



Úvod do programovacích stylů

Knihovna `micro_widget` verze 2

verze z 10. října 2023

Knihovna se nalézá v souboru `micro_widget.py`. Soubor musí být ve stejném adresáři jako program, který jej má používat. Pro použití stačí importovat modul:

```
import micro_widget as mw
```

Knihovna se nejlépe používá v prostředí IDLE. Pro použití jinde si přečtěte závěrečnou sekci.

Následuje popis procedur rozdělený podle prvků, s kterými procedury pracují.

Okno (`window`)

```
mw.display_window() => window
```

window: grafické okno

Zobrazí nové grafické okno. První zobrazené okno je hlavní, ostatní jsou vedlejší. Při zavření hlavního okna dojde k zavření všech vedlejších oken. Počátek soustavy souřadnic je v levo nahoře. Osa x roste směrem doprava a osa y dolů. Jednotkou vzdálenosti je jeden pixel. Výchozí rozměry okna jsou 297 na 210 pixelů.

```
mw.destroy_window(window) => None
```

window: grafické okno

Zavře grafické okno.

```
mw.is_window(value) => boolean
```

value: libovolná hodnota

boolean: logická hodnota

Rozhodne, zda je hodnota grafickým oknem.

```
mw.main_loop(main_window) => None
```

main_window: hlavní okno

Předá řízení hlavnímu oknu. Nutné volat na závěr programu mimo prostředí IDLE.
Viz poslední sekce.

Ovládací prvek (*widget*)

Následující procedury lze použít na libovolné ovládací prvky.

```
mw.destroy_widget(widget) => None
```

widget: ovládací prvek

Odstraní prvek z okna.

```
mw.get_widget_window(widget) => window
```

widget: ovládací prvek

window: grafické okno

Vrátí okno, kde se prvek nachází.

```
mw.get_widget_x(widget) => x
```

widget: ovládací prvek

x: číslo

```
mw.get_widget_y(widget) => y
```

widget: ovládací prvek

y: číslo

Vrací *x*ovou a *y*ovou souřadnici prvku.

```
mw.set_widget_x(widget, x) => None
```

widget: ovládací prvek

x: číslo

```
mw.set_widget_y(widget, y) => None
```

widget: ovládací prvek

y: číslo

Nastaví *x*ovou a *y*ovou souřadnici prvku.

Následuje popis jednotlivých ovladacích prvků.

Popisek (**label**)

```
mw.make_label(window) => label
```

window: okno

label: popisek

Vloží nový popisek do okna.

```
mw.is_label(value) => boolean
```

value: libovolná hodnota

boolean: logická hodnota

Rozhodne, zda je hodnota popiskem.

Položka **text** určuje text popisku. Výchozí hodnotou je prázdný řetězec "".

```
mw.get_label_text(label) => text  
mw.set_label_text(label, text) => None
```

label: popisek

text: řetězec

Získání a nastavení textu popisku.

Tlačítko (**button**)

```
mw.make_button(window) => button
```

window: okno

button: tlačítko

Vloží nové tlačítko do okna.

```
mw.is_button(value) => boolean
```

value: libovolná hodnota

boolean: logická hodnota

Rozhodne, zda je hodnota tlačítkem.

Položka `text` určuje text tlačítka. Výchozí hodnotou je prázdný řetězec `""`.

```
mw.get_button_text(button) => text  
mw.set_button_text(button, text) => None
```

button: tlačítko

text: řetězec

Získání a nastavení textu tlačítka.

Položka `command` určuje příkaz tlačítka. Příkaz je procedura bez parametrů, která se zavolá při kliku uživatele na tlačítko. Výchozí hodnotou je procedura s prázdným tělem; při zavolání nic neudělá.

```
mw.get_button_command(button) => command  
mw.set_button_command(button, command) => None
```

button: tlačítko

command: procedura bez parametrů

Získání a nastavení příkazu tlačítka.

Textové pole (`entry`)

```
mw.make_entry(window) => entry
```

window: okno

entry: textové pole

Vloží nové textové pole do okna.

```
mw.is_entry(value) => boolean
```

value: libovolná hodnota

boolean: logická hodnota

Rozhodne, zda je hodnota textovým polem.

