

# Funkcje wbudowane oraz zapis do pliku w PHP

PHP oferuje setki gotowych funkcji wbudowanych – nie trzeba więc wszystkiego tworzyć samodzielnie, wystarczy wywołać “gotowe” funkcje z odpowiednimi argumentami. No, pod warunkiem, że je znamy – właśnie dlatego taka lekcja powinna pojawić się w kursie! Znajomość wbudowanych funkcji przyspiesza pisanie kodu i pozwala w pełni korzystać z możliwości PHP oraz szybko uporać się z przeróżnymi zadaniami na egzaminie INF.03.

W części drugiej lekcji zajmiemy się także sposobami zapisu plików na dysku serwera przez skrypty PHP – tego rodzaju zadania także już pojawiały się na egzaminach INF.03.

## Generowanie liczb losowych

Generowanie liczb losowych przydaje się m.in. w quizach, grach, losowych hasłach.

Funkcja

`rand()` zwraca losową liczbę całkowitą z określonego zakresu:

```
echo rand(1, 10); // Losuje liczbę od 1 do 10
```

Funkcja

`mt_rand()` to nowocześniejsza, szybsza i dokładniejsza wersja generatora:

```
echo mt_rand(100, 999); // Losuje liczbę trzycyfrową
```

## Zaokrąglanie i liczby zmiennoprzecinkowe

Do operacji matematycznych służą funkcje:

- `round()` – zaokrągla liczbę
- `floor()` – zaokrąglenie w dół
- `ceil()` – zaokrąglenie w górę
- `abs()` – wartość bezwzględna

```
echo round(5.678); // 5.68
echo floor(5.678); // 5
echo ceil(5.678); // 6
echo abs(-10); // 10
```

## Przetwarzanie tekstu – operacje na łańcuchach znaków

PHP oferuje bogaty zestaw funkcji do pracy z tekstem:

- `strlen()` – długość łańcucha
- `strtolower()` – zamienia na małe litery
- `strtoupper()` – zamienia na wielkie litery
- `substr()` – wycina fragment łańcucha

- `str_replace()` – zamiana fragmentu tekstu

```
$tekst = "Programowanie w PHP";
echo strlen($tekst);           // 20
echo strtolower($tekst);       // programowanie w php
echo strtoupper($tekst);       // PROGRAMOWANIE W PHP
echo substr($tekst, 0, 12);     // Programowanie
echo str_replace("PHP", "HTML", $tekst); // Programowanie w HTML
```

## Operacje logiczne i porównania

Bardzo przydatne funkcje to także:

- `isset()` – sprawdza, czy zmienna istnieje i nie ma wartości null
- `empty()` – sprawdza, czy zmienna jest pusta (0, "", null, false)
- `is_numeric()` – sprawdza, czy zmienna zawiera liczbę

```
$x = "123";
if (is_numeric($x)) {
    echo "To liczba!";
}
$y = "";
if (empty($y)) {
    echo "Zmienna jest pusta!";
}
```

## Funkcje konwertujące typ danych

Do rzutowania i konwersji danych używamy m.in.:

- `intval()` – konwersja na liczbę całkowitą
- `floatval()` – konwersja na liczbę zmiennoprzecinkową
- `strval()` – konwersja na tekst (string)

```
$x = "42.5";
echo intval($x);    // 42
echo floatval($x);  // 42.5
echo strval(123);   // "123"
```

## Funkcje zapisujące pliki na dysku serwera

### Inne przydatne funkcje

Oto jeszcze kilka narzędzi, które mogą się przydać:

`uniqid()` – generuje unikalny identyfikator

```
echo uniqid(); // np. 5ebf5bfe83bcf
```

`var_dump()` – wypisuje szczegółową zawartość zmiennej (typ + wartość)

```
var_dump(123); // int(123)
var_dump("PHP"); // string(3) "PHP"
```

`explode()` – rozdziela łańcuch znaków na tablicę według separatora.

```
$tekst = "PHP,HTML,CSS,JS";
$tablica = explode(",", $tekst);
print_r($tablica);
// Array ( [0] => PHP [1] => HTML [2] => CSS [3] => JS )
```

`implode()` – łączy elementy tablicy w jeden łańcuch znaków.

```
$jezyki = ["PHP", "HTML", "CSS"];
$tekst = implode(" - ", $jezyki);
echo $tekst;
// PHP - HTML - CSS
```

`trim()` – usuwa białe znaki z początku i końca tekstu.

```
$napis = "   Witaj świecie!   ";
echo trim($napis);
// "Witaj świecie!"
```

`ucfirst()` – zamienia pierwszą literę na wielką.

```
echo ucfirst("php to super język");
// Php to super język
```

