

## 20. Algoritmizace - cykly (klasifikace, principy, rozdíly), konkrétně cyklus for (PHP, Action Script)

### PHP cykly

Cykly se používají chceme-li opakovat určitou činnost, několikrát za sebou, