DUM 4 téma: Cykly

ze sady: 2 **tematický okruh sady:** Tvorba dynamických www stránek
ze šablony: 08 – Internet **určeno pro:** 3. ročník
vzdělávací obor: 18-20-M/01 Informační technologie
vzdělávací oblast: odborné vzdělávání
metodický list/anotace: viz VY_32_INOVACE_08204ml.pdf
pomocné soubory:

1. Dnes se podíváme na tvorbu programu pomocí cyklů, cykly slouží k odstranění monotónní práce. Umožní vytvořit tabulku o velikosti 10x10 buněk pouhými třemi řádky. Je jasné, že cykly samy nám nebudou moc platné, ale v budoucnu, až se naučíme pracovat s polem, nám v tvorbě programů moc pomohou.
   a. **For** cyklus – tento cyklus je používán známe-li předem počet opakování. Jeho struktura vypadá takto: *for(deklarace;podmínka;inkrement){tělo cyklu}*
      i. Ukázka for cyklu

```php
<?php
  for($i=0;$i<20;$i++){
    echo $i;
  }
?>
```

Vypíše hodnoty od 0 do 19 jelikož podmínka je $i<20 a poslední číslo menší než 20 je 19

      ii. ještě aby nedošlo k nejasnostem do těla cyklu nemusíme dosazovat $i=0 ani $i<20 a $i++

```php
<?php
  for($i=5;$i<=30;$i=$i+5){
    echo $i;
  }
?>
```

Toto vypíše násobky 5 5, 10, 15, 20, 30 včetně, jelikož podmínka je menší nebo rovno, čemuž třicítka odpovídá.

   b. **While** cyklus – tento cyklus je používán když neznáme předem opakování samozřejmě že se dá použít i když známe ale je to zbytečné. Jeho struktura vypadá takto: *while(podmínka){tělo cyklu}*

```php
<?php
  $i=0;
  while($i!=5) {
    echo $i++;      //interval vypsaných čísel bude 0 - 4
                    //protože nejprve vypíše a až poté přičte
  }
?>
```

   c. **Dowhile** cyklus – užití tohoto cyklu není tak časté jako předchozích dvou. Používá se tehdy víme li že dojde ke splnění podmínky minimálně jednou. Proč?

Nejprve si ukážem strukturu a pak bude možné pochopit to lépe. Struktura :

do{tělo cyklu}while(podmínka);

```php
<?php
    $i=0;
    do{
      echo $i++;
    }while($i<6);
?>
```

Vypíše čísla 0, 1, 2, 3, 4, 5, 6 protože nejprve zde dochází k výpisu a až poté ke kontrole podmínky. Teď už tušíte proč je jeho použití minimální.

d. **Foreach** cyklus - tento cyklus se používá pouze při práci s polem, proto si jen řekneme o jeho existenci, ale neukážem si jeho syntaxi.

2. Příklady použití cyklů:

a. Kombinace cyklů a podmínek

```php
<?php
  for($j=1;$j<19;$j++){    //struktura for cyklu
      if($j%3==0){         //podmínka kontrolující dělitelnost 3 pokud je číslo dělitelné 3
          echo $j."<br>"; //vypíše hodnotu j a zalomí řádek
      }
      else{                //pokud není číslo dělitelné 3
          echo $j.",";     //vypíše hodnotu j a čárku
      }
  }
?>
```

1,2,3
4,5,6
7,8,9
10,11,12
13,14,15
16,17,18

b. Kombinace cyklu a SESSION

```php
<?php
  if(!isset($_SESSION["navstiveno"])){          //pokud neni nastavena session
      $_SESSION["navstiveno"]="ne";             //nastaví session na ne
  }
  else;                                         // pokud je nastavena neudela nic
  $cislo=10;                                    //deklarace pomocne proměnné cislo
  while($_SESSION["navstiveno"]!="ano") {       //kontrola jestli uz nema session hodnotu ano
      $cislo++;                                 //inkrement pomocne promenne
      if($cislo==20)$_SESSION["navstiveno"]="ano"; //pokud cislo = 10 session se nastavi na ano
      echo $cislo;
  }                                             /*nyní když nevypneme prohlížeč daný script
                                                už neproběhne znovu*/
?>
```

i. První spuštění

11, 12, 13, 14, 15, 16, 17, 18, 19, 20,

ii. Druhé spuštění bez vypnutí okna nevypíše nic protože session je nastaveno na hodnotu „ano“.

