

Інтерфейс користувача на мові Python.

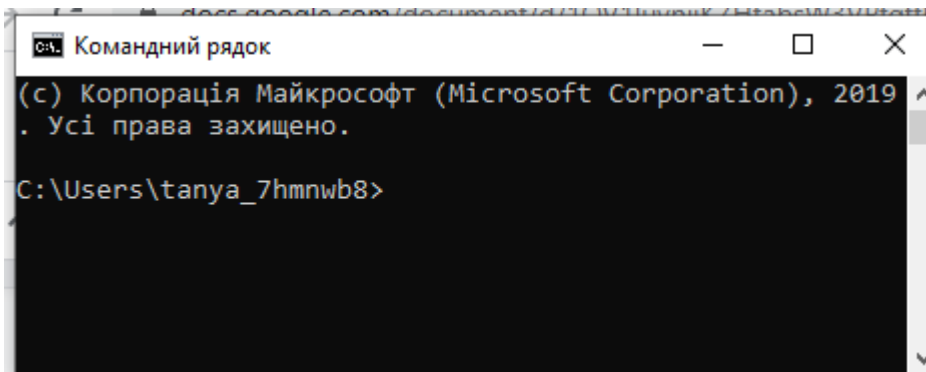
Створення вікон та налаштування їх властивостей

1. Інтерфейс користувача у мові Python

Інтерфейс – це сукупність засобів і правил, що забезпечують взаємодію пристроїв обчислювальної системи або програм з користувачем.

Існує 2 види інтерфейсу програм:

- **командний:** взаємодія користувача та комп'ютера відбувається за допомогою команд, які користувач вводить в командний рядок з клавіатури



- **графічний** інтерфейс є більш зручнішим для користувача, адже взаємодія відбувається за допомогою **кнопок, полів, та інших елементів управління.**

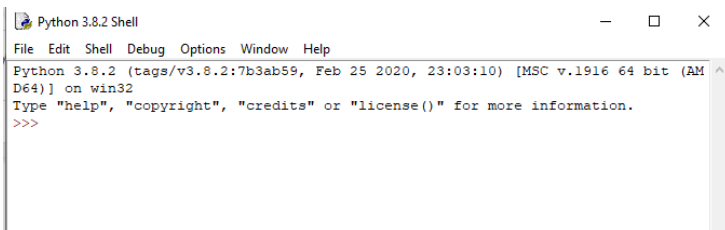
Ми навчимося створювати мовою програмування Python програми, які будуть мати свій власний графічний інтерфейс.

Форма - об'єкт, в якому можна розмістити різні компоненти (елементи керування), зокрема кнопки, поля, написи, меню та інше. Зазвичай форми представляють собою звичайні програмні вікна, в яких будуть відображатися вищезазначені елементи. Тож і ми спробуємо створити вікно програми мовою Python з різноманітними властивостями.

Мова Python більше орієнтована на командний інтерфейс, але в ній існує спеціальний модуль за допомогою якого можна створити інтерфейс користувача.

Модулем у програмуванні називають певний пакет додаткових функцій для роботи над розробкою програм.

Сам IDLE не зможе створити окреме вікно, максимум, що ми можемо отримати – це рядки з текстом, безпосередньо у вікні самого IDLE.



Потрібно створити новий файл і там писати програму. Оскільки IDLE не може створювати окреме вікно програми самостійно, то для цього ми повинні підключити додатковий модуль **tkinter**.

Для підключення будь-яких модулів використовується конструкція:

```
import назва_модуля
```

Для того, щоб повторно не вказувати модуль при використанні функцій використовують конструкцію:

```
from назва_модуля import *
```

Знак **"*"** вказує, що з даного модуля ми імпортуємо всі функції.

Модуль, який нам знадобиться для створення графічного інтерфейсу, програми називається **tkinter**. За допомогою цього модуля можна виконувати як окремі графічні побудови, так і створити повноцінний графічний інтерфейс користувача.

Перший рядок коду в нашому новому файлі буде завжди виглядати так:

```
from tkinter import * (1)
```

2. Створення вікон та налаштування їх властивостей

Після підключення модулів, створимо вікно.

