Hoy en día, hay una gran cantidad de navegadores web, todos ellos con el objetivo de hacer nuestra navegación por Internet más rápida, amigable y eficiente.

El navegador web funciona gracias a 2 factores primordiales, el lenguaje HTML y el protocolo HTTP. De hecho ya hemos hablado de este tema en un artículo publicado hace unos meses atrás.

Pero hoy no estoy escribiendo este artículo con el fin de hablar nuevamente de Internet. Estoy aquí por quiero enseñarte a realizar tu propio navegador web de una forma tan sencilla, que no lo podrás creer.

¿Listos? Comencemos.

Primero lo primero:

Antes de comenzar con el proyecto de hoy, quisiera aclarar cuáles son los requisitos necesarios para seguir los pasos que mostrare a lo largo del artículo.

Lo primero que necesitas son conocimientos sobre el lenguaje de programación Python. Si sabes programar en cualquier otro lenguaje, puedes aprender a cerca de Python revisando la gran cantidad de información que se encuentra en este blog.

Ya que tengas conocimientos sobre Python, es necesario disponer de la librería PyQt5, la cual puedes descargar ingresando al siguiente enlace. O de manera más rápida, puedes ingresar el siguiente comando en tu consola: pip install PyQt5.

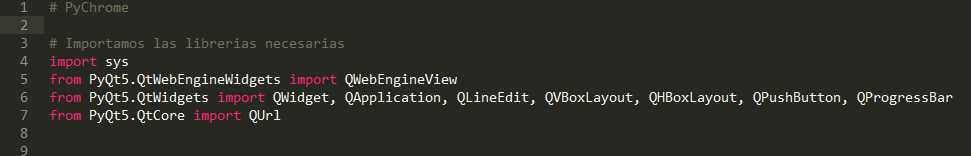
Excelente, ya estamos listo para comenzar con el proyecto de hoy.

El nacimiento de PyChrome:

Perfecto, estamos preparados para comenzar. Antes, debemos bautizar a nuestro navegador, siempre que realicemos un proyecto, así no sea la gran cosa, debemos colocarle un nombre. Yo, como me caracterizo por ser original, he decidido bautizar al proyecto de hoy como: “PyChrome”.

Muy bien, abre tu editor de texto preferido y comencemos a escribir código.

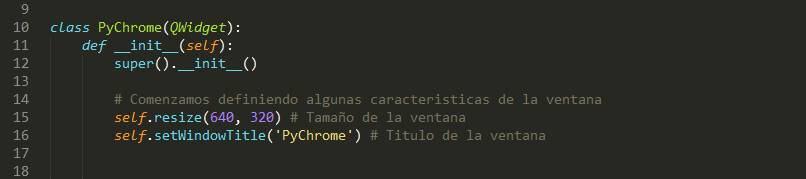
Como siempre, comenzaremos importando todos los recursos necesarios:



Como se observa en la imagen, lo que hacemos es importar algunos métodos y funciones de PyQt5. Como por ejemplo, algo indispensable, es QWebEngineView, el cual nos permitirá generar un cuadro cuyo contenido sea una página web.

Luego de esto, importamos algunas funciones adicionales, por lo cual no digo que no sean importantes, importamos algunos layout los cuales nos ayudaran a posicionar nuestros widget de manera ordenada. También importamos el evento QPushButton el cual nos reaccionara cada vez que un botón sea presionado.

Luego de tener todo lo necesario, podemos proseguir a utilizar estos recursos.

