**UJIAN AKHIR SEMESTER**

**PEMROGRAMAN BERORIENTASI OBYEK**

**DOKUMENTASIAN KIVY APLIKASI NOTE**



**Disusun Oleh :**

**KELOMPOK**

TANGGAL : 07 Desember 2018

NAMA : As’Adur Rofiq

NRP : 140411100121

NAMA : Renaldo Fachrozi

NRP : 140411100138

\

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2018**

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Prinsip pemrograman berorientasi objek adalah membagi modul program yang besar menjadi kumpulan objek-objek yang saling berinteraksi, dengan prinsip seperti ini maka pemrograman berorientasi objek lebih menekankan pada proses yang terjadi antar objek. Sehingga dikemudian akan terdapat kemungkinan sebuah program merupakan kumpulan objek-objek yang saling bertukar informasi dan data dan memiliki fungsi dan tanggung jawab masing-masing.

Kelas dan objek merupakan satuan yang berbeda, Kelas Merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu dimana didalamnya terdapat kumpulan atribut dan method, contoh kelas : binatang, kendaraan, benda dsb.

1. **Tujuan**
2. Mahasiswa dapat memahami karakteristik OOP
3. Mahasiswa dapat membuat kelas project OOP
4. Mahasiswa dapat membuat karya software secara berkelompok berbasis OOP

**BAB II**

**DASAR TEORI**

* 1. **Pengertian OOP**

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

* + 1. **Konsep OOP**
* Abstrak Class
  1. Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.
  2. Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.
  3. Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.
  4. Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (method), dan sifat kelas (property).
* Encapsulation

1. Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.
2. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.
3. Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.
4. Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

* Inheritance
  1. Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
  2. Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
  3. Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
  4. Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.
* Polymorphism
  + 1. Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.
    2. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
    3. Method overriding.
    4. Method name overloading.
  1. **Pengertian Kivy**

Kini Python merupakan salah satu bahasa pemrograman yang populer digunakan oleh developer dalam mengembangkan aplikasi web, web, aplikasi perkantoran, simulasi, dan masih banyak lagi. Python sendiri dapat populer karena disebabkan oleh bahasa pemrogramannya yang mudah dipahami dan dinamis.

Selain itu, berdasarkan tren masa kini dimana developer dapat mengembangkan aplikasi multiplatform dalam sekali langkah, Python juga kini memiliki beberapa alat pengembangan maupun framework untuk mengembangkan aplikasi mobile secara multiplatform. Bagi pembaca yang tertarik untuk mengembangkan aplikasi mobile secara multiplatform menggunakan bahasa pemrograman Python, ada baiknya mencoba salah satu framework yang bernama Kivy.

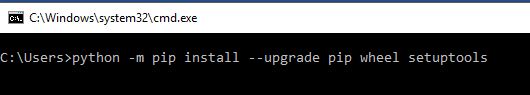
Kivy ini sendiri merupakan framework yang dibangun menggunakan library dari bahasa pemrograman Python yang bersifat Open Source. Tujuan dikembangkannya framework ini agar dapat membantu developer secara cepat dalam mengembangkan aplikasi yang memiliki tampilan antarmuka inovatif seperti aplikasi yang mendukung multitouch.

## **2.2.1 Cara menginstall Kivy**

Buka cmd dengan cara tekan tombol “windows + r” ketik “cmd” lalu tekan enter.

### 1. Pada tampilan cmd silahkan anda ketik perintah dibawah ini lalu tekan enter

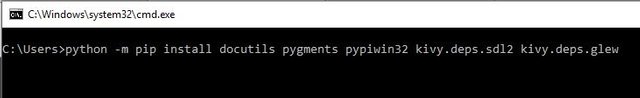
python -m pip install --upgrade pip wheel setuptools



Pastikan bahwa perangkat anda terhubung ke internet, lalu tunggu prosesnya hingga selesai, mungkin akan memakan waktu sekitar 5 menit tergantung pada kecepatan internet anda.

