

1. Opis języka

Skryptowy język programowania, służący np. do zarządzania plikami na komputerze. Ten język będzie łatwiejszą wersją shell/bash, stosowanie której na co dzień nie będzie wywoływać trudności u użytkownika.

Język jest dynamicznie typowany (jak Python). Kod źródłowy jest wykonywany od pierwszej instrukcji zapisanej w pliku, robiąc analogię do C/C++ cały plik jest funkcją `main()`. Żeby używać moduły/programy stosowana jest funkcja wbudowana `execute`. Czyli jeżeli chcemy stworzyć funkcję np. do obliczania silni to tworzymy nowy plik np. `silnia.wl` i wywołujemy ją za pomocą `execute`.

Skrypty języka Wordlify są tłumaczone na kod źródłowy Pythona

2. Składnia języka

- Przypomina język Pascal z blokami kodu ograniczonymi słówkami **begin** oraz **end**
- Rozróżnia wielkość liter
- Instrukcje nie kończą się średnikami, dlatego muszą być zapisywane w oddzielnych liniach

2.1 Operatory

- `+`, `-`, `*`, `/`, `()`
- `^` - potęgowanie
- `==`, `!=`, `<`, `>`, `<=`, `>=` - porównywanie
- `not`, `and`, `or` - operatory logiczne

2.2 Słowa kluczowe

- `if` - instrukcja warunkowa, musi mieć ciało które zaczyna się od `then` i kończy się `end` oraz warunek, który musi być zawarty między `if` oraz `then`.
- `else` - po tym słowie występuje blok instrukcji wykonywany, gdy warunek nie jest spełniony
- `then` - służy do rozpoczęcia ciała instrukcji warunkowej.
- `end` - służy do zakończenia ciała instrukcji warunkowej lub pętli.
- `while` - pętla wykonywana dopóki warunek będzie prawdziwy
- `for` - pętla `for`, podobnie jak w języku C/C++ może mieć przypisanie, warunek, oraz zwiększanie licznika (`for a=1; a<10; a=a+1`). Ciało pętli musi zaczynać się od słowa kluczowego `do` i kończyć się `end`.
- `foreach` - pętla `foreach` służy do iterowania po elementach listy lub np. po plikach w danym katalogu
- `do` - służy do rozpoczęcia ciała pętli

2.3 Funkcje wbudowane

- exist <nazwa_pliku> - sprawdza czy podany plik istnieje, zwraca false lub true
- rename <old_filename> <new_filename> - zmienia nazwę pliku
- remove <filename> - usuwa plik
- move <path_to_file> <new_path> - przenosi plik do innego miejsca na dysku
- copy <path_to_file> <new_path> - kopiowanie pliku
- download <link> <filename> - pobieranie pliku z internetu i zapisywanie pod nazwą <filename>
- read <file> - wczytywanie zawartości pliku, zwraca string
- write <filename> <content> - nadpisuje plik <filename> zawartością <content>
- wait <time> - wstrzymuje wykonywanie programu na <time> sekund
- execute <filename> <arg1> <arg2> <arg3> - wykonuje plik i przekazuje mu argumenty
- time - obecny czas w sekundach od 1.01.1970
- getFiles <path> - zwraca listę nazw plików zawartych w katalogu <path>
- dateModified <path> - data modyfikacji pliku w sekundach od 1.01.1970
- size <path> - rozmiar pliku/katalogu w bajtach
- file - nazwa pliku, który jest aktualnie wykonywany
- folder - nazwa katalogu, gdzie jest plik aktualnie wykonywany
- args - zwraca listę argumentów przekazanych do programu
- exit <exit_code> - zakończenie działania programu, zwraca <exit_code>

3. Konstrukcje

Instrukcja warunkowa

```
if warunek then
    instrukcje
else if warunek then
    instrukcje
else
    instrukcje
end
```

Pętla while

```
while warunek do
    instrukcje
end
```

Pętla for

```
for myVariable = 0; myVariable < 10; myVariable++ do
    instrukcje
end
```

Pętla foreach

```
foreach element in lista do
    instrukcje
end
```

4. Przykładowe programy

Program do usuwania plików w katalogu, które były zmienione do 5 minut temu:

```
foreach file_in_dir in folder do
    if dateModified file_in_dir + 5*60 > time then
        remove file_in_dir
    end
end
```

Program do pobierania plików ze strony internetowej

```
if exist "repo.zip" then remove "repo.zip" end #usuwamy stary plik
download "https://github.com/antlr/antlr4/archive/master.zip" "repo.zip"
rename "repo.zip" "master.zip"
wait 1
move "master.zip" "C:\Program Files (x86)\"
```