Для створення вікна використовується функція **Tk()**, але, так як таких вікон можна створити нескінченно багато, то кожному вікну присвоюється певне ім'я:

```
назва_вікна=Tk()
```

Назва вікна може складатися з великих і малих літер англійського алфавіту, цифр та знаку нижнього пробілу. Наприклад, створимо вікно *Window1*:

Window1=Tk() (2)

Назва вікна потрібна для того, щоб потім при написанні коду ми змогли звертатися до даного вікна, присвоювати йому певні властивості та вставляти у нього елементи управління.

Важливо!!! після створення вікна та його елементів управління потрібно вказати інтерпретатору, що ми закінчили працювати з вікном за допомогою метода *mainloop()*. Застосовується він так:

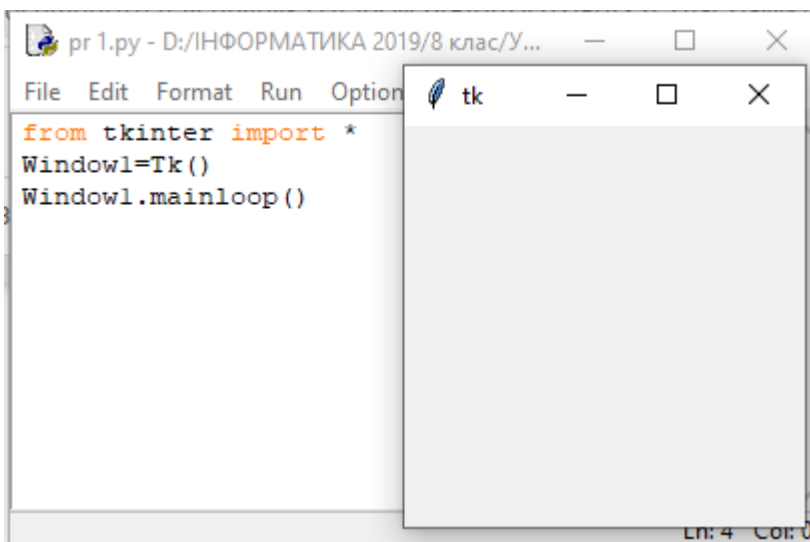
назва_вікна.mainloop()

У випадку із нашим вікном цей рядок виглядатиме так:

Window1.mainloop() (3)

Ці всі команди та рядки коду (1-3) об'єднаємо у один файл, який ми створили та подивимося на те, що ми отримали. Спробуємо запустити нашу програму. Як ми бачимо у нас утворилося пусте вікно:

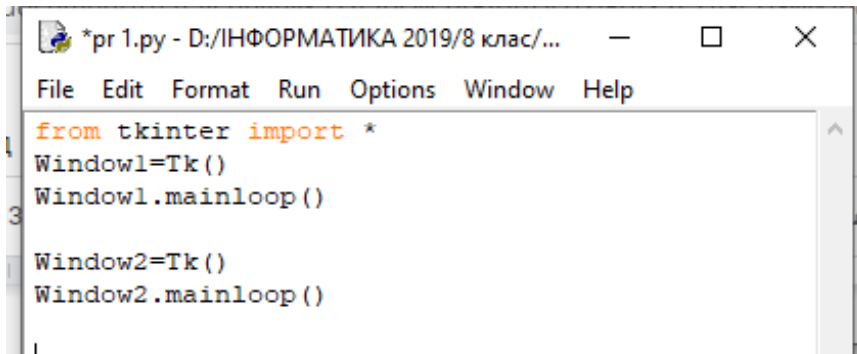
Приклад №1



Можна створювати велику кількість вікон, але в залежності від того, як ми розмістимо рядки коду, ці вікна будуть по різному відображатися.