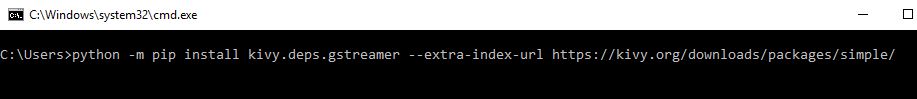
### 2. Jika langkah pertama sudah selesai ketik perintah berikut pada cmd pula, tunggu hingga proses selesai

python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew



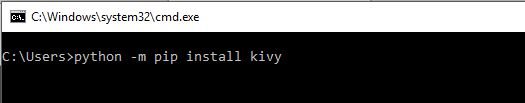
### 3. Sama seperti sebelumya

python -m pip install kivy.deps.gstreamer --extra-index-url https://kivy.org/downloads/packages/simple/

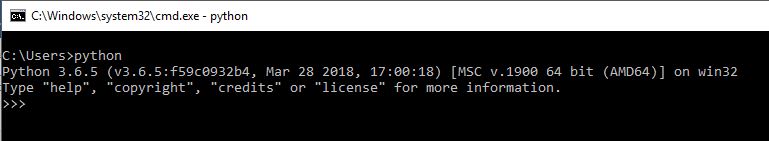


### 4. Langkah terakhir

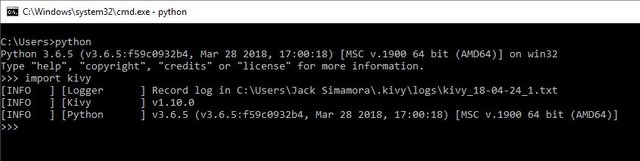
python -m pip install kivy



Untuk memastikan bahwa Kivy sudah terinstall dengan benar silahkan buka “cmd” lalu ketik python dan enter.



Ketik import kivy lalu enter



Jika kivy sudah terinstall dengan benar maka tampilannya akan sedikit seperti gambar yang diatas

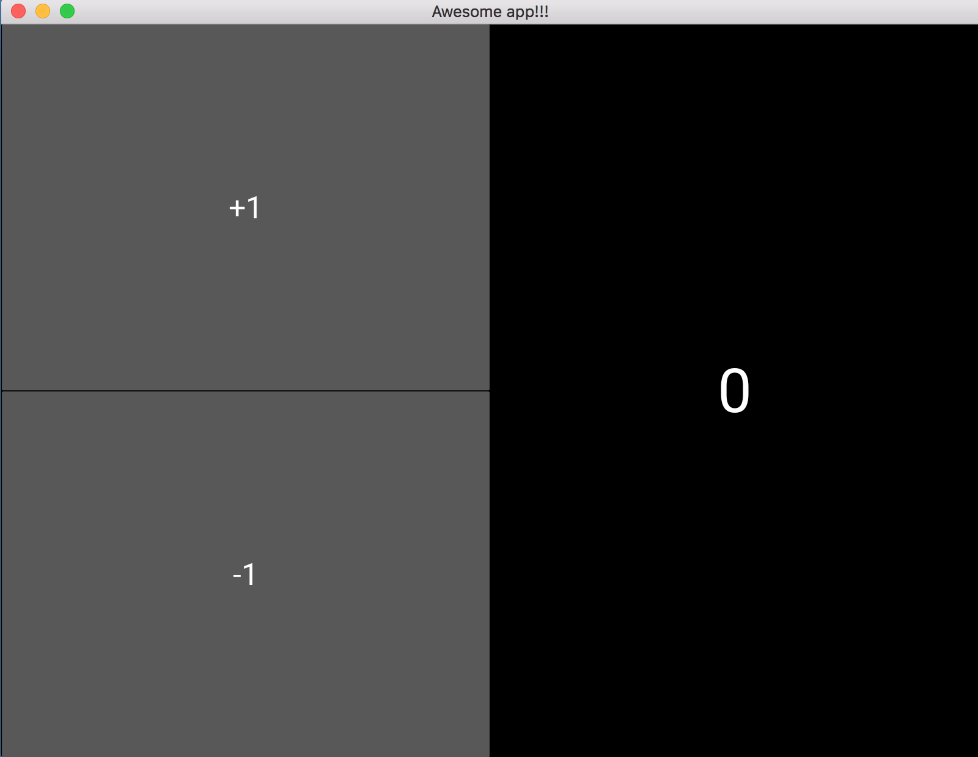
**2.2.2 Kivy(Implementasi OOP dengan Python dan Kivy)**