`number_format()` – formatuje liczby (np. z przecinkami i separatorami tysięcy).

```
echo number_format(1234567.891, 2, ",", " ");
// 1 234 567,89
```

`nl2br()` – zamienia znaki nowej linii (\n) na znaczniki <br>.

```
$tekst = "Witaj świecie!\nMiło Cię widzieć.";
echo nl2br($tekst);
// Witaj świecie!
// Miło Cię widzieć.
```

`str_repeat()` – powtarza łańcuch określoną liczbę razy.

```
echo str_repeat("*", 10);  
// *****
```

`min()` i `max()` – zwracają najmniejszą i największą wartość z zestawu.

```
echo min(5, 9, 2); // 2  
echo max(5, 9, 2); // 9
```

`strrev()` – odwraca tekst (np. do zabaw z palindromami lub szyfrowaniem).

```
echo strrev("Programowanie w PHP");  
// Wynik: PHP w einamargorP
```

`array_reverse()` – odwraca kolejność elementów w tablicy.

```
$tablica = [1, 2, 3];  
print_r(array_reverse($tablica));  
// Array ( [0] => 3 [1] => 2 [2] => 1 )
```

## Funkcje PHP zapisujące pliki na dysku serwera

W PHP możemy bezpośrednio zapisywać dane do plików znajdujących się na serwerze. To bardzo przydatne w wielu aplikacjach webowych – np. do tworzenia logów, zapisywania danych użytkownika, tworzenia kopii zapasowych lub eksportów danych.

Najważniejsze funkcje służące do zapisu plików w PHP to:

`fopen()`, `fwrite()`, `fclose()` (gdzie literka `f` oczywiście oznacza `file`) oraz uproszczona: `file_put_contents()`.

### Tworzenie i zapis do pliku – `fopen()`, `fwrite()`, `fclose()`

```
<?php  
$plik = fopen("dane.txt", "w"); // tryb "w" = zapis (usuwa poprzednią zawartość)  
fwrite($plik, "To jest pierwsza linia pliku.\n");  
fwrite($plik, "To jest druga linia pliku.");  
fclose($plik);  
?>
```

Po uruchomieniu niniejszego skryptu na dysku serwera powstanie plik o nazwie

`dane.txt` z wpisanymi liniami tekstu. Znak `\n` powoduje oczywiście przejście do nowej linii (*ang. new line*).

### Szybki zapis całej zawartości – `file_put_contents()`

```
$dane = "To jest zawartość pliku zapisana jedną funkcją."  
file_put_contents("log.txt", $dane);
```

Funkcja ta tworzy plik lub nadpisuje jego zawartość w całości – bez potrzeby otwierania i zamykania pliku.

### Dopisywanie do pliku – tryb `"a"` lub `FILE_APPEND`

```
file_put_contents("log.txt", "Nowy wpis do logu.\n", FILE_APPEND);
```

Dzięki parametrowi

`FILE_APPEND`, nowa linia zostanie dopisana na końcu istniejącego pliku – zamiast nadpisać jego zawartość.

## Uprawnienia do zapisu

Aby PHP mogło zapisać plik, katalog musi mieć oczywiście odpowiednie uprawnienia. W systemach Windows upewnijmy się, że serwer Apache ma prawa do zapisu w danym folderze. Przyda się nam zatem następujący fragment kodu. Sprawdzanie, czy zapis pliku się udał:

```
<?php
$zapisano = file_put_contents("plik.txt", "Zawartość");
if ($zapisano !== false) {
    echo "Zapisano $zapisano bajtów.";
} else {
    echo "Błąd zapisu!";
}
?>
```

Dzięki tym funkcjom możemy zrealizować zapis formularzy, eksport danych do pliku CSV, tworzenie kopii danych lub logowanie błędów aplikacji. Pamiętajmy tylko o bezpieczeństwie – nigdy nie pozwalajmy użytkownikowi ustawić dowolnej nazwy pliku czy treści bez filtrów i walidacji!

## Podsumowanie

Wbudowane funkcje w PHP to potężny zestaw narzędzi, który znacznie ułatwia codzienną pracę programisty. Znajomość funkcji do pracy z tekstem, liczbami, losowaniem, konwersją typów i warunkami logicznymi to absolutna podstawa na egzaminie INF.03, jak i w praktycznej pracy przy tworzeniu aplikacji internetowych. Nie musimy uczyć się wszystkich – warto jednak opanować te, które najczęściej pojawiają się w zadaniach i projektach.

Natomiast dzięki funkcjom realizującym zapis plików na dysku przez skrypt PHP możemy zrealizować zapis formularzy, eksport danych do pliku CSV, tworzenie kopii danych lub logowanie błędów aplikacji.