Приклад №2

Якщо функцію створення та метод закриття одного вікна ми розмістимо послідовно з функціями та методами іншого, *то друге вікно буде створене тільки після того, як користувач закрив перше:*

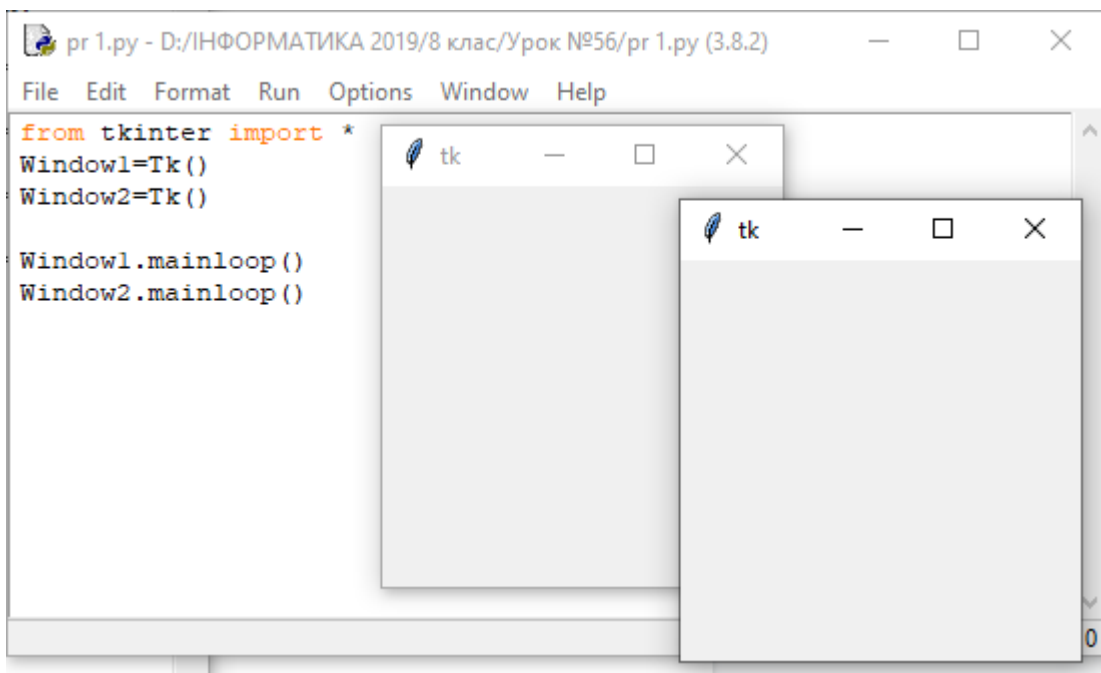


```
*pr 1.py - D:/ІНФОРМАТИКА 2019/8 клас/...
File Edit Format Run Options Window Help
1 from tkinter import *
2 Window1=Tk()
3 Window1.mainloop()
4
5 Window2=Tk()
6 Window2.mainloop()
7
```

Для того, щоб краще зрозуміти написане, рекомендую повторити дії та запустити команду і побачити на практиці, як це працює.

Приклад №3

А для того, щоб створити ці вікна паралельно, ми повинні послідовно розмістити відповідні команди між собою:



```
pr 1.py - D:/ІНФОРМАТИКА 2019/8 клас/Урок №56/pr 1.py (3.8.2)
File Edit Format Run Options Window Help
1 from tkinter import *
2 Window1=Tk()
3 Window2=Tk()
4
5 Window1.mainloop()
6 Window2.mainloop()
7
```

Властивості вікна:

- **title("Заголовок")** – заголовок вікна. За замовчуванням заголовок вікна «Tk»;

- **minsize(x,y)** – мінімальний розмір вікна у пікселях, x- ширина, y – висота. Якщо не встановити цю властивість, то вікно не матиме обмежень у зменшенні;

- **maxsize(x,y)** – максимальний розмір вікна у пікселях, , x- ширина, y – висота. Якщо не встановити цю властивість, вікно може приймати повні розміри екрану користувача;

- **geometry("400x200+450+150")** – розміри та положення вікна відносно розширення екрану у пікселях. 400 - ширина, 200-висота, 450- відстань від лівого краю екрану, 150 – відстань від верхнього краю екрану. Параметри відступу (відстань від лівого краю екрану та відстань від верхнього краю екрану) не є обов'язковими, вони додаються за бажанням. **Зверніть увагу!** Всі числа були введені як приклад;

- **resizable(x,y)** – чи може користувач змінювати розміри вікна і на скільки: x – ширина, y – висота. **Для заборони змінення розмірів вікна замість параметрів встановити нулі: resizable(0,0)**

До вікна властивості застосовуються таким чином:

назва_вікна.властивість(параметри)

Наприклад, **встановимо розміри нашого вікна Window1: ширина та висота по 500 пікселів:**

Window1.geometry("500x500") (4)

Аналогічно застосовуються й інші властивості.

