



Úvod do programovacích stylů

Manuál ke knihovně fmw (druhá verze)

verze z 15. listopadu 2023

Knihovnu tvoří soubory `micro_widget.py` a `fmw.py`. Soubory se musí nalézat v téže adresáři jako program, který jej používá. Pro použití stačí importovat modul:

```
from fmw import *
```

Ovládací prvky

```
label(text="", x=0, y=0) => widget
```

text: řetězec
x: celé číslo
y: celé číslo
widget: ovládací prvek

Prvek *widget* je popiskem s textem *text* na souřadnicích *x*, *y*.

```
button(text="", action=None, x=0, y=0) => widget
```

text: řetězec
action: libovolná hodnota
x: celé číslo
y: celé číslo
widget: ovládací prvek

Prvek *widget* je tlačítko s textem *text* na souřadnicích *x*, *y*. Při stisku v případě, že *action* není `None`, vyvolá událost *action*.

```
entry(text="", action=None, x=0, y=0) => widget
```

text: řetězec
action: libovolná hodnota

x: celé číslo
y: celé číslo
widget: ovládací prvek

Prvek *widget* je textovým polem s textem *text* na souřadnicích *x*, *y*. Umožňuje změnu uživatelem, právě když *action* není *None*. V případě změny uživatelem vyvolá akci [*action*, *new_text*], kde *new_text* je změněný text.

```
checkbox(value=False, action=None, x=0, y=0) => widget
```

value: logická hodnota
action: libovolná hodnota
x: celé číslo
y: celé číslo
widget: ovládací prvek

Prvek *widget* je zaškrťovacím polem, které je vybrané, právě když je *value* pravda. Umožňuje změnu uživatelem, právě když *action* není *None*. V případě změny uživatelem vyvolá akci [*action*, *new_value*], kde *new_value* je změněná hodnota.

```
radiobutton(value=False, action=None, x=0, y=0) => widget
```

value: logická hodnota
action: libovolná hodnota
x: celé číslo
y: celé číslo
widget: ovládací prvek

Prvek *widget* je přepínačem, který je vybraný, právě když je *value* pravda. Umožňuje změnu uživatelem, právě když *action* není *None*. V případě změny uživatelem vyvolá akci [*action*, *new_value*], kde *new_value* je změněná hodnota.

```
moved(widget, dx, dy) => moved_widget
```

widget: ovládací prvek
dx: celé číslo
dy: celé číslo
moved_widget: ovládací prvek

Prvek *changed_widget* je prvkem *widget* posunutým o přírůstky *dx* a *dy*.

```
action_changed(widget, change) => changed_widget
```

widget: ovládací prvek

change: funkce jednoho parametru nebo hodnota, která není funkcí

changed_widget: vládací prvek

Prvek *changed_widget* je shodný s prvkem *widget* až na to, že každá akce *action* vyvolaná prvkem *widget* se změní na *change(action)*, pokud je *change* funkce, jinak na [*change*, *action*].

```
group(widget1, widget2) => group
```

group: ovládací prvek

widget1: ovládací prvek

widget2: ovládací prvek

Ovládací prvek *group* se skládá z prvků *widget1* a *widget2*.

```
empty_widget
```

Prázdný ovládací prvek.

Zobrazení okna

```
display_window(content, init_state=None, update=None))
```

content: funkce jednoho parametru nebo ovládací prvek

init_state: libovolná hodnota

update: hodnota None nebo funkce dvou parametrů

Pokud *content* není funkce, pak se místo *content* používá funkce jednoho parametru, která vždy vrací *content*. Neboli funkce:

```
lambda state: content
```

Pokud je *update* hodnota None, pak se místo *update* používá funkce dvou parametrů, která vždy vrací druhý argument. Což je funkce:

```
lambda state, action: action
```

Aktuální *stav okna* si označíme *state*. Na začátku je *state* rovno *init_state*. Obsah okna se získá zavoláním funkce *content* na *state*. Okno tedy zobrazuje ovládací prvek:

```
content(state)
```

Po každé změně stavu okna se znovu získá jeho obsah.

Při vyvolání akce *action* ovládacím prvkem v okně, se změní stav okna *state* na výsledek volání funkce *update* na původní stav okna *state* a akci *action*. Tedy:

```
state = update(state, action)
```

```
display_window_and_loop(content, init_state=None, update=None))
```

content: funkce jednoho parametru nebo ovládací prvek

init_state: libovolná hodnota

update: hodnota *None* nebo funkce dvou parametrů

Chování funkce je stejné jako funkce `display_window` až na to, že se předá řízení oknu. Nutné použít mimo `IDLE`.