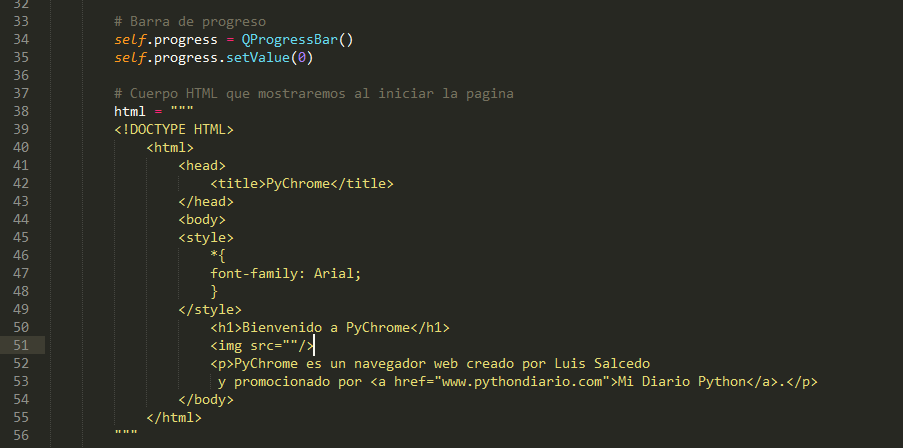


Muy bien, si tienen experiencia con PyQt5 sabran que la mejor manera de crear una ventana es creando una clase que herede lo método y funciones de QWidget. He declarado la clase PyChrome en la cual plasmare todas las características que quiero para mi ventana gráfica.

Como se aprecia en la imagen, lo primero que hago es asignar un tamaño a la ventana, seguido del título de la ventana gráfica.

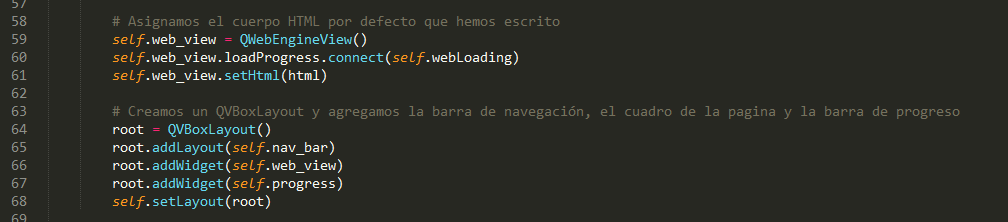


Seguimos asignado valores, creamos la variable “page” la cual será la URL por defecto que aparecerá en la barra de búsqueda. Creamos una barra de búsqueda con QLineEdit, lo cual nos ayuda permite generar una entrada de texto editable (barra de búsqueda). Luego declaramos un botón cuyo texto será “Ir”. Después de declarar nuestros widget, los agregamos un layout “QHBoxLayout”.

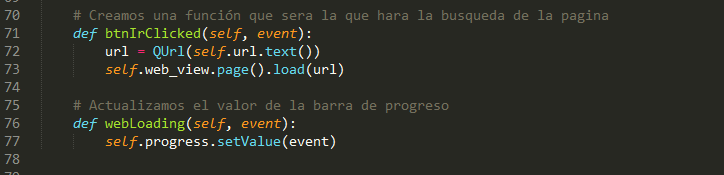


Declaramos una barra de progreso cuyo valor inicial será 0.

Seguido de esto, en una variable llamada “html”, escribiremos el cuerpo inicial que el navegador tendrá al abrir el programa. Como pueden observar simplemente es un texto en formato HTML. Pueden hacerlo como ustedes prefieran, incluso agregar archivos extras a su directorio y crear archivo CSS o JS.

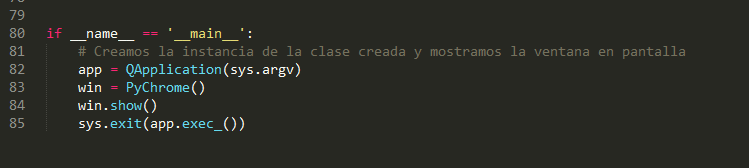


Perfecto, vamos bien hasta ahora. Que hacemos ahora es crear el objeto QWebEngineView, el cual generara el cuadro contenedor de texto HTML. Le decimos al programa al iniciar, el cuerpo HTML por defecto será el que hemos escrito anteriormente. Luego de hacer esto, creamos un layout QVBox y agregamos en el la barra de navegación (layout que contiene la barra de texto y el boton), el “cuadro web” y la barra de progreso.



