función que calcule el área de un cuadrado. Como parámetro se recibirá un número real, correspondiente a la longitud de la base. Lanza las excepciones pertinentes siempre que el parámetro no se trate de un número real o entero positivo.

Ejercicio 9

Crea una función que solicite al usuario la edad. Lanza una excepción en caso de que no se trate de una edad válida. Se considera edad no válida una edad negativa o mayor a 150. En cada caso, lanza el mensaje correspondiente.

Ejercicio 10

Crea una función que solicite al usuario una letra en mayúscula. Lanza una excepción en caso de que el usuario no haya introducido nada, no haya introducido una letra, no haya introducido una letra mayúscula o no haya introducido solamente un carácter.