Колір фону вікна

Щоб застосувати певний колір фону для вікна виконується інша конструкція:

назва_вікна["bg"]="колір"

Bg скорочено від background (фон), а параметром є назва кольору на англійській мові взята в лапки.

Наприклад, встановимо зелений колір для нашого вікна:

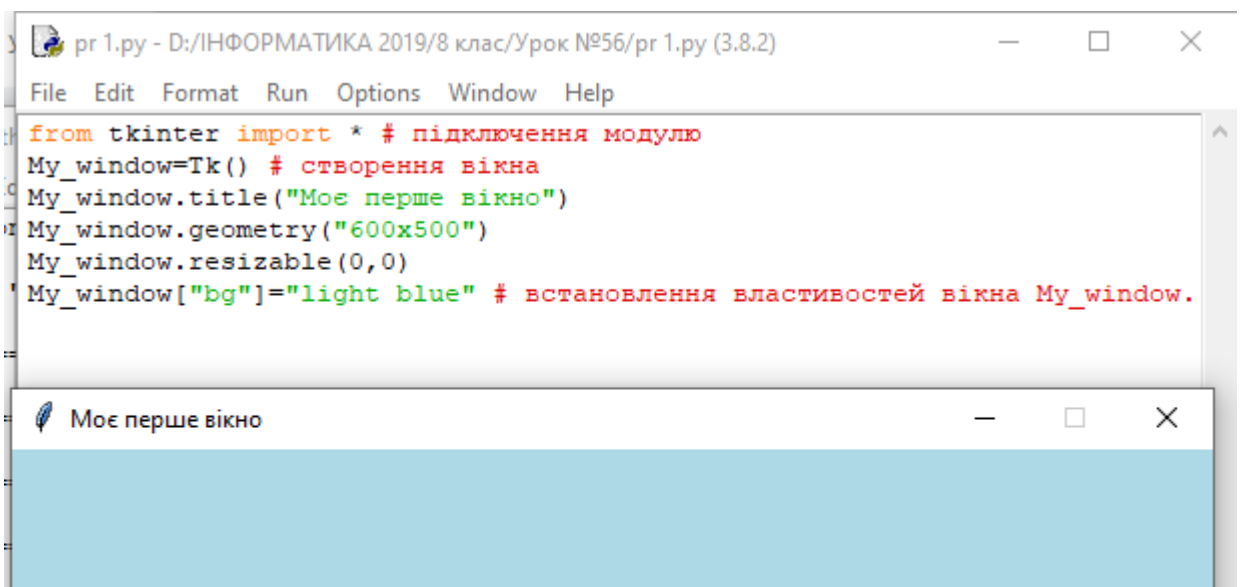
Window1["bg"]="green" (5)

На самостійне опрацювання! Знайти якомога більше кольорів, які можливо встановити до вікна tkinter в мові програмування Python.

Приклад №4

Створити вікно світло-блакитного кольору зі сталими розмірами: ширина 600, висота 500 та з заголовком «Моє перше вікно».

Спочатку створюємо вікно, назвемо його як завгодно, наприклад **My_window**. Потім встановимо його розміри за допомогою властивості **geometry**, заборонимо змінювати його розміри та встановимо світло блакитний колір (**light blue**). Не забуваємо додати заголовок вікна та закриваючу команду **mainloop**. Ось що ми отримали після запуску:



Контрольні питання

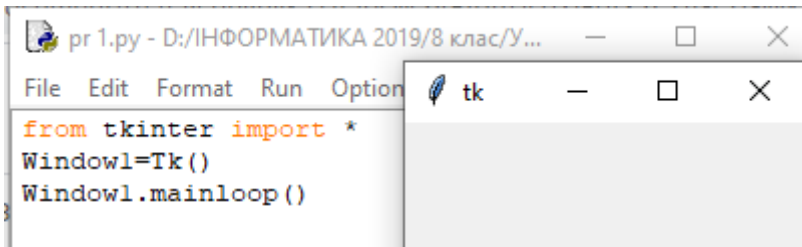
1. Які існують два види інтерфейсу? Чим вони між собою відрізняються?
2. Що таке форма?
3. Що представляють собою форми в мові програмування Python?
4. Як називається модуль для створення графічного інтерфейсу користувача на мові Python? Як його підключити?
5. Як створити вікно на мові Python з використанням модуля tkinter?
6. Які властивості можна надати вікну? Як їх застосовувати?

Практичні завдання до уроку №48

Увага! При роботі за комп'ютером пам'ятайте про дотримання правил техніки безпеки та виконання санітарно-гігієнічних норм.

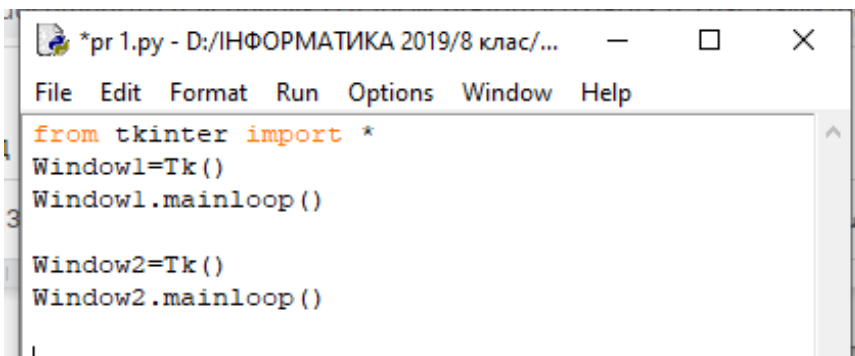
Завдання №1

Напишіть і запустіть код програми для створення вікна за допомогою модуля **tkinter**.



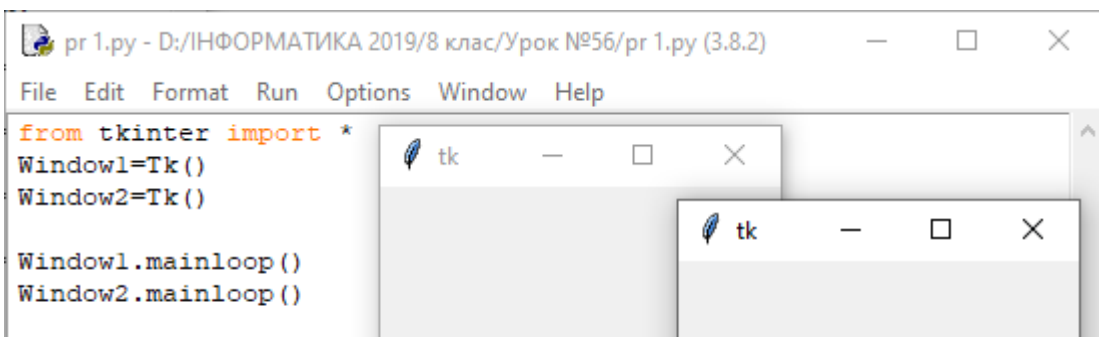
Завдання №2

Напишіть і запустіть код програми для послідовного створення двох вікон (щоб друге вікно відкривалося лише після закриття першого) за допомогою модуля **tkinter**.



Завдання №2.1 Напишіть і запустіть код програми для послідовного створення 4 вікон (щоб друге вікно відкривалося лише після закриття першого) за допомогою модуля **tkinter**. (див. завдання 2)

