Php

Wstęp

- PHP jest szeroko stosowanym **skryptowym językiem programowania** służącym do tworzenia dynamicznych i interaktywnych serwisów internetowych.
- Do edycji plików PHP można wykorzystywać **dowolny edytor tekstowy** (np. Notatnik) umożliwiający edycję zwykłych plików tekstowych. Możliwe jest również zastosowanie specjalnego edytora tekstowego wyposażonego w dodatkowe funkcje ułatwiające edycję plików PHP, takie jak kolorowanie składni.
- Bloki programu PHP rozpoczynają się zawsze od **<?php**, a kończą na **?>**. Blok programu PHP może być **wielokrotnie umieszczany** w dowolnym miejscu dokumentu HTML.

Pierwszy program php

- Utwórz szkielet strony HTML
- Dodaj blok PHP
- Utwórz program w PHP wyświetlający napis *hello world*.
- W sekcji body umieść fragment programu PHP według poniższego przykładu:

```
<html>
<body>
    <?php
    echo "hello world";
    ?>
</body>
</html>
```

- Plik PHP zwykle zawiera pewną ilość znaczników HTML i pewną ilość bloków PHP.
- Każda linia kodu w PHP musi kończyć się średnikiem. Średnik jest separatorem, dzięki, któremu możliwe jest odróżnienie jednej instrukcji od drugiej.
- W powyższym przykładzie użyta została instrukcja **echo** służąca do wyświetlania napisów.

Zmienne

- Zmienne używane są w skryptach PHP w celu przechowywania wartości takich jak łańcuchy znaków (napisy), liczby, tablice (wektory) lub wyniki działania funkcji.

- Wszystkie zmienne w PHP zaczynają się od znaku $.

- Za pomocą operatora = odbywa się przypisanie wartości do zmiennej:

```
<?php
    $napis = "hello world";
    $liczba = 123;
    echo $napis;
?>
```

Operator połączenia łańcuchów znaków

- Do łączenia łańcuchów znaków służy operator . (kropka).
- Poniższy przykład pokazuje w jaki sposób można łączyć ze sobą napisy.

```
<html>
<body>
    <?php
            $napis = "hello world";
            $liczba = 123;
            echo $napis . " " . $liczba;
    ?>
</body>
</html>
```

Zadanie

- Zmodyfikuj wcześniejszy program w taki sposób, aby wartość zmiennej $liczba, została wypisana pogrubioną czcionką.

- **Podpórka:**

  Do pogrubiania służy znacznik . Zmodyfikowany skrypt PHP oprócz wartości zmiennych powinien również wyświetlać znaczniki HTML.

Operatory

- W języku PHP istnieją operatory wykonujące operacje arytmetyczne (m.in. + - * /) oraz operacje porównania (m.in. < > ==) itd.
- W tym ćwiczeniu napiszemy program, który będzie obliczał cenę brutto na podstawie znanej ceny netto i wartości podatku VAT.

```
<html>
<body>
    <?php
        $cena_netto = 1200;
        $vat = 22;
        $cena_brutto = $cena_netto + $cena_netto * $vat / 100;
        echo $cena_netto;
        echo $cena_brutto;
    ?>
</body>
</html>
```

- Popraw program w taki sposób, aby cena netto i brutto wyświetlane były w osobnych liniach (potrzebny będzie znacznik
)

Zadanie

- Zmodyfikuj program w taki sposób, aby wyświetlana została również wartość zapłaconego podatku.

- **Podpórka:**

  Wartość podatku obliczamy poprzez odjęcie od ceny brutto ceny netto.

Instrukcja If..Else

- Utwórz program w PHP obliczający podatek do zapłacenia. Zakładamy podatek liniowy w wysokości 19%. Podatek liczony jest od różnicy między przychodem a kosztem. Podatek jest jednak należny tylko w przypadku, gdy ta różnica jest większa od zera. Do tego celu wykorzystamy funkcję **if**.

- Umieść we właściwym miejscu fragment programu PHP według poniższego przykładu:

```
$przychod = 123000;
$koszt = 70000;
$stawka =          19;
$dochod = $przychod - $koszt;
if ($dochod > 0)
$podatek = $dochod * $stawka / 100;
else
$podatek = 0;
echo "Podatek wynosi <b>$podatek</b> PLN";
```

- Zmień wartości przychodu i kosztu, w taki sposób, aby koszt przekraczał przychód i sprawdź działanie warunku.

Zadanie

- Zmodyfikuj wcześniejszy program tak, aby kwota podatku była wyświetlana tylko w przypadku dodatniego dochodu. W przeciwnym przypadku, gdy nie ma podatku do zapłacenia, powinien pojawiać się napis "Nic nie płacisz, ale będzie kontrola".

Pętle – instrukcja While

- Pętle umożliwiają wykonywanie tego samego kawałka programu odpowiednią liczbę razy.

