


Open · Opened 4 days ago by  **Rubén Montero**

Conversión de cualquier cosa a número

Resumen

- Añadiremos una nueva función a `example3.py` que calcula el coste anual de un seguro dental
- Nos enfrentaremos a problemas *multiplicando* valores `str` con `*`
- Usaremos la utilidad propia de Python `int(...)` para solventarlos

Descripción

Completemos un poco más nuestro fichero `example3.py`.

De la tarea anterior, tenemos una función que devuelve el precio mensual de un seguro en **tipo** `str` ó `int`, según cuadre:

```
def dental_insurance_cost(company_name):  
    if company_name == "Ratoncito Pérez S.L.":  
        return "40"  
    else:  
        return 30
```

Una nueva función

...que calcula e imprime por pantalla el coste anual de una compañía recibida por parámetro.

Este código debería ser fácilmente comprensible:

```
def show_annual_cost(a_company):  
    cost = dental_insurance_cost(a_company)  
    annual_cost = cost * 12  
    print("Si contratas la compañía " + a_company + ", el coste anual será " + annual_cost + " euros")
```

Añadámoslo justo debajo de `dental_insurance_cost`, en el mismo nivel (primer nivel) de *indentación*.

Ahora, ¡podremos *invocarlo* desde el cuerpo principal!

Usando `show_annual_cost`

```
if __name__ == '__main__':  
    # Código de tarea anterior  
    # ...  
  
    # Nueva tarea - 11  
    show_annual_cost("Dientes Baratos S.L.")
```

Si lo ejecutamos...

```
File "C:\Users\Developer\...\python-introduction\intro\example3.py", line 11, in show_annual_cost  
    print("Si contratas la compañía " + a_company + ", el coste anual será " + annual_cost + " euros")  
TypeError: can only concatenate str (not "int") to str
```

¡Vaya! El código contiene un **error** dentro de `show_annual_cost`.

Pero es el mismo **problema** que en la tarea anterior. Ya sabemos **solventarlo** con `str(...)`:

```
- print("Si contratas la compañía " + a_company + ", el coste anual será " + annual_cost + " euros")  
+ print("Si contratas la compañía " + a_company + ", el coste anual será " + str(annual_cost) + " euros")
```

Ejecutamos de nuevo

¡Y vemos un excelente resultado **después** del *print* de la tarea anterior!

```
...
Si contratas la compañía Dientes Baratos S.L., el coste anual será 360 euros
```

Probemos de nuevo...

Usemos ahora el **nombre** de otra compañía, como **Ratoncito Pérez S.L.** :

```
if __name__ == '__main__':
    # Código de tarea anterior
    # ...

    # Nueva tarea - 11
    show_annual_cost("Dientes Baratos S.L.")
    show_annual_cost("Ratoncito Pérez S.L.")
```

¡Ups!

¿Qué ves?

No se trata de un error de ejecución... Pero tampoco es que esté *bien*.

Parece demasiado caro, ¿no?

Sí.

Creo que entiendo *cómo* el operador ***** afecta a **strings**

¡Genial!

Pero yo quiero multiplicar un **número**

Entonces debemos asegurarnos de que el valor que se *multiplica por 12* es numérico.

Este es el **problema** de la tarea anterior, pero **a la inversa**. O sea, ahora queremos convertir un **string** a **int**.

¡Fácil!

En Python, podemos usar **int(cualquierCosa)** para convertir **cualquierCosa** a una variable que es un *número int*.

Si no es convertible, saltará un **ValueError**, aunque no nos preocuparemos por eso ahora.

Tan sólo **sustituiremos** la línea problemática en **show_annual_cost** :

```
- annual_cost = cost * 12
+ annual_cost = int(cost) * 12
```

¡Listo!

Por último

Verifica que tu código pasará el *test* asociado a la tarea correctamente.

Haz **commit** y **push** para subir los cambios al repositorio.



Rubén Montero @ruben.montero changed milestone to [%Sprint 3](#) 4 days ago