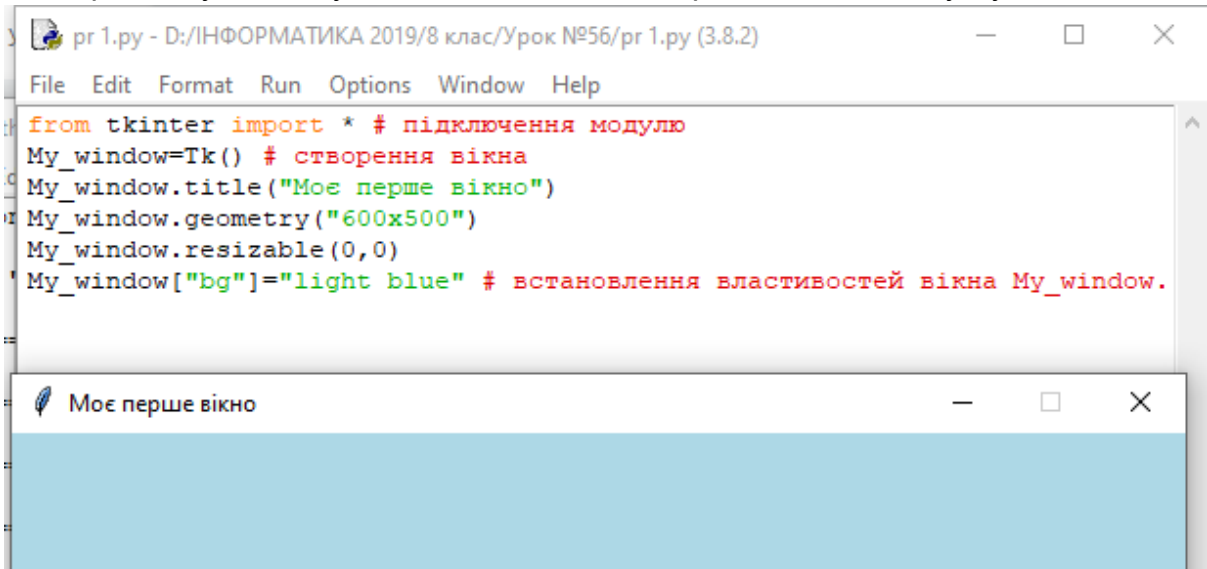
Завдання №3 Напишіть і запустіть код програми для одночасного створення двох вікон за допомогою модуля **tkinter**.



Завдання №3.1 Напишіть і запустіть код програми для одночасного створення 4 вікон за допомогою модуля **tkinter**. (див. завдання 3)

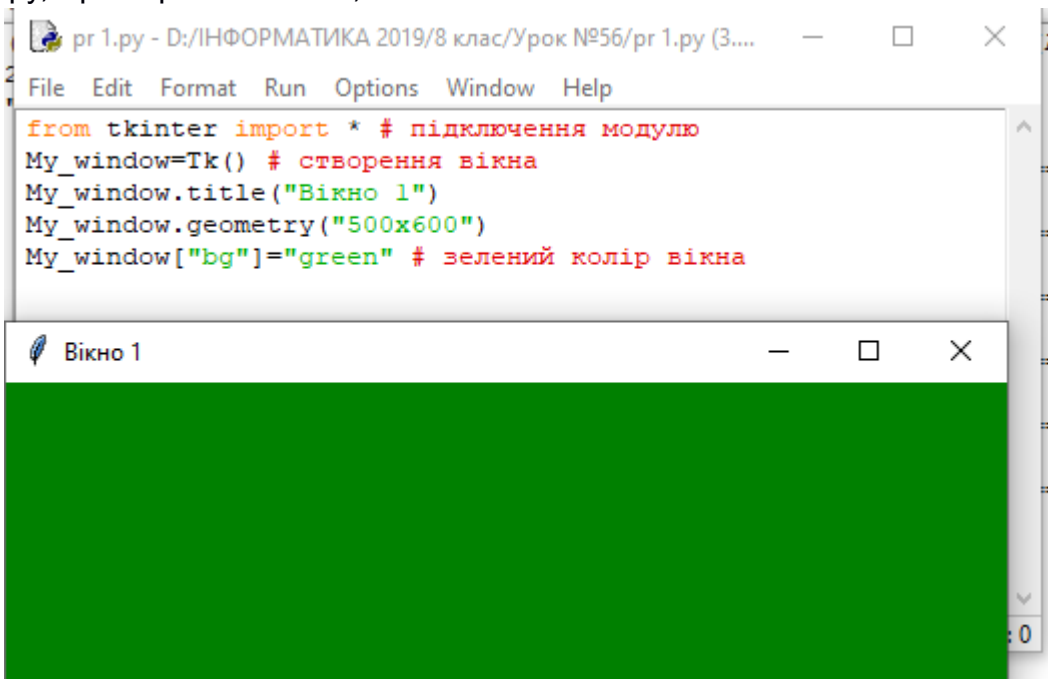
Завдання №4 Створити вікно світло-блакитного кольору зі сталими розмірами: ширина 600, висота 500 та з заголовком «Моє перше вікно».

