

شروع کار با رابط کاربری گرافیکی (GUI) در رسیبری پای با پایتون (قسمت سوم)



در این آموزش خواهیم دید که چگونه می‌توان رابط‌های گرافیکی (GUIs) ساده را در رسیبری پای به زبان پایتون بسازیم.

مرور

در دو آموزش "[شروع کار با رابط کاربری گرافیکی \(GUI\) در رسیبری پای با پایتون \(قسمت اول\)](#)" و "[شروع کار با رابط کاربری گرافیکی \(GUI\) در رسیبری پای با پایتون \(قسمت دوم\)](#)" شما آموختید که چگونه چند ویجت ساده را استفاده کنید. کد کامل برای این بخش از آموزش‌ها را می‌توانید در انتهای آموزش دانلود کرده و ببینید تا متوجه شوید که یک مثال کامل چگونه کار می‌کند.

اکنون بیایید نگاهی به ویجت‌های پیچیده‌تر GUI بیندازیم: جعبه‌های combo، جعبه‌های check، دکمه‌های رادیو و منوی ابزار. در ادامه این آموزش با نحوه کار با این ویجت‌ها آشنا خواهیم شد.

یک App جدید

- یک فایل جدید ایجاد کرده و بانام py ذخیره کنید.
- یک خط کد در ابتدای فایل خود برای ورود کلاس app از کتابخانه guizero اضافه کنید.

```
from guizero import App
```

- اکنون دو خط کد دیگر برای ساختن App و سپس نمایش آن بر روی صفحه نمایش اضافه کنید.

```
(app = App(title="My second GUI app", width=300, height=200, layout="grid") app.display
```

این بار ما تعدادی کلمه کلیدی آرگومان را مشخص کرده‌ایم: width و height که ابعاد پنجره App را تغییر می‌دهد و layout که اجازه می‌دهد ویجت خود را روی یک گرید قابل مشاهده تنظیم کنیم.

ویجت Combo

ویجت combo اجازه انتخاب یک گزینه را از لیست کشویی می‌دهد.

- ویجت combo را به جمله ورودی اضافه کنید.
- ویجت combo را با دستور زیر به GUI اضافه کنید:

```
("film_choice = Combo(app, options=["Star Wars", "Indiana Jones", "Batman"], grid=[0,1], align="left
```

طبق معمول پارامتر app را اضافه کرده‌ایم تا به ویجت بگوییم که app رئیس است.

آرگومان option فهرستی از گزینه‌هایی است که مایلیم در ویجت combo نمایش داده شود.

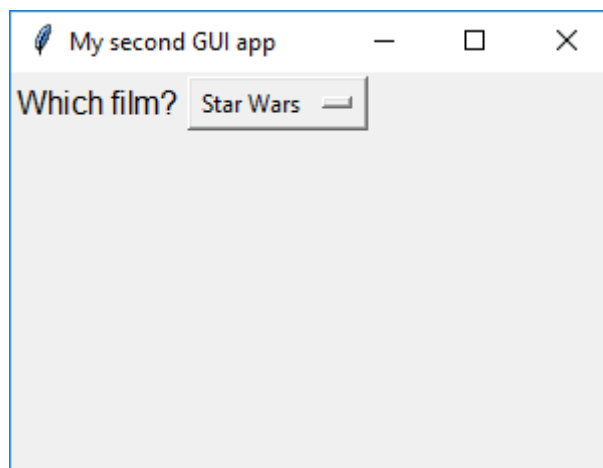
به دلیل اینکه ما layout=grid را درون app مشخص کرده‌ایم، اکنون باید به همراه هر ویجت آرگومان grid را وارد کنیم تا بگوییم در کجا ظاهر شود. آرگومان گرید باید فهرستی شامل مقادیر [x,y] باشد برای جایی که دوست دارید ویجت قرار بگیرد و [0,0] روی گرید گوشه بالا سمت چپ است. ما می‌توانیم ویجت را درون مربع گرید align کنیم.

کدهای خود را ذخیره کنید و F5 را برای اجرای برنامه فشار دهید. توجه داشته باشید که combo در قسمت بسیار بالا و چپ روی صفحه نمایش ظاهر می‌شود، بنابراین ما مکان گرید را [0,1] مشخص کردیم. این به این دلیل است که گرید خالی هیچ عرض یا ارتفاعی ندارد، بنابراین مانند مربع گرید خالی [0,0] است و تأثیری ندارد.

متن را به جمله ورودی وارد کرده، آنگاه یک ویجت متنی در مربع گرید [0,0] اضافه کنید تا مقداری توضیحات برای کسی که از combo برای انتخاب استفاده می‌کند، فراهم آورد.

```
("film_description = Text(app, text="Which film?", grid=[0,0], align="left
```

برنامه را اجرا کرده و بررسی کنید که متن و combo هر دو به خوبی نمایش داده شوند.



ویجت check box

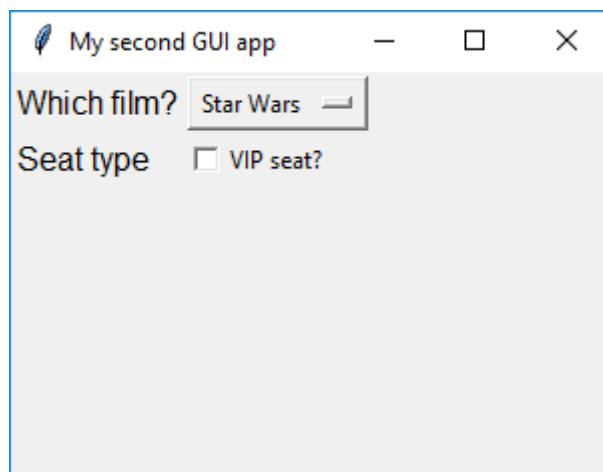
این ویجت به شما اجازه می‌دهد تا یک انتخاب درست یا غلط را انتخاب یا رد کنید.

- ویجت check box را به جمله ورودی اضافه کنید.
- ویجت check box را به GUI اضافه کنید:

```
("vip_seat = CheckBox(app, text="VIP seat?", grid=[1,1], align="left
```

ما همچنین تصمیم گرفتیم یک ویجت متنی در گرید [1,0] برای توضیح اینکه check box برای چه چیزی است، اضافه کنیم که نحوه نوشتن آن مانند قبل است.

- F5 را برای اجرای کد بفشارید. شما باید ببینید که check box ظاهر می‌شود و شما قادرید تیک بزنید یا تیک را بردارید.



ویجت دکمه گروهی (ButtonGroup widget)

ویجت دکمه گروهی به شما اجازه ایجاد گروهی از دکمه‌ها را می‌دهد بنابراین، شما می‌توانید یکی را برای تنظیم گزینه‌ها انتخاب کنید.

- ویجت دکمه گروهی را به جمله ورودی خود اضافه کنید.
- یک ویجت دکمه گروهی به GUI اضافه کنید:

```
row_choice = ButtonGroup(app, options=[ ["Front", "F"], ["Middle", "M"], ["Back", "B"] ], selected="M",
("horizontal=True, grid=[2,1], align="left
```

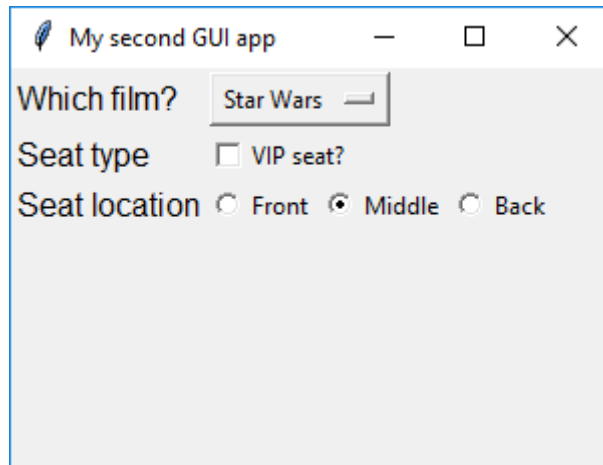
بیا بید نگاه دقیق‌تری به این ببندازیم:

- App به دکمه‌ها می‌گوید که app رئیس آن‌هاست.
- Option یک لیست از گزینه‌هاست که به‌عنوان دکمه ظاهر می‌شود. هر گزینه همچنین یک فهرستی شامل متن است که ظاهر می‌شود و مقدار پنهانی به همراه آن گزینه ظاهر می‌گردد.
- Selected به دکمه‌ها می‌گوید که کدامیک برای شروع با آن انتخاب شده است.
- Horizontal می‌گوید دکمه‌ها به‌صورت خط افقی نمایش داده شوند.
- Grid جایی است که دکمه‌ها روی گرید قرار می‌گیرند.
- align موقعیت دکمه‌ها را روی سمت چپ مربع گرید می‌گوید.

ما همچنین ویجت متنی دیگری در [2,0] برای توضیح اینکه دکمه‌ها که چه هستند اضافه کرده‌ایم.

- F5 را فشار دهید تا برنامه اجرا شود. شما باید دکمه‌ها ببینید. دکمه وسطی باید زمانی که برنامه شروع شود انتخاب شده باشد و شما باید قادر باشید بین

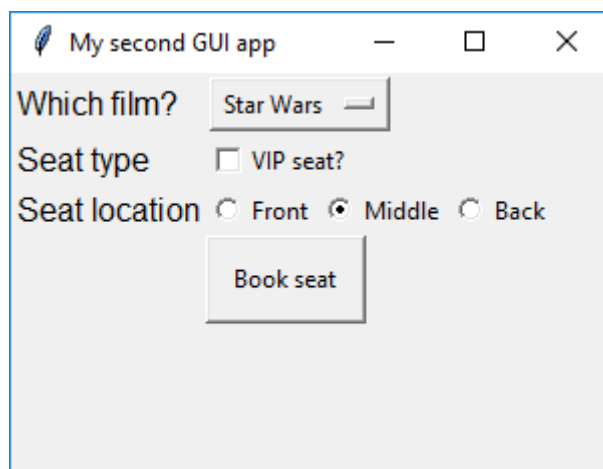
گزینه‌ها انتخاب‌ها را تغییر دهید.



تمام کردن برنامه

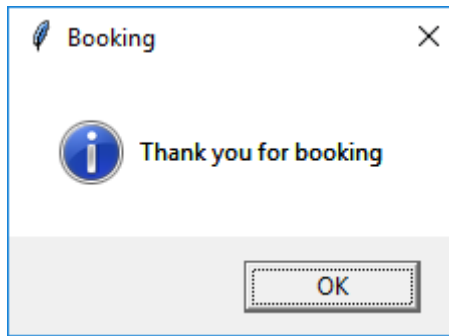
- در انتها، `PushButton` را به لیست ورودی اضافه کرده و سپس یک ویجت `PushButton` در `[3,1]` اضافه کنید تا تابعی را به نام `do_booking` را زمانی که فشرده می‌شود فراخواند.

```
("book_seats = PushButton(app, command=do_booking, text="Book seat", grid=[3,1], align="left
```



- `Info` را به خط ورودی اضافه کنید تا به ما اجازه استفاده از تابع جعبه اطلاعات (`info box`) را از `guizero` بدهد. نحوه اضافه کردن به خط ورودی یا `Import` (`line`) را می‌توانید در "شروع کار با رابط کاربری گرافیکی (GUI) در رسیبری پای با پایتون (قسمت اول)" ببینید.
- خارج از `GUI`، تابع `do_booking` را بنویسید. این یک جعبه اطلاعات فرعی را بازخواهد کرد. توجه شود که این کد باید در خارج از محیط `GUI` نوشته شود.

```
("def do_booking(): info("Booking", "Thank you for booking
```



- شما احتمالاً می‌خواهید بدانید چگونه انتخاب کاربران را بازیابی کنید. این کد را به تابع `do_function` اضافه کنید:

```
( )print(film_choice.get() )print(vip_seat.get_value() )print(row_choice.get
```

توجه داشته باشید که `check box` اگر تیک نخورده باشد به صورت 0 و اگر انتخاب شده باشد به شکل 1 برمی‌گردد؛ و `ButtonGroup` به مقادیر پنهان B, F و یا M به جای متن کامل برمی‌گردد.

```
Star Wars
0
M
```

شما می‌توانید از این آموزش برای نوشتن برنامه‌های GUI خودتان استفاده کنید. به‌عنوان مثال می‌توانید از برنامه‌هایی که قبلاً نوشته‌اید استفاده کنید و سعی کنید آن‌ها را به صورت GUI تبدیل کنید و نمایش دهید.

نظرات، پیشنهادات و انتقادات خود را برای بهتر شدن محتوای مطالب با ما در میان بگذارید....

ترجمه شده توسط تیم الکترونیک صنعت بازار | منبع: سایت raspberrypi.org