



Rychlý návod na Innovation – dokumentace

Všechny příkazy vypadají takto:
příkaz(argument1, argument2)

Tady je ukázka příkazu „write“:
write(„text“)

Proměnné jsou něco jako slovník, to znamená, že každá proměnná má hodnotu. Abychom mohli proměnnou nastavit, potřebujeme znát typy proměnných.

num (number) = Číslo (třeba 28)

text (text) = Text (třeba „Hello“, uvozovky jsou důležité!)

bit (bit) = Hodnota buď 1 nebo 0 (ano nebo ne) (třeba 1)

some (some) = Něco / neuvedeno

Proměnnou nastavíme takto:

typ: název = hodnota

num: number = 28

text: text = „Ahoj“

bit: jednaNeboNula = 1

some: něco



Pro definici some proměnné můžeme také udělat: něco.

Nebo typ nemusíme upřesňovat:

number = 28

text = „Ahoj“

jednaNeboNula = yes

něco = some

Výjimka je tedy jenom u bitu (yes nebo no)

Write

Do funkce (třeba write) můžeme buď napsat už hodnotu, nebo název proměnné:

write(„Nějaký text“)

write(10)

write(yes)

write(some)

Bit = yes

write(Bit)

Výstup:

Nějaký text

10

1

some

1



Nyní ježtě put (vstup)

Ukázka:

vstup.

put(*vstup)

write(vstup)

Všimněte si té hvězdičky, ta se používá, když se specifikuje proměnná do zapsání.

Nyní už známe funkci write a jak to nějak funguje. Tady je seznam většiny funkcí:

write(text)	Vypíše argument 1
put(*out)	Získá vstup do argumentu 1
wait()	Počká, dokud uživatel nezmáčkne enter
vars()	Vypíše proměnné
put(text)	Vypíše argument 1 stále na jeden řádek

Možná vás napadlo, jak nastavit jednu proměnnou na druhou, tady je ukázka:

A = 10 Nastaví A na 10

B. Definuje proměnnou B

B <- A Nastaví proměnnou B na A



Jak zapsat komentář:

Když cheme něco okomentovat, tak stačí napsat před komentář „//“

Ukázka:

```
// Vypíše „Ahoj“  
write(„Ahoj“)
```

Math:

Možná vás napadlo, jestli je možné si udělat nějakou svojí matematickou funkci. A na to je přesně „math“.

Ukázka:

```
// Definujeme matematickou funkci „obvodObdelniku“  
math: obvodObdelniku = 2 * (A + B)  
// Vypočítáme obvod obdelníku a = 5, b = 10  
obvodObdelniku(5, 10)
```

```
// Výstup se zapíše do proměnné „returned“ (30)  
write(returned)
```

If:

Jednoduše, pokud budeme chtít něco zkontrolovat, použijeme if.



Ukázka:

```
// Nastavíme proměnnou A na 20
```

```
A = 20
```

```
// Zkontrolujeme ji, pokud je větší než 15
```

```
If(A > 15) (
```

```
    // Tak vypíšeme, že A je větší než 15
```

```
    write(„A je vetsi nez 15“)
```

```
)
```

WinFuns:

WinFuns jsou jednoduše funkce, které jsou dostupné jenom pro Windows. Tady je seznam funkcí:

win:execute(text: command)	Vyvolá příkaz do shell
win:boxInfo(text: Text, text: title)	Ukáže message box s informací
win:boxError(text: Text, text: title)	Ukáže message box s chybou



<code>win:boxContinue(text: Text, text: title)</code>	Ukáže message box s otázkou, jestli uživatel chce pokračovat
<code>win:click(int: x, int: y)</code>	Klikne na pozici x, y
<code>win:newRectangle(int x1, int y1, int x2, int y2, int r, int g, int b)</code>	Vytvoří nový obdelník na x1, y1 do x2, y2, s barvou r, g, b

Space:

Pomocí „space“ můžete udělat takovou jakoby „strukturu“.

Ukázka:

```
space test {  
    a = 10  
    math: b = A + B  
}  
test:b(2, 2)  
write(returned)  
// 4  
write(test:a)  
// 10
```

Space můžeme taky definovat takto:



```
nazev = „test“  
space(„test“) {  
    // ...  
}
```

Program má ještě další funkce, tento návod budu ještě rozšiřovat.