Спочатку створюємо вікно, назвемо його як завгодно, наприклад *My_window*. Потім встановимо його розміри за допомогою властивості **geometry**, заборонимо змінювати його розміри та встановимо світло блакитний колір (**light blue**). Не забуваємо додати заголовок вікна та закриваючу команду **mainloop**. Ось що ми отримали після запуску:



Завдання №5

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно зеленого кольору, з розмірами 500x600, та заголовком "Вікно 1".



Завдання №6

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно жовтогарячого кольору, з розмірами 400x400, з заголовком "Це вікно!". (див. завдання №5)

Завдання №7

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно білого кольору, з розмірами 1000x500, з заголовком "Заголовок вікна", та з заборонаю змінювати

розміри вікна. (див. завдання №4)

Завдання №8

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно жовтого кольору, з розмірами 654x456, відступом від лівого краю 300 та відступом від верхнього краю 400. Заголовок “Вікно8”, та заборона змінювати розміри вікна.

Завдання №9

Створіть новий файл Python. Підключіть відповідний модуль та створіть вікно блакитного кольору, з розмірами 700x300, відступом від лівого краю 100 та відступом від верхнього краю 0. Заголовок “Вікно9”, та заборона змінювати розміри вікна. Та створити вікно із заголовком “Вікно10” синього кольору, з ідентичними розмірами та заборону змінювати розмір вікна. Вікна розташувати послідовно, тобто 2-ге вікно відкриється лише після закриття першого. (див. завдання 2 та завдання 4).

Додаток

Кольори для Python (tkinter)

Named colour chart												
snow	deep sky blue	gold	seashell3	SlateBlue2	LightBlue3	SpringGreen2	DarkGoldenrod1	brown4	pink3	purple1	gray26	
ghost white	sky blue	light goldenrod	seashell4	SlateBlue3	LightBlue4	SpringGreen3	DarkGoldenrod2	salmon1	pink4	purple2	gray27	
white smoke	light sky blue	goldenrod	AntiqueWhite1	SlateBlue4	LightCyan2	SpringGreen4	DarkGoldenrod3	salmon2	LightPink1	purple3	gray28	
gainsboro	steel blue	dark goldenrod	AntiqueWhite2	RoyalBlue1	LightCyan3	green2	DarkGoldenrod4	salmon3	LightPink2	purple4	gray29	
floral white	light steel blue	rosy brown	AntiqueWhite3	RoyalBlue2	LightCyan4	green3	RosyBrown1	salmon4	LightPink3	MediumPurple1	gray30	
old lace	light blue	indian red	AntiqueWhite4	RoyalBlue3	PaleTurquoise1	green4	RosyBrown2	LightSalmon2	LightPink4	MediumPurple2	gray31	
linen	powder blue	saddle brown	bisque2	RoyalBlue4	PaleTurquoise2	chartreuse2	RosyBrown3	LightSalmon3	PaleVioletRed1	MediumPurple3	gray32	
antique white	pale turquoise	sandy brown	bisque3	blue2	PaleTurquoise3	chartreuse3	RosyBrown4	LightSalmon4	PaleVioletRed2	MediumPurple4	gray33	
papaya whip	dark turquoise	dark salmon	bisque4	blue4	PaleTurquoise4	chartreuse4	IndianRed1	orange2	PaleVioletRed3	thistle1	gray34	
blanched almond	medium turquoise	salmon	PeachPuff2	DodgerBlue2	CadetBlue1	OliveDrab1	IndianRed2	orange3	PaleVioletRed4	thistle2	gray35	
bisque	turquoise	light salmon	PeachPuff3	DodgerBlue3	CadetBlue2	OliveDrab2	IndianRed3	orange4	maroon1	thistle3	gray36	
peach puff	cyan	orange	PeachPuff4	DodgerBlue4	CadetBlue3	OliveDrab4	IndianRed4	DarkOrange1	maroon2	thistle4	gray37	
navajo white	light cyan	dark orange	NavajoWhite2	SteelBlue1	CadetBlue4	DarkOliveGreen1	sienna1	DarkOrange2	maroon3		gray38	
lemon chiffon	cadet blue	coral	NavajoWhite3	SteelBlue2	turquoise1	DarkOliveGreen2	sienna2	DarkOrange3	maroon4		gray39	
mint cream	medium aquamarine	light coral	NavajoWhite4	SteelBlue3	turquoise2	DarkOliveGreen3	sienna3	DarkOrange4	VioletRed1		gray40	
azure	aquamarine	tomato	LemonChiffon2	SteelBlue4	turquoise3	DarkOliveGreen4	sienna4	coral1	VioletRed2		gray41	
alice blue	dark green	orange red	LemonChiffon3	DeepSkyBlue2	turquoise4	khaki1	burlywood1	coral2	VioletRed3		gray42	
lavender	dark olive green	red	LemonChiffon4	DeepSkyBlue3	cyan2	khaki2	burlywood2	coral3	VioletRed4		gray43	
lavender blush	dark sea green	hot pink	cornsilk2	DeepSkyBlue4	cyan3	khaki3	burlywood3	coral4	magenta2		gray44	
misty rose	sea green	deep pink	cornsilk3	SkyBlue1	cyan4	khaki4	burlywood4	tomato2	magenta3		gray45	
dark slate gray	medium sea green	pink	cornsilk4	SkyBlue2	DarkSlateGray1	LightGoldenrod1	wheat1	tomato3	magenta4		gray46	
dim gray	light sea green	light pink	ivory2	SkyBlue3	DarkSlateGray2	LightGoldenrod2	wheat2	tomato4	orchid1		gray47	
slate gray	pale green	pale violet red	ivory3	SkyBlue4	DarkSlateGray3	LightGoldenrod3	wheat3	OrangeRed2	orchid2		gray48	
light slate gray	spring green	maroon	ivory4	LightSkyBlue1	DarkSlateGray4	LightGoldenrod4	wheat4	OrangeRed3	orchid3		gray49	
gray	lawn green	medium violet red	honeydew2	LightSkyBlue2	aquamarine2	LightYellow2	tan1	OrangeRed4	orchid4		gray50	
light gray	medium spring green	violet red	honeydew3	LightSkyBlue3	aquamarine4	LightYellow3	tan2	red2	plum1		gray51	
midnight blue	green yellow	medium orchid	honeydew4	LightSkyBlue4	DarkSeaGreen1	LightYellow4	tan4	red3	plum2		gray52	
navy	lime green	dark orchid	LavenderBlush2	SlateGray1	DarkSeaGreen2	yellow2	chocolate1	red4	plum3		gray53	
cornflower blue	yellow green	dark violet	LavenderBlush3	SlateGray2	DarkSeaGreen3	yellow3	chocolate2	DeepPink2	plum4		gray54	
dark slate blue	forest green	blue violet	LavenderBlush4	SlateGray3	DarkSeaGreen4	yellow4	chocolate3	DeepPink3	MediumOrchid1		gray55	
slate blue	olive drab	purple	MistyRose2	SlateGray4	SeaGreen1	gold2	firebrick1	DeepPink4	MediumOrchid2		gray56	
medium slate blue	dark khaki	medium purple	MistyRose3	LightSteelBlue1	SeaGreen2	gold3	firebrick2	HotPink1	MediumOrchid3		gray57	
light slate blue	khaki	thistle	MistyRose4	LightSteelBlue2	SeaGreen3	gold4	firebrick3	HotPink2	MediumOrchid4		gray58	
medium blue	pale goldenrod	snow2	azure2	LightSteelBlue3	PaleGreen1	goldenrod1	firebrick4	HotPink3	DarkOrchid1		gray59	
royal blue	light goldenrod yellow	snow3	azure3	LightSteelBlue4	PaleGreen2	goldenrod2	brown1	HotPink4	DarkOrchid2		gray60	
blue	light yellow	snow4	azure4	LightBlue1	PaleGreen3	goldenrod3	brown2	pink1	DarkOrchid3		gray61	
dodger blue	yellow	seashell2	SlateBlue1	LightBlue2	PaleGreen4	goldenrod4	brown3	pink2	DarkOrchid4		gray62	