# Początek kodu, zmienne, tablice

## 1. Wstęp

Zanim zaczniesz czytać dalszą część poradnika musisz posiadać podstawową wiedzę na temat programowania – zmienne, pętle czy funkcje – te pojęcia powinny być Ci znane. Jeżeli udało Ci się przebrnąć już kiedyś przez Pascala czy podstawy C/C++, to mam nadzieję, że bez problemu odnajdziesz się w kolejnych rozdziałach mojego poradnika do PHP. Bardzo często będę używał porównań właśnie do języka C/C++ w celu zobrazowania różnic.

## 2. Początek kodu

W C++ po prostu piszemy kod. W pliku *.cpp załączamy interesujące nas biblioteki, tworzymy funkcje, inicjujemy zmienne itd. Wszystkie linie wpisane przez nas (oczywiście oprócz tych, które oznaczyliśmy jako komentarze) zostaną potraktowane przez kompilator jako kod, który musi odpowiednio przetłumaczyć na język maszynowy. Ponieważ często (choć nie jest to zasadą) język PHP przeplata się np. z HTML-em, nasz serwer, a w zasadzie interpreter PHP musi wiedzieć, który kod ma być wykonywany po stronie serwera, a który ma zwrócić do klienta. Aby poinformować nasz interpreter w którym miejscu ma zacząć „tłumaczyć”, umieszczamy nasz kod w znacznikach:

```
<?php
      //nasz kod php
?>
```

Przy takim oznaczeniu interpreter wie, że ten kod ma zostać wykonany po stronie serwera i nie może wyświetlić go użytkownikowi.
Jeżeli nasz kod PHP przeplata się z HTML-em:

```
<p><?php zwroc_paragraf(); ?></p>
```

znacznik ***<?php*** musi być bezwzględnie zamknięty. W innym przypadku nasz serwer będzie próbował wykonać kod, który jest kodem HTML jako skrypt PHP co będzie powodować błędy. Wyjątkowym przypadkiem jest plik, który w całości zawiera kod PHP. W takim przypadku nie powinniśmy go zamykać.

## 3. Zmienne

Już na samym początku widzimy zasadniczą różnicę między C/C++ a PHP. W tych pierwszych musimy najpierw zadeklarować zmienną oraz określić jej typ np.

```
int a = 10;       //zmienna typu całkowitego
char b = 'b';     //zmienna typu znakowego
float c = 10.23; //zmienna typu zmiennoprzecinkowego
```

W przypadku PHP wygląda to inaczej. Aby interpreter PHP rozpoznał, że chcemy użyć zmiennej, a nie np. funkcji, przed jej nazwą umieszczamy znak ***$***. Zmiennych nie musimy deklarować przed ich użyciem. Możemy je tworzyć „w locie”. Jeżeli chcemy wynik jakiejś funkcji przypisać do zmiennej wpisujemy:

```
$zmienna = funkcja();
```

Deklaracja tych samych zmiennych, których przykład umieściłem powyżej, w języku PHP wygląda tak:

```
$a = 10;     //zmienna typu całkowitego
$b = 'b';    //zmienna typu znakowego
$c = 10.23; //zmienna typu zmiennoprzecinkowego
```

Istnieje jeszcze jedna właściwość zmiennych w PHP o której warto wspomnieć. Na potrzeby jakiejś funkcji potrzebujemy stworzyć zmienną chwilową ***$tmp***. Typ zmiennej określamy dopiero w momencie przypisania do niej wartości – nie musi on być jednak taki sam w całym naszym programie. Do zmiennej ***$tmp*** możemy najpierw przypisać liczbę całkowitą, później zmiennoprzecinkową, a kilka linii dalej ciąg znaków czy nawet tablicę.

## 4. Tablice

A jeżeli już przy tablicach jesteśmy to też mają one bardzo ciekawe własności. W PHP używamy tablic asocjacyjnych – nie musimy ich indeksować liczbami całkowitymi. Typową, jednowymiarową tablicą asocjacyjną jest np. kwestionariusz osobowy:

| Imię | Jan |
|---|---|
| Nazwisko | Nowak |
| PESEL | 80123112345 |

Taką tablicę możemy stworzyć w następujący sposób:

```
$osoba = array('imię' => 'Jan',
               'nazwisko' => 'Nowak',
               'PESEL' => 80123112345);
```

W przypadku języka C++, mogliśmy w bardzo prosty sposób poruszać się po zwykłej, indeksowanej tablicy np. za pomocą pętli ***for***. Inicjalizowaliśmy licznik który wskazywał nam na konkretny element tablicy i w tej właśnie pętli wykonywaliśmy interesujące nas operacje np.

```
for (int i = 0; i < 10; i++) {
      cout << tablica[i] << endl;
}
```

W przypadku PHP, jeżeli tablica nie jest indeksowana w ten sposób, musimy wykonać to inaczej. Tutaj przydatną funkcją, jest funkcja ***foreach***.

```
foreach ($osoba as $wartosc) {
      echo $wartosc . '<br>';
}
```

lub

```
foreach ($osoba as $klucz => $wartosc) {
      echo $klucz . ' - ' . $wartosc . '<br>';
}
```

Na przykładzie tablicy ***$osoba***, w której umieściliśmy dane Nowaka, pierwszy ***foreach*** wypisze na ekranie wszystkie wartości wpisane do tablicy oraz zakończy znakiem przerwania linii:

```
Jan
Nowak
80123112345
```

Dzięki zastosowaniu konstrukcji, w której używamy zmiennej ***$klucz***, możemy również dostać się do nazw kolejnych pól w tablicy. Efektem wywołania drugiego ***foreach'a*** będzie:

```
imię - Jan
nazwisko - Nowak
PESEL - 80123112345
```

## 5. Podsumowanie

Po przeczytaniu tego poradnika powinieneś potrafić zagnieżdżać kod PHP pomiędzy znacznikami HTML, stworzyć szkielet pliku w całości napisanego w PHP, używać zmiennych oraz zmiennych tablicowych, a także poruszać się po tablicach za pomocą pętli ***foreach.***