Položka `text` určuje text textového pole. Výchozí hodnotou je prázdný řetězec `""`. Uživatel může text měnit.

```
mw.get_entry_text(entry) => text  
mw.set_entry_text(entry, text) => None
```

entry: textové pole
text: řetězec

Získání a nastavení textu textového pole.

Položka `command` určuje příkaz textového pole. Příkaz je procedura bez parametrů, která se zavolá při každé změně textu textového pole. A to jak uživatelem, tak programem. Výchozí hodnotou je procedura s prázdným tělem.

```
mw.get_entry_command(entry) => command  
mw.set_entry_command(entry, command) => None
```

entry: textové pole
command: procedura bez parametrů

Získání a nastavení příkazu textového pole.

Zaškrťovací pole (`checkbox`)

```
mw.make_checkbox(window) => checkbox
```

window: okno
checkbox: zaškrťovací pole

Vloží nové zaškrťovací pole do okna.

```
mw.is_checkbox(value) => boolean
```

value: libovolná hodnota
boolean: logická hodnota

Rozhodne, zda je hodnota zaškrťovacím polem.

Položka `value` určuje, zda je zaškrťovací pole vybrané. Hodnotou položky je logická hodnota. Výchozí hodnotou je `False`. Uživatel může hodnotu měnit.

```
mw.get_checkbox_value(checkbox) => value  
mw.set_checkbox_value(checkbox, value) => None
```

checkbox: zaškrťovací pole

value: logická hodnota

Získání a nastavení hodnoty zaškrťovacího pole.

```
mw.is_checkbox_selected(checkbox) => boolean
```

checkbox: zaškrťovací pole

boolean: logická hodnota

Rozhodne, zda je zaškrťovací pole vybrané.

Položka `command` určuje příkaz zaškrťovacího pole. Příkaz je procedura bez parametrů, která se zavolá při každé změně hodnoty zaškrťovacího pole. A to jak uživatelem, tak programem. Výchozí hodnotou je procedura s prázdným tělem.

```
mw.get_checkbox_command(checkbox) => command  
mw.set_checkbox_command(checkbox, command) => None
```

checkbox: zaškrťovací pole

command: procedura bez parametrů

Získání a nastavení příkazu zaškrťovacího pole.

Přepínač (`radiobutton`)

```
mw.make_radiobutton(window) => radiobutton
```

window: okno

radiobutton: zaškrťovací pole

Vloží nový přepínač do okna.

```
mw.is_radiobutton(value) => boolean
```

value: libovolná hodnota

boolean: logická hodnota

Rozhodne, zda je hodnota přepínačem.

Položka `value` určuje, zda je přepínač vybraný. Hodnotou položky je logická hodnota. Výchozí hodnotou je `False`. Uživatel může hodnotu měnit.

```
mw.get_radiobutton_value(radiobutton) => value  
mw.set_radiobutton_value(radiobutton, value) => None
```

radiobutton: zaškrťovací pole
value: logická hodnota

Získání a nastavení hodnoty přepínače.

```
mw.is_radiobutton_selected(radiobutton) => boolean
```

radiobutton: zaškrťovací pole
boolean: logická hodnota

Rozhodne, zda je přepínač vybraný.

Položka `command` určuje příkaz přepínače. Příkaz je procedura bez parametrů, která se zavolá při každé změně hodnoty přepínače. A to jak uživatelem, tak programem. Výchozí hodnotou je procedura s prázdným tělem.

```
mw.get_radiobutton_command(radiobutton) => command  
mw.set_radiobutton_command(radiobutton, command) => None
```

radiobutton: přepínač
command: procedura bez parametrů

Získání a nastavení příkazu přepínače.

Mimo IDLE

V jiném prostředí je na závěr potřeba zavolat proceduru `mg.main_loop` na hlavní okno. Například:

```
import micro_widget as mw  
w = mw.display_window()  
l = mw.make_label(w)  
mw.set_label_text(l, "Borůvka")  
mw.main_loop(w)
```