

J2	BPE 8: Entwicklung von mobilen Applikationen Informationsmaterial	Informatik
----	---	------------

L3.1 Sounddateien, Bilder und einheitliche Stile

Die Einbindung von Multimedia-Dateien wie Bildern und Sounds in Python-Projekten mit Kivy ermöglicht die einfache Erstellung interaktiver und visueller Anwendungen. Mithilfe von Kivy-Modulen wie **Image** für Bilder und **SoundLoader** für Audio lässt sich Multimedia nahtlos in die Benutzeroberfläche integrieren.

3.1.1 Einbinden von Sounds

SoundLoader ist eine Klasse in Kivy, die verwendet wird, um Audiodateien zu laden und zu verwalten. Es unterstützt verschiedene Audioformate wie *mp3* und *wav*.

Eine ausführliche Dokumentation der SoundLoader-Klasse ist hier zu finden:

<https://kivy.org/doc/stable/api-kivy.core.audio.html#kivy.core.audio.SoundLoader>

```

1 from kivy.core.audio import SoundLoader
2 ...
3 b = SoundLoader.load(filename = 'sounds/mysound.mp3')
4 b.play()
```

Zeile 1: Diese Zeile importiert das SoundLoader-Modul aus kivy.core.audio.

Zeile 3: Hier wird versucht, die Audiodatei *mysound.mp3* zu laden, die sich in einem Ordner *sounds* befindet. Die *load*-Methode lädt die Audiodatei und gibt ein Sound-Objekt zurück, das in der Variablen *b* gespeichert wird. Falls die Datei erfolgreich geladen wurde, kann das Sound-Objekt Methoden wie *play*, *stop*, *seek*, usw. aufrufen, um die Audiowiedergabe zu steuern. Wenn die Datei nicht gefunden wird oder das Laden fehlschlägt, ist der Rückgabewert *None*.

Zeile 4: Diese Zeile startet die Wiedergabe des geladenen Sounds. Sie ruft die *play*-Methode des Sound-Objekts auf. Falls der Inhalt des Sound-Objekts *None* ist (z. B. wenn die Datei nicht existiert oder nicht geladen werden konnte), führt der Aufruf *b.play()* zu einem Fehler.

3.1.2 Ausstattung bestimmter Widgets mit einem einheitlichen Stil

Sobald mehrere Widgets des gleichen Typs in einer graphischen Benutzeroberfläche eingesetzt werden, die darüber hinaus auch über den gleichen Stil (z. B. eine einheitliche Schriftgröße des Textes) verfügen sollen, bietet es sich an, einen speziellen Widget-Typ hierfür zu definieren.

```

1 <MyButton@Button>
2     font_size: 32
3
4 MyButton:
5     text: "Vogel"

```

main.kv

Zeile 1: Zunächst wird mit MyButton ein Alias (oder eine Subklasse) des Kivy-Widgets Button definiert. Alle MyButton-Widgets, die später verwendet werden, erben automatisch die hier definierten Eigenschaften.

Zeile 2: Hier werden die zu erbenden Eigenschaften definiert. In diesem Fall wird festgelegt, dass die Schriftgröße aller MyButton-Instanzen standardmäßig 32 ist.

Zeile 4: Es wird eine Instanz von MyButton mit den oben definierten Standardattributen erstellt.

Zeile 5: Der soeben erstellten Instanz wird ein individueller Text zugewiesen.

3.1.3 Einbinden von Bildern

Image ist eine Klasse in Kivy, die verwendet wird, um Bilddateien zu laden und anzuzeigen. Sie unterstützt verschiedene Bildformate wie *.png* und *.jpg*.

Eine ausführliche Dokumentation der Image-Klasse ist hier zu finden: <https://kivy.org/doc/stable/api-kivy.uix.image.html>

```

1 Image:
2     source: 'images/vogel.png'
3     center_x: self.parent.center_x
4     center_y: self.parent.center_y
5     fit_mode: 'contain'
6     size: root.width*0.25, root.height*0.25

```

main.kv

Zeile 1: Ein Image-Widget wird angelegt.

Zeile 2: Das Attribut source gibt den Pfad zur Bilddatei an, die angezeigt werden soll. In diesem Fall wird das Bild *vogel.png* aus dem Ordner *images* geladen und im *Image*-Widget dargestellt.

Zeile 3: Hier wird die horizontale Mitte (*center_x*) des Bildes so ausgerichtet, dass sie mit der horizontalen Mitte des übergeordneten Widgets (*parent*) übereinstimmt. Dadurch wird das Bild horizontal im Zentrum des Containers positioniert.

J2	BPE 8: Entwicklung von mobilen Applikationen Informationsmaterial	Informatik
----	---	------------

Zeile 4: Dies richtet die vertikale Mitte (*center_y*) des Bildes an der vertikalen Mitte des übergeordneten Widgets aus, wodurch das Bild auch vertikal zentriert wird.

Zeile 5: Der Modus *contain* des Attributs *fit_mode* sorgt dafür, dass das Bild so skaliert wird, dass es vollständig im Widget angezeigt wird, ohne das Seitenverhältnis zu ändern. Es wird so vergrößert oder verkleinert, dass es in den verfügbaren Platz passt.

Zeile 6: Diese Zeile setzt die Größe des Bildes auf 25 % der Breite und 25 % der Höhe des Root-Widgets (meistens das übergeordnete Layout oder die Anwendung).