



Przejście na skróty Python => C++



Python 3

C++

◆ Przykładowe “lekkie” środowisko online

[Python Interpreter @ Programiz.com](#)

[C++ Compiler @ Programiz.com](#)

◆ Początek wykonywania programu

Pierwsze polecenie. Nie musi być funkcji.

Zaczyna od funkcji `main`

◆ Pogrupowanie poleceń

W bibliotekach, załączanych przez:

`import nazwa`

W bibliotekach, załączanych przez:

`#include <nazwa>`

Wszystkie polecenia są w przestrzeni nazw `std`. Załączamy ją tak:

`using namespace std;`

◆ Rama funkcji

```
def Nazwa ( x ):  
    x *= 5  
    return x - 3
```

↑
Wcięcia konieczne, linie zlicowane

↵ typ zwracany

```
int funkcja (int X) {  
    x *= 5 ;           ← średnik  
    return x - 3 ;     ← średnik  
} ↵ wcięcia niekonieczne, ale zalecane
```

◆ Wypisanie `Hello World` na ekranie

`print ("Hello world")`

`cout << "Hello world" << endl ;`
↑
(jeśli chcemy Enter)

◆ Tworzenie zmiennych

Przy przypisaniu, komputer sam ustali typ:
`var = 123` ← typem będzie `int`

Wpierw deklaracja zmiennej z typem:
`int var ;`
Deklaracja może mieć przypisanie:
`int var = 123 ;`

◆ Garść prostych typów zmiennych

`int` ← l. całkowita (wiele bajtów)
`float` ← l. zmiennoprzecinkowa (8 bajtów)
`string` ← napis

`int` ← l. całkowita (zwykle 8 bajtów)
`float, double` ← l. zmiennoprzecinkowa (4, 8 bajtów)
`char, string` ← 1 znak, cały napis

◆ Wypisanie kilku rzeczy na raz

`print ("var = ", var)`

`cout << "var = " << var << endl;`



Przejdźcie na skróty Python ⇒ C++



Python 3

C++

◆ Wczytanie z klawiatury do zmiennej

```
var = input ("Podaj zmienna:")
```

```
cout << "Podaj zmienna: " ;  
cin >> var ;
```

◆ Przypisanie wielokrotne

Możliwe, np często stosuje się:
`a, b = b, a`

Niemożliwe. Nb. do zamiany stosuje się:
`swap (a, b) ;`

◆ Wyrażenie warunkowe

```
if warunek :  
    ...  
elif warunek :  
    ...  
else:  
    ...
```

```
if ( warunek ) {  
    ... ;  
} else if ( warunek ) {  
    ... ;  
} else {  
    ... ;  
}
```

◆ Pętla while

```
while warunek:  
    ...
```

```
while ( warunek ) {  
    ... ;  
}
```

◆ Pętla z inicjalizacją, warunkiem i ostatnim działaniem kroku:

```
i = 0  
while i < 5:  
    ...  
    i += 1
```

```
for (int i = 0; i < 5; i++ ) {  
    ... ;  
}
```

◆ Proste tablice

```
MyList = [1, 2, 3, 4]  
MyNumpy = np.linspace (1, 4)
```

```
int MyArray[] = {1, 2, 3, 4} ;  
valarray<int> MyArr = {1, 2, 3};
```

◆ Pętla zakresowa

```
for x in MyList:  
    print (x)
```

```
for (int x : MyArray) {  
    cout << x << endl;  
}
```