

Formatowanie kodu HTML wypisanego przez PHP

Podczas generowania kodu HTML w PHP, bardzo często chcemy zadbać o czytelność wynikowego kodu źródłowego. Jeśli nie zastosujemy odpowiedniego formatowania, kod HTML może być wypisany w "jednej długiej linii", co utrudnia debugowanie i analizę źródła strony w przeglądarce.

Dzisiaj poznamy kilka metod formatowania kodu HTML w PHP:

- Znaki specjalne (`\r`, `\t`, `\n`) – pozwalają wstawić nowe linie, tabulatory i powroty karetki.
- Wieloliniowe `echo` – zapis wielowierszowy sprawia, że kod HTML w PHP wygląda bardziej przejrzysto.
- Heredoc i Nowdoc – alternatywne sposoby na czytelne generowanie większych fragmentów HTML.

Problem: PHP generuje kod HTML w jednej linii

Kiedy wypisujemy kod HTML w PHP za pomocą

`echo`, domyślnie nie ma żadnych znaków nowej linii – cała zawartość wyświetla się jako jedna długa linia kodu w źródle strony:

```
echo "<html><head><title>Test</title></head><body><h1>Witaj!</h1><p>To jest testowy
paragraf.</p></body></html>";
```

Formatowanie kodu HTML za pomocą `\n` (new line)

Znak

`\n` (ang. newline) powoduje przejście do nowej linii. Jest to najczęściej używany znak formatowania w PHP i działa również w systemach Linux oraz macOS.

```
echo "<html>\n<head>\n<title>Test</title>\n</head>\n<body>\n";
echo "<h1>Witaj!</h1>\n";
echo "<p>To jest testowy paragraf.</p>\n";
echo "</body>\n</html>";
```

Porównajmy w przeglądarce (po wybraniu opcji "Wyświetl źródło strony") jak wygląda kod źródłowy z dodanymi przejściami do nowej linii.

Znaki: `\r` (powrót karetki) oraz `\t` (tabulator)

- `\r` – tzw. powrót karetki (ang. carriage return), stosowany głównie w systemie Windows.
- `\t` – tabulator, który pozwala dodać poziome wcięcia dla lepszej czytelności kodu.

```
echo "<html>\n<head>\n<title>Test</title>\n</head>\n<body>\n";
echo "\t<h1>Witaj!</h1>\n";
echo "\t<p>To jest testowy paragraf.</p>\n";
echo "</body>\n</html>";<b></b>
```

Wieloliniowe łańcuchy tekstowe – heredoc i nowdoc

W PHP, oprócz klasycznego zapisu łańcuchów znaków za pomocą cudzysłowów (

`""` lub `' '`), mamy także możliwość tworzenia tzw. wieloliniowych stringów – czyli ciągów znaków, które rozciągają się na kilka linii kodu. Idealne np. do generowania HTML lub treści e-maila.

W tym celu możemy wykorzystać dwie konstrukcje – tzw. *heredoc* i *nowdoc*.

Heredoc – tekst z interpolacją

Konstrukcja

`heredoc` działa podobnie do tekstu w podwójnych cudzysłowach – czyli pozwala interpolować zmienne, stosować znaki specjalne (np. `\n`) i ogólnie rzecz biorąc: kod “rozumie”, że to PHP:

```
<?php
$imie = "Ania";
echo <<<KONIEC
Cześć $imie!
Dzisiaj jest: <strong>poniedziałek</strong>
Miłego dnia!
KONIEC;
?>
```

Uwaga: końcowy identyfikator (np.

`KONIEC`) musi znajdować się w osobnej linii (bez spacji, tabulatorów – choć średnik po identyfikatorze jest dozwolony) i musi być dokładnie taki sam jak identyfikator otwierający.

Nowdoc – tekst surowy, jak w apostrofach

Konstrukcja tzw.

`nowdoc` działa jak tekst w pojedynczych cudzysłowach – a zatem nie interpoluje zmiennych oraz nie interpretuje znaków specjalnych – wszystko traktuje jako czysty tekst:

```
<?php
$imie = "Ania";
echo <<<'KONIEC'
Cześć $imie!
Dzisiaj jest: <strong>poniedziałek</strong>
Miłego dnia!
KONIEC;
?>
```

Efekt działania powyższego kodu będzie taki, iż wypisany zostanie tekst dosłownie taki, jaki zapisaliśmy – czyli

`$imie` nie zostanie zamienione na “Ania”. To przydatne np. przy wklejaniu fragmentów kodu źródłowego lub dokumentacji technicznej, gdzie nie chcemy, aby PHP ingerował w treść.

Heredoc czy nowdoc – kiedy używać?

- **Heredoc** – gdy chcemy wypisać tekst z dynamicznymi danymi, np. zmiennymi użytkownika, datą, wartościami liczbowymi powstawanymi ze zmiennych

- **Nowdoc** – gdy piszemy “na sztywno”, np. kod HTML, JS, SQL, który ma zostać wyświetlony dosłownie, bez wartości wstawianych w miejsce zmiennych.

Dzięki tym mechanizmom nasz kod staje się bardziej czytelny i uporządkowany – szczególnie przy pracy z dłuższymi blokami tekstu.

Podsumowanie wiedzy z lekcji

- Podczas generowania kodu HTML w PHP, domyślne wypisywanie zawartości przez `echo` może prowadzić do nieczytelnych wyników – cała zawartość HTML łąduje w jednej długiej linii. Aby temu zapobiec, stosujemy znaki specjalne, które poprawiają przejrzystość kodu źródłowego widocznego w przeglądarce:
 - `\n` (nowa linia),
 - `\t` (tabulator),
 - `\r` (powrót karetki),
- W przypadku większych fragmentów tekstu HTML warto stosować konstrukcje tzw. *heredoc* i *nowdoc*:
 - **Heredoc** umożliwia interpolację zmiennych (czyli ich podmianę na wartości w miejscu wystąpień w kodzie) i działa jak tekst w cudzysłowie,
 - **nowdoc** traktuje tekst dosłownie (jak w apostrofach) – idealny do prezentowania kodu źródłowego lub dokumentacji, bo nie wstawimy do niego wartości zmiennych.

Dzięki tym metodom nasze skrypty PHP stają się bardziej przejrzyste i wygodne w utrzymaniu, co jest istotne zarówno w projektach edukacyjnych, jak i na egzaminie INF. Oczywiście formatowanie źródła zazwyczaj nie jest koniecznością na egzaminie INF.03, ale warto zadbać także i o ten aspekt tworzenia kodów źródłowych.