

## Lekcja 44. (kl. II. PR)

### Temat: Funkcje w języku PHP, czyli jak wzbogacić naszą stronę.

#### Cele lekcji:

– skonstruowanie własnej funkcji w języku PHP

#### Uczeń:

– poznaje i używa niektórych wbudowanych funkcji PHP;  
– układa i używa własnej funkcji w skrypcie PHP.

Podręcznik str. 268 – Informatyka. 2. PR, Operon

#### **Przebieg lekcji:**

1. Zapoznanie się z celami lekcji.
2. Pojęcie funkcji w PHP.
3. Funkcje własne i wbudowane.
4. Ćwiczenia praktyczne – stosowanie funkcji w PHP.

**Podręcznik str. 271.**

#### **Material:**

## Funkcje

Funkcja jest kawałkiem kodu, który można wywołać podając jej nazwę. Już wcześniej podane zostało kilka funkcji dostarczanych przez PHP, takich jak `array()`, `each()`, `list()`, `explode()` czy `implode()`. Jak można zauważyć, funkcje rozpoznaje się po nawiasach występujących po ich nazwie. W nawiasach tych (jeśli funkcja tego wymaga) podaje się parametry dla danej funkcji. Niektóre funkcje zwracają jakąś wartość, a więc wywołanie takiej funkcji można wstawić jak dowolne inne wyrażenie do listy parametrów innej funkcji czy np. do obliczenia.

Programista może też tworzyć własne funkcje.

#### **Przykład 6.1. Definicja funkcji**

```
<?php
function suma($parametr1 = 0, $parametr2 = 0)
{
    $wartosc = $parametr1 + $parametr2;
    return $wartosc;
}
?>
```

Jak widać, definiowana jest funkcja o nazwie `suma`. Służy ona do obliczania sumy dwóch wyrażeń podawanych jako parametry (kolejne parametry oddziela się przecinkiem). W przypadku pominięcia któregoś z parametrów przypisana mu będzie wartość 0 – dzieje się tak dzięki zaznaczeniu `” = 0”` po określeniu nazwy parametru w definicji funkcji (nie jest to konieczne). Z parametrów funkcji można korzystać jak z dowolnej innej zmiennej, ale oczywiście tylko wewnątrz tej funkcji. Wewnątrz funkcji dostępne są tylko zmienne w niej definiowane i parametry. Można się dostać też do innych zmiennych, ale o tym w punkcie „Zasięg zmiennych”. Do określania co jest zwracane przez funkcję służy instrukcja `return`. Może ona być wywołana w dowolnym miejscu funkcji – po dojściu do tej instrukcji kończy się wykonanie funkcji.

Funkcja może nie zwracać żadnych wartości. Taka funkcja przydaje się jeśli często powtarzany jest jakiś fragment kodu. Taka funkcja kończy swoje wykonanie po dojściu do końca jej definicji lub instrukcji `return` (tym razem wywoływanej bez parametru, ponieważ funkcja nie zwraca żadnych wartości).

Funkcje w PHP nie mogą być ponownie definiowane. Mogą być definiowane w zasadzie w dowolnym miejscu kodu. Wewnątrz funkcji znajdować się może dowolny, prawidłowy pod względem składniowym kod – nawet definicje innych funkcji czy klas.

---

## PHP funkcje – czym jest funkcja?

Funkcje w PHP są częściowo podobne do funkcji matematycznych. Mogą przyjmować argumenty oraz zwracać wartości. Nie jest to jednak wymogiem. Funkcją nazywamy napisany przez nas kod, zamknięty w nawiasy klamrowe, poprzedzony słowem kluczowym `function` oraz unikatową nazwą.

Intencją tworzenia funkcji jest ich ponowne użycie. Żeby nie kopiować i nie wklejać tego samego kodu w kilku miejscach w pliku, piszemy go raz w postaci funkcji. Wtedy możemy go wykorzystywać podając jedynie jej nazwę. Tak samo łatwiej nam poprawić błąd w jednym miejscu (jeśli znajdziemy jakiś wewnątrz funkcji) niż poprawiać go w każdym miejscu skopiowanego skryptu.

Nazwy funkcji nie mogą się powtarzać. Nie można również używać nazw istniejących już poleceń, takich jak [while](#), `if`, `echo` itp.

Funkcje możemy podzielić na cztery kategorie:

- bezargumentowa, nie zwracająca wartości,
- bezargumentowa, zwracająca pewną wartość,
- przyjmująca argumenty, nie zwracająca wartości,
- przyjmująca argumenty oraz zwracająca wartość.

# Funkcja bezargumentowa, nie zwracająca wartości

Na tej lekcji zajmiemy się pierwszym typem funkcji. Zobaczmy poniższy listing:

```
<?php

function wyswietl_powitanie() // deklaracja funkcji
{
    echo "Witam serdecznie!"; // ciało funkcji, czyli
    echo "Proszę się zarejestrować."; // instrukcje do wykonania
}

?>
```

Przeanalizujmy napisany kod. Przedstawia funkcję, której zadaniem jest wyświetlenie na ekranie dwóch komunikatów. Jest to jedynie deklaracja. Żeby użyć tak napisanej funkcji należy ją wywołać. Wywołanie przedstawione jest poniżej:

```
<?php

// pozostały kod w pliku
wyswietl_powitanie();
// pozostały kod w pliku
wyswietl_powitanie();
wyswietl_powitanie();
// pozostały kod

?>
```

Jak widać, żeby użyć napisanej funkcji, wystarczy napisać jej nazwę. Jak każde polecenie w PHP, tak również wywołanie napisanej przez nas funkcji, musi być zakończone średnikiem. Co znaczą nawiasy między nazwą, a średnikiem? Jest to miejsce na podanie argumentów funkcji. Skoro nasza funkcja nie przyjmuje żadnych, są one puste. Nie należy jednak zapominać o ich umieszczeniu, nawet gdy funkcja jest bezargumentowa!

---

## Funkcja zwracająca wartość

Poprzednio omawialiśmy funkcje, których wykonywane zadanie zamykało się wewnątrz struktury. Znaczy to tyle, że nie miały wpływu na wykonanie dalszego kodu. Tym razem zajmiemy się funkcjami, które coś wnoszą do programu. Żeby funkcja zwróciła wartość do programu głównego, musimy umieścić zwracane wyrażenie po słowie `return`.

Zobaczmy przykład:

```
<?php

function tresc_powitania() // deklaracja funkcji
{
    return "Witam wszystkich!";
}

$powitanie = tresc_powitania();
echo $powitanie;

?>
```

Żeby zrozumieć zasadę działania funkcji zwracającej wartość, wyobraźmy sobie, że funkcja to taka zmienna, której wartość zmienia się dynamicznie. Co za tym idzie, funkcję możemy przypisać zmiennej. Dodatkowo

możemy sprawdzić, czy funkcja jest mniejsza lub większa od pewnej liczby. Oczywiście, mówiąc funkcja, mamy na myśli wartość zwracaną przez daną funkcję.

Przeanalizujmy skrypt:

```
<?php

function oblicz()
{
    $zm1 = 3;
    $zm1 += 5;
    $zm1++;
    return $zm1;
}

if (oblicz() > 5)
    echo "Funkcja zwraca wartość większą od 5";
else
    echo "Wartość zwracana przez funkcję jest mniejsza od 6";

?>
```