Muy bien, ahora creamos otros métodos dentro de nuestra clase, como “btnIrClicked”, el cual será llamado cuando el botón “Ir” sea presionado, lo cual actualizara el cuerpo HTML con el de la URL de la página ingresada en la barra de búsqueda.

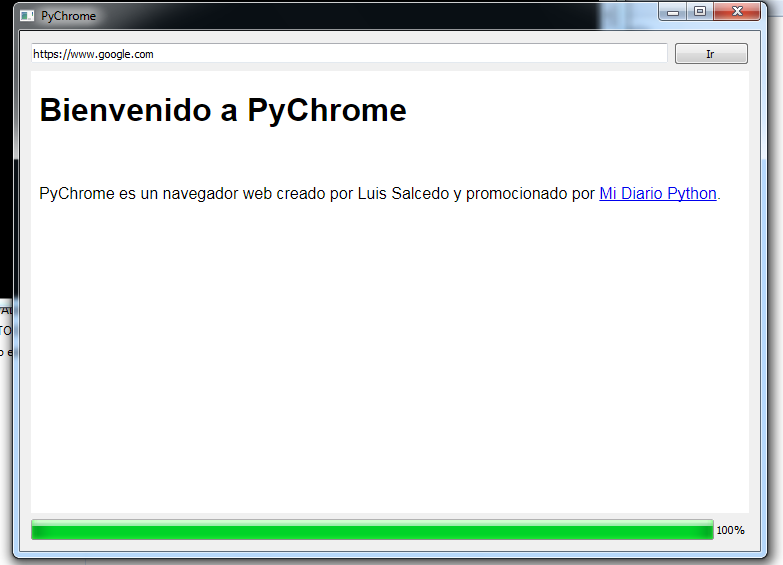
Luego creamos el método “webLoading”, el cual actualizara el valor de la barra de progreso.



Por último, pero no menos importante, creamos una instancia de “QApplication”, y mostramos nuestro programa en pantalla.

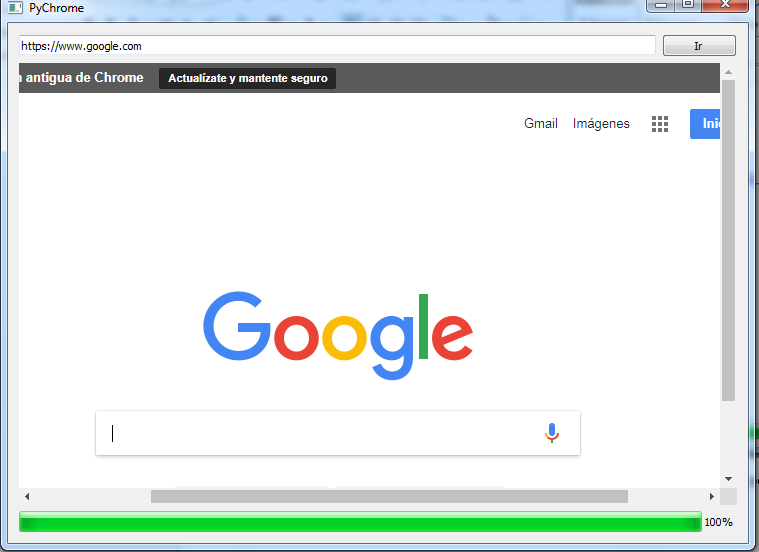
Excelente, si has llegado hasta aquí y seguido todos los pasos, ya estás listo para poner a prueba a “PyChrome”.

Solo debe hacer doble click sobre el script y puedes colocar su ruta en tu consola de comandos.



Y ya tendremos el resultado. Como pueden observar, ya podemos ver el cuerpo HTML por defecto.

¿Pero funciona? Podemos probar.



Como pueden observar. Sí funciona!!!

Puedes probarlo con cualquier sitio web.

Bueno, espero que te haya servido. Estos tipos de proyectos nos ayudan a reforzar nuestros conocimientos y habilidades.

¿Alguna duda? No olvides dejar tu comentario.

Mi nombre es Luis, y fue un placer compartir mis conocimientos con todos ustedes :D.