* cara untuk membuat aplikasi Android dengan Python adalah ... apa yang dapat Anda lakukan dengannya? Apakah cukup berkinerja untuk game, dapatkah Anda memanggil API Android, apakah semua aplikasi tampak sama?
* aplikasi mobile secara multiplatform menggunakan bahasa pemrograman Python
  + official home page: https://kivy.org/#home
  + Download kivy: https://kivy.org/#download
  + Instalation:https://steemit.com/aceh/@alucard14/cara-membuat-aplikasi-android-python-and-kivy-1-or-install-program-python-and-kivy
  + Dokumentation:https://www.codementor.io/kiok46/beginner-kivy-tutorial-basic-crash-course-for-apps-in-kivy-y2ubiq0gz
  + Exampleproject : https://kivy.org/doc/stable/examples/gallery.html

**2.2.3 Contoh Tutorial Game**

Konten Posting ini akan mencakup aplikasi yang sangat dasar pemula mutlak di Kivy) di mana Anda hanya menambahkan (+1) atau mengurangi (-1) menggunakan 2 tombol dan memperbarui nilai baru ke label. Pada akhir tutorial ini, Anda akan memiliki (saya harap) beberapa pengetahuan dan pemahaman yang baik tentang cara bekerja dengan Kivy. Berikut adalah beberapa hal yang akan kami bahas dalam posting ini:

* Memulai
* Metode App () & GridLayout secara detail
* Apa itu metode membangun dan apa fungsinya?
* Penataan aplikasi anda
* Apa itu on\_release dan id?
* Properti objek



### **Mari kita mulai**

Baiklah, mari kita mulai dengan jendela hitam yang tidak melakukan apa-apa. Buat file main.py.

**from** kivy.app **import** App

**from** kivy.uix.gridlayout **import** GridLayout

**class** **Container**(GridLayout):

**pass**

**class** **MainApp**(App):

**def** **build**(self):

self.title = 'Awesome app!!!'

**return** Container()

**if** \_\_name\_\_ == "\_\_main\_\_":

app = MainApp()

app.run()

**dan buat file main.kv**

<Container>:

rows: 1

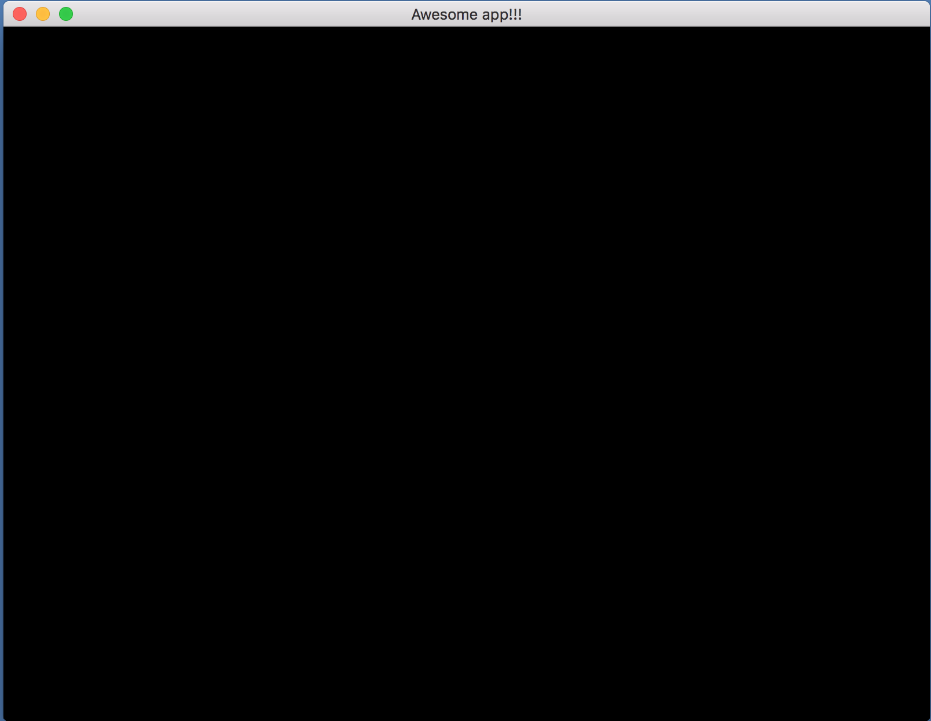
**Current file structure**:

