

Operacje na łańcuchach w PHP

przydatne funkcje:

`mb_strtolower(ciąg_znaków, kodowanie)` - zamienia litery na małe (łącznie z polskimi)

`mb_strtoupper(ciąg_znaków, kodowanie)` - zamienia litery na duże (łącznie z polskimi)

`mb_convert_case (ciąg_znaków, tryb, kodowanie)`

tryb:

`MB_CASE_UPPER` - zmiana wszystkich liter na wielkie

`MB_CASE_LOWER` - zmiana wszystkich liter na małe

`MB_CASE_TITLE` - zmiana pierwszych liter w wyrazach na wielkie

`str_replace(to_co_chcemy_zastapic,to_czym_chcemy_zastapić,ciąg_znaków)` - zamienia ciąg znaków na inny podany jej w parametrze.

```
"kot pies mysz mysz kot mysz"
```

```
str_replace("mysz","sowa","kot pies mysz mysz kot mysz")
```

```
$imie="Tomek";
```

`substr(ciąg_znaków,początek_wycinania,ile_znakow)` - wycina z podanego tekstu jakiś fragment

jako pierwszy paramter pobiera ona ciąg z którego będzie wycinany fragment, jako drugi miejsce, z którego będzie rozpoczęte wycinanie (0 jeśli od pierwszego znaku), a ostatni, opcjonalny parametr wskazuje ilość znaków do wycięcia (jeśli pominie się ten parametr, to zwrócony zostanie pod-ciąg od wskazanego znaku początkowego do końca ciągu).

`explode(separator,ciąg_znaków)` - wycina fragmenty tekstu ograniczone separatorem

np.

```
$znak=explode(";", "aaa;bbb;ccc");
```

```
echo($znak[0]); //wyświetli aaa
```

```
echo($znak[1]); //wyświetli bbb
```

```
echo($znak[2]); //wyświetli ccc
```

`trim(ciąg_znaków)` - usuwa z tekstu spacje z początku i z końca

`strip_tags()` - usuwa ze stringu znaczniki HTML i PHP

`htmlspecialchars()` - zamienia wybrane znaki w wyrażeniu, na odpowiadające im zamienniki (tzw. encje), dzięki czemu te nie będą interpretowane przez przeglądarkę jako znacznik HTML i zostaną prawidłowo wyświetlone.

`rand(x,y)` - losuje liczby z przedziału od x do y

szyfrowanie:

`md5(ciąg_znaków)` - szyfruje do 32 znaków

sha1(ciąg_znaków) - szyfruje do 40 znaków

te dwie funkcje działają tak, że za każdym razem szyfrują do takiej samej postaci

```
password_hash(ciąg_znaków,PASSWORD_DEFAULT);
```

ta funkcja szyfruje tak, że za każdym razem ten sam ciąg znaków szyfruje inaczej, żeby sprawdzić czy ciąg_znaków się zgadza z tym zaszyfrowanym, to trzeba użyć funkcji **password_verify(ciąg_znaków,po_szyfrowaniu)** ta funkcja zwraca true, gdy się zgadza i false, gdy nie.

Zadania

Zad1

1. Utwórz formularz składający się z dwóch pól tekstowych do podania imienia i nazwiska.
2. Zabezpiecz go tak, aby można było wpisywać tylko małe i duże litery (również polskie) //poprzednia lekcja
3. korzystając z dzisiejszych funkcji:
 - zabezpiecz imię i nazwisko, tak, żeby zamieniał wpisane dane do postaci wpisane np. sYLwiA, do postaci Sylwia
 - pousuwaj znaczniki html, niepotrzebne spacje
 - utwórz login, login ma się składać z pierwszej litery imienia i całego nazwiska (wcześniej trzeba zamienić w imieniu i nazwisku polskie litery na ich odpowiedniki, np ą na a)
 - wygeneruj hasło użytkownikowi (liczba z przedziału od 100 do 1000)
 - zaszyfruj hasło 3 metodami
 - wyświetl komunikat np.
Łukasz Świątkowski
login: lswiatkowski
hasło:123
szyfr md5: //i tu to co wyjdzie
szyfr sha1: //i tu co wyjdzie
szyfr password_hash: //i tu co wyjdzie

Zad2

- jedno pole tekstowe i przycisk
- w pole podaje się datę urodzenia w postaci rok-miesiac-dzien (poprzednia lekcja)
- w kodzie sprawdzamy czy podana data zgadza się z dzisiejszą (porównujemy tylko miesiąc i dzień, trzeba też skorzystać z funkcji date) i jeżeli się zgadza wyświetla się komunikat "dziś są Twoje urodziny!".