



Python : Mini projet N°9

DECORME Hector
22/11/2021

Vous devez fournir la fonction draw_n_squares(n) qui renvoie une ch

```
>>> from solution import draw_n_squares
```

```
>>> print(draw_n_squares(1))
```

```
+---+
```

```
| |
```

```
+---+
```

```
>>> print(draw_n_squares(5))
```

```
+---+---+---+---+---+
```

```
| | | | | |
```

```
+---+---+---+---+---+
```

```
| | | | | |
```

```
+---+---+---+---+---+
```

```
| | | | | |
```

```
+---+---+---+---+---+
```

```
| | | | | |
```

```
+---+---+---+---+---+
```

```
| | | | | |
```

```
+---+---+---+---+---+
```



```

def draw_n_squares(n):
    """Cette fonction sert à dessiner des carrés
    -----
    Parametre:
    |         (n)int: nombre de carrés sur la diagonale
    -----
    Retour:
    |         result: le carré * n
    -----
    On ne peut pas dépasser 35 carré sur la diagonale car l'afficheur n'est pas assez grand
    """
    result=""
    for i in range(n):
        result+="+"+"---+"*n+"\n"+"|"+(" "+"|")*n+"*(3*n)+"\n"#création du carré en fonction de n
        result+="+"+"---+"*n #création de la base du carré +---+
    return result

```

```

from solution import draw_n_squares
nbr=0
nbr=int(input("donnez un nombre entre 0 et 35: "))
while nbr>35 or nbr<0 or nbr==0:
    print("nombre invalide")
    nbr=int(input("donnez un nombre entre 1 et 35 compris: "))
print(draw_n_squares(nbr))

```

donnez un nombre entre 0 et 35: 5

```
+---+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+---+
```

```
(function) def draw_n_squares(n: Any) -> Any
```

Cette fonction sert à dessiner des carrés

Parametre:

(n)int: nombre de carrés sur la diagonale

Retour:

result: le carré * n

On ne peut pas dépasser 35 carré sur la diagonale car l'afficheur n'est pas assez grand

```
"""Cette fonction sert à dessiner des carrés
```

```
-----
```

```
Parametre:
```

```
| (n)int: nombre de carrés sur la diagonale
```

```
-----
```

```
Retour:
```

```
| result: le carré * n
```

```
-----
```

```
On ne peut pas dépasser 35 carré sur la diagonale car l'afficheur n'est pas assez grand
```

```
"""
```



Python Sphinx Highlighter v0.3.0

Leonhard S.

Highlight Sphinx-specific markup inside Python docstrings.

Disable

Uninstall



This extension is enabled globally.

DETAILS

FEATURE CONTRIBUTIONS

RUNTIME STATUS

Python Sphinx Highlighter

This is an extension for [Visual Studio Code](#) designed to improve readability of [Python](#) docstrings using the [Sphinx docstring](#) format.



Categories

Programming Languages

More Info

```

import tkinter as tk
root = tk.Tk()
canvas1 = tk.Canvas(root, width=300, height=300)
canvas1.pack()
def nbr():
    valeur=int(input("quelle est votre valeur de n ? : "))
def draw_n_squares(n):
    """Cette fonction sert à dessiner des carrés
    -----
    Parametre:
    |         (n)int: nombre de carrés sur la diagonale
    -----
    Retour:
    |         result: le carré * n
    -----
    On ne peut pas dépasser 35 carré sur la diagonale car l'afficheur n'est pas assez grand
    """
    label1 = tk.Label(
        root, text="n", fg="blue", font=("helvetica", 12, "bold")
    )
    canvas1.create_window(150, 200, window=label1)
    result=""
    for i in range(n):
        result+="+"+"---+"*n+"\n"+"|"+"("   "+"|")*n+""*(3*n)+"\n"#création du carré en fonction de n
        result+="+"+"---+"*n #création de la base du carré +---+
    return result
button2 = tk.Button(text="choisissez votre valeur de n : ", command=nbr, bg="brown", fg="white")
canvas1.create_window(300, 300, window=button2)
button1 = tk.Button(text="executer", command=draw_n_squares(nbr), bg="brown", fg="white")
canvas1.create_window(150, 150, window=button1)

root.mainloop()

```