Basic\_app/

- main.py

- main.kv

**Dan anda membuat.**



**BAB III**

**TUGAS**

* 1. **Soal**

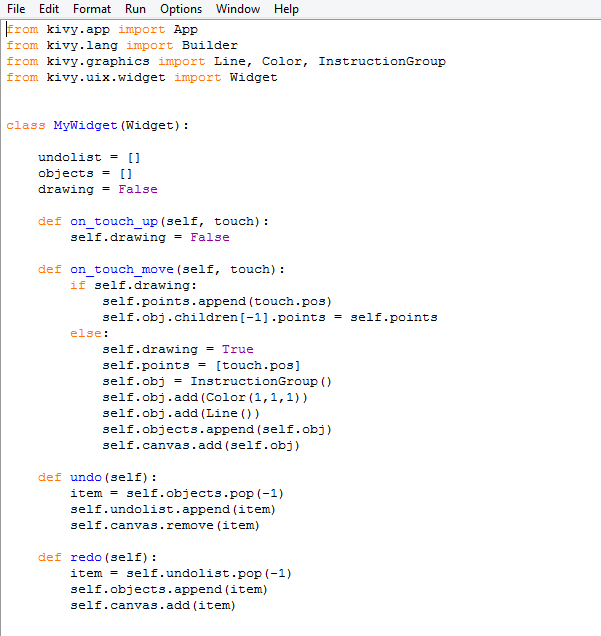
1. Mencoba Kivy dan Memahami Kivy Menjalankan project kivy yang sudah ada ditambahkan dengan mendokumentasikan Meng-customize project kivy di sertai dengan lampirkan referensi sumber

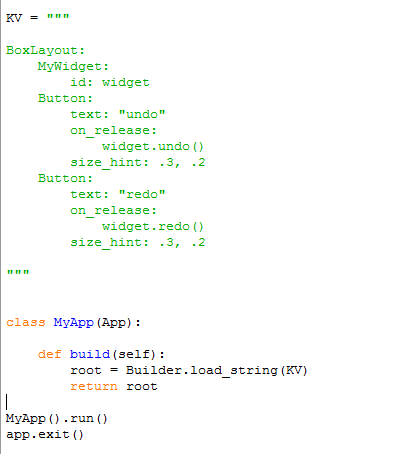
Keterangan:

Sebelum memulai ke pemerogaman saya ingin menjelaskan bahwa saya mengunakan sebuah software Python dengan liberary sebuah Kivy

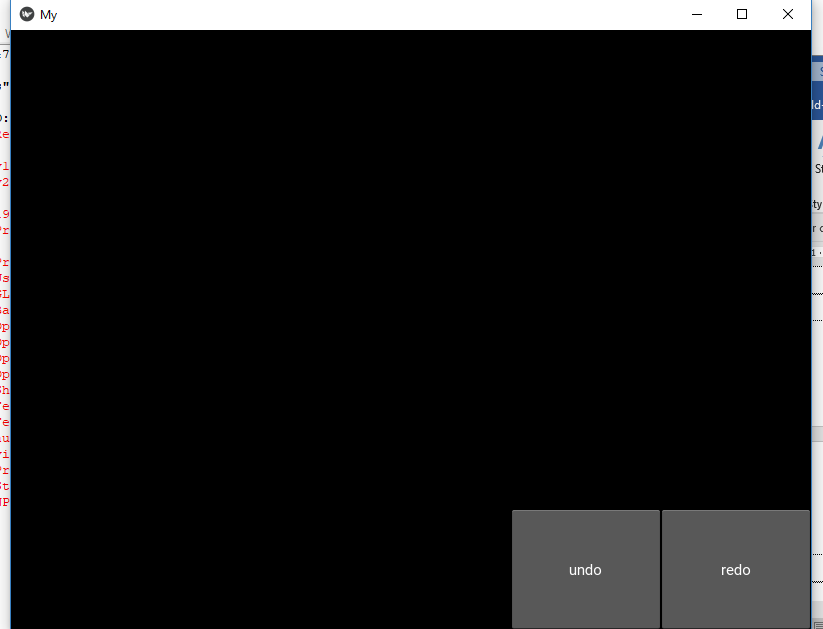
Dengan tujuan agar mempermudah saya untuk membuat program dengan konsep OOP dengan terstruktur dan untuk mempermudah tentunya dan dalam proses pendokumentasian ini hanya berfokus kepada penjelasan konsep OOP nya saja, untuk code programnya saya sertakan dalam Gambar, Dalam project ini kami membuat sebuah aplikasi MengambarRGB menunakan konsep OOP