c. Vnoření výce cyklů do sebe

```
<table border="1">
<?php
  $radek=rand(5,20);          //k určení počtu řádků, bude generováno náhodně číslo mezi 5 a 20
  $sloupek=rand(5,20);        //k určení počtu sloupků, bude generováno náhodně číslo mezi 5 a 20
  $e=0;                       //deklarace pomocne proměnné e
  for($i=1;$i<=$radek;$i++){  //forciklus pro tvorbu řádků;
    echo "<tr>";              //výpis řádku
    for($j=1;$j<=$sloupek;$j++){  //forciklus pro tvorbu buněk
        echo "<td>".$e++."</td>";  //výpis buněk a navýšení pomocné proměnné e
    }
    echo "</tr>";             //výpis ukončení řádku
  }
?>
```

Tab 4x5

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

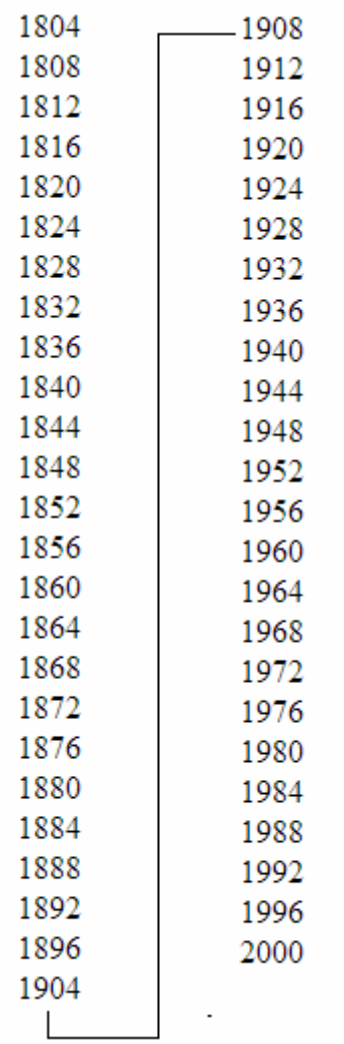
Tab 6x8

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47

První forcyklus slouží k vypsání řádku protože se vykoná méněkrát než cyklus druhý. Cyklus druhý se provede vícekrát, protože se splní vždy, když se jednou splní cyklus první. V případě hodnot řadek=4 sloupek=4 se první cyklus stane právě 4krát, ale druhý cyklus se stane 4x4=16krát. 4x4 protože první cyklus proběhne 4krát a při každém spuštění prvního cyklu proběhne druhý 4x.

3. Úkol:
   a. Vytvořte čítač přestupného roku pomocí cyklu, od roku 1800 do roku 2000, podmínky přestupného roku jsou:

      i. Roky dělitelné 4 jsou přestupné roky, ale roky dělitelné 100 nejsou přestupné roky, ale roky dělitelné 400 jsou přestupné roky.
      ii. roky dělitelné 100 nejsou přestupné roky, ale roky dělitelné 400 jsou přestupné roky.
      iii. roky dělitelné 400 jsou přestupné roky.

Potřeba přestupných roků je vyvolána skutečností, že skutečná délka roku je 365,242 dní a ne tedy 365 dní, jak je obecně počítáno. K vyrovnání této odchylky se přidává jeden den 29. února v letech, které jsou dělitelné 4 (jako byl například rok 1992). Jelikož má rok o něco méně než 365,25 dní, způsobuje přidání jednoho dne každé čtyři roky chybu dalších tří dnů, které se přidávají v periodě 400 let. Z tohoto důvodu musí být také jeden z každých čtyř roků století přestupným rokem.

   b. Výpis: