

# Skript zur Entwicklung einer grafischen Oberfläche in MatLab

---

*Sebastian Zirfaß ©2011*

1. Oberfläche erzeugen
2. Oberfläche mit **Pushbutton** erzeugen
  - a. Was ist ein Pushbutton
  - b. Programmcode und Erläuterung
3. Oberfläche mit **statischem/editierbarem Textfeld** erzeugen.
  - a. Programmcode und Erläuterung
4. Oberfläche mit **Slider**
  - a. Was ist ein Slider
  - b. Programmcode und Erläuterung

## 1. Oberfläche erzeugen

```
% Anwählen einer Oberfläche  
  
figure (x) % x kann einen Wert von 1 bis 2147483646  
           annehmen
```

## 2. Oberfläche mit Pushbutton erzeugen

### a. Was ist ein Pushbutton?

Ein Pushbutton führt eine ihm zugewiesenen Aktion/Funktion durch Betätigung mit der linken Maustaste aus.

### b. Programmcode und Erläuterung

```
figure (1)  
  
Button1 = uicontrol(gcf, ...  
    'Style', 'push', ...  
    'Position', [10 120 120 25], ...  
    'String', 'Push', ...  
    'Callback', '');
```

Der Befehl „figure ()“ kann in diesem Fall weggelassen werden, da durch den Befehl gcf (get current figure) eine Oberfläche erzeugt wird. „gcf“ kann auch durch den „figure ()“-Befehl ersetzt werden, sollte an eine bestimmte Oberfläche ansprechen.

„Style“ legt die Art des Controls fest das erzeugt werden soll. In diesem Fall handelt es sich um einen Pushbutton. Dafür wird der Befehl „push“ genutzt.

„Position“ legt die Position und Größe des Pushbuttons fest. Die Koordinaten werden in folgender Reihenfolge gelesen [ x1 y1 x2 y2 ]. x1 legt den Abstand in Pixeln vom linken Fensterrand fest. y1 legt den Abstand vom unteren Fensterrand fest. x2 und y2 legen die Größe des Pushbuttons in Pixeln fest.

„String“ ist der Befehl für die Beschriftung des Buttons. In dem hier aufgeführten Beispiel steht ‚Push‘ auf dem Button.

Mit dem Befehl „Callback“ legt man fest was passieren soll wenn man den Button betätigt. Mit „...“ kann man den Programmcode in der nächsten Zeile fortsetzen, sodass der Code etwas übersichtlicher und strukturierter dargestellt werden kann. Dies sollte unbedingt genutzt werden, um die Übersichtlichkeit des Programmcodes zu erhalten.

### **3. Oberfläche mit statischem/editierbarem Textfeld**

#### **a. Programmcode und Erläuterung**

```
textvar = uicontrol(gcf, ...  
    'Style', 'text/edit', ...  
    'Position', [10 90 120 25], ...  
    'String', 'Hallo');
```

Der Grundsätzliche Aufbau ist identisch zu dem des „Pushbuttons“. Möchte man ein statisches Textfeld erzeugen muss man lediglich den Befehl „text“ nutzen statt „push“. Für ein editierbares Textfeld nutzt man den Befehl „edit“.

**Im Folgenden werden nur neue Befehle erläutert, da der grundsätzliche Aufbau nun verstanden sein sollte.**

### **4. Oberfläche mit Slider**

#### **a. Was ist ein Slider?**

Ein Slider ist ein Schieberegler dem ein bestimmter Zahlenbereich (z.B. von 0 bis 20) zugewiesen werden kann. Die Position des Sliders bestimmt dann welchen Wert das Control annimmt.

#### **b. Programmcode und Erläuterung**

```
slidervar = uicontrol (gcf,...  
    'Style', 'slider',...  
    'Position', [120 120 120 40],...  
    'Min', (0), ...  
    'Max', (20), ...  
    'SliderStep', [0.05 0.1], ...  
    'Callback', 'x = sliderpos(slidervar)');
```

Zum aufrufen des Sliders verwendet man unter ‚Style‘ den Befehl ‚slider‘. Um den Wertebereich für den Slider festzulegen werden die Befehle ‚Min‘ und ‚Max‘ verwendet. Standardmäßig hat ein Slider den Bereich 0 bis 1. Mit ‚SliderStep‘ legt man die Schrittweite (in % des Wertebereichs) und die Größe des Sliderschiebeelements fest. In unserem Beispiel ist die Schrittweite 5% (0.05) so dass wir 20 Einzelschritte erhalten. Die Größe des Schiebeelements liegt hier bei 0.1. Dieser Wert hat aber lediglich optische Auswirkungen.

Um die Position des Sliders auszulesen, dient folgende selbst erstellte Funktion ‚sliderpos‘:

```
function x = sliderpos(slidervar)
    x = get(slidervar, 'Value');
end
```

Diese Funktion ist im Callback des Sliders aufzurufen. Die Position des Sliders wird dann der globalen Variable x zugewiesen. Slidervar ist der Zeiger der auf den Slider zeigt. Mit der Funktion `get(slidervar, 'Value')` wird der aktuelle Wert/Position ('Value') des Sliders der Variablen x zugewiesen.