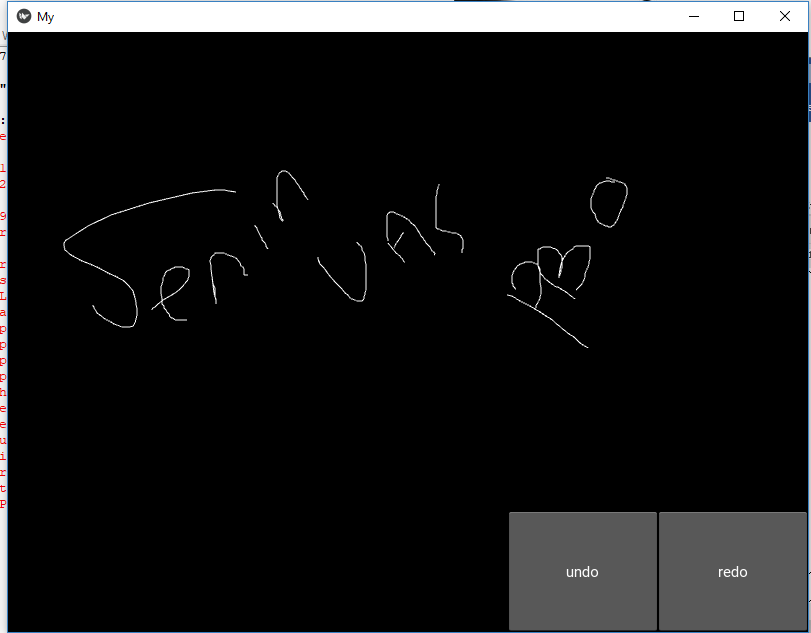
* 1. **Source Code Aplikasi Note**

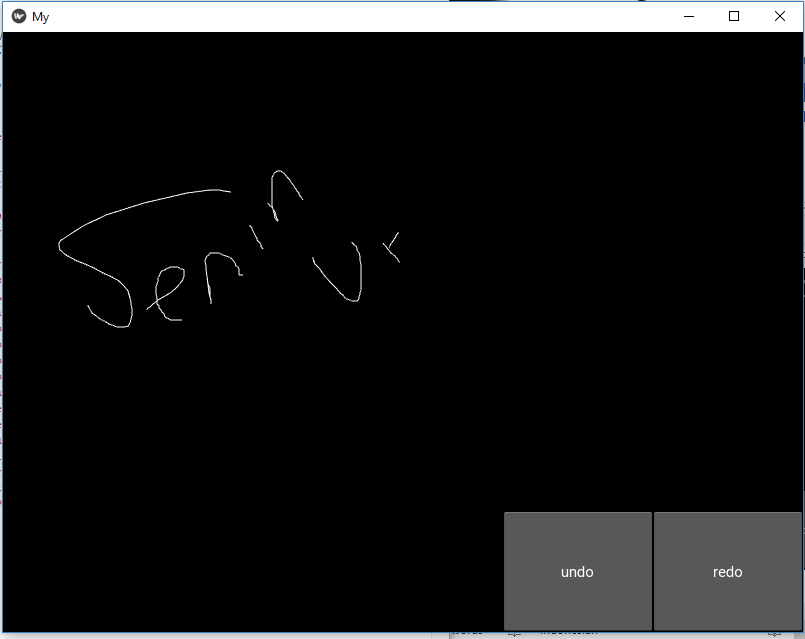




**HASIL RUN :**







**BAB IV**

**PENUTUP**

**Kesimpulan**

Pemrograman berorientasi objek (*object-oriented programming* disingkat OOP) merupakan [paradigma pemrograman](https://id.wikipedia.org/wiki/Paradigma_pemrograman) yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika [pemrograman terstruktur](https://id.wikipedia.org/wiki/Pemrograman_terstruktur). Setiap objek dapat menerima [pesan](https://id.wikipedia.org/wiki/Pesan), memproses data, dan mengirim pesan ke objek lainnya,

Kelas dan objek merupakan satuan yang berbeda, Kelas Merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu dimana didalamnya terdapat kumpulan atribut dan method,liberary pada sebuah bahasa pemerogaman berperan sebagai pembantu untuk mempermudah dan mempersingkat sebuah program.

**Refrensi :**

https://kivy.org/doc/stable/guide/widgets.html

https://www.youtube.com/watch?v=MYUPeCZSs3U

http://robertour.com/2013/07/19/10-things-you-should-know-about-the-kivy-canvas/

https://blog.kivy.org/2014/10/updating-canvas-instructions-declared-in-python/