- Instrukcja **While** powoduje wykonywanie danego fragmentu programu tak długo jak podany warunek jest spełniony.
- Poniższy program demonstruje pętlę, która wykonywana jest dopóki zmienna i jest mniejsza lub równa 5. Zmienna i natomiast zostaje zwiększona o jeden przy każdym powtórzeniu pętli.

```
<html>
<body>
    <?php
            $i = 1;
            while ($i <= 5)
            {
            echo "Liczba to $i <br/>";
            $i = $i + 1;
            }
    ?>
</body>
</html>
```

Zadanie

- Zmodyfikuj program w taki sposób, aby generował on tabelę kwadratów liczby.

- **Podpórka**

  Zajrzyj do zadań z HTML, aby przypomnieć sobie, w jaki sposób tworzona jest tabela w HTML.

1	1
2	4
3	9
4	16
5	25

Formularze

- Utwórz nowy plik z formularzem wg wzoru:

```
<html>
<body>
    <form action="welcome.php" method="GET">
    Imie: <input type="text" name="name"><br/>
    Wiek: <input type="text" name="age"><br/>
    <input type="submit">
    </form>
</body>
</html>
```

- Powyższy plik tworzy formularz, na którym znajdują się dwa pola tekstowe oraz przycisk do wysyłania danych formularza. Znacznik <form> posiada dwa atrybuty. Pierwszy – **action** – mówi o tym, jaka strona zostanie załadowana po przyciśnięciu przycisku wyślij. Jest to również strona, do której zostaną przesłane dane z formularza. W naszym przypadku załadowana zostanie strona *welcome.php*. Drugi atrybut – **method** – określa sposób przesyłania danych formularza. W przypadku metody **GET**, dane są przesyłane jako część adresu URL. Każde pole tekstowe posiada nazwę (tutaj *name* oraz *age*). Za pomocą tej nazwy, w
- skrypcie PHP, moŜliwe będzie odczytanie danych wprowadzonych w dane pole.

Odczytanie danych formularza

- Utwórz nowy plik PHP wg wzoru:

```
<html>
<body>
   Witaj <?php echo $_REQUEST["name"]; ?>.<br />
   Masz <?php echo $_REQUEST["age"]; ?> lat.
</body>
</html>
```

- Zmień w pliku formularz.html metodę z **GET** na **POST**. Sprawdź, jaka będzie różnica.

Zadanie

- Zmodyfikuj program w taki sposób, aby pozdrowienie pojawiało się tylko w przypadku, gdy wiek jest większy lub równy 18. W przeciwnym wypadku niech pojawia się napis "Niestety jesteś zbyt młody".

- **Podpórka**:

  Należy użyć instrukcji warunkowej **if**.

Sesja

- W Internecie istnieje poważny problem: serwer WWW nie wie kim jesteś, co robisz i co robiłeś wcześniej, ponieważ **protokół HTTP** nie utrzymuje stanu.
- Sesja PHP rozwiązuje ten problem poprzez stosowanie zmiennych sesyjnych, które umożliwiają przechowywanie pewnych informacji na serwerze (np. zalogowanego użytkownika czy zawartości koszyka) w trakcie trwania sesji użytkownika.

Rozpoczęcie sesji

- Utwórz nowy plik wg wzoru:

```
<?php session_start(); ?>
    <html>
    <body>
    </body>
    </html>
```

- Sesję tworzy się poprzez wykonanie funkcji session_start(). Wywołanie tej funkcji musi mieć miejsce przed wystąpieniem znacznika <html>.

Zapis i odczyt zmiennej sesyjnej

- Zmienne sesyjne odczytuje się przy pomocy specjalnej zmiennej $_SESSION. W poniższym przykładzie utworzony został licznik odwiedzin strony. Funkcja isset sprawdza czy zmienna sesyjna 'licznik' została już wcześniej ustawiona. Jeżeli licznik został już ustawiony zwiększamy go o jeden. W przeciwnym przypadku ustawiamy licznik na wartość 1.

```
<?php
    session_start();
    if ( isset( $_SESSION["licznik"] ) )
    $_SESSION["licznik"] = $_SESSION["licznik"] + 1;
    else
    $_SESSION["licznik"] = 1;
    ?>
<html>
<body>
    Odwiedziłeś tę stronę już
<?php echo $_SESSION["licznik"]; ?> razy.
</body>
</html>
```

Zadanie

- Utwórz skrypt, który w zmiennej sesyjnej będzie przechowywał imię użytkownika. Jeżeli imię będzie już ustawione, skrypt będzie wyświetlał powitanie. Jeżeli natomiast imię nie będzie zapamiętane w sesji, skrypt będzie wyświetlał formularz z prośbą o podanie imienia.

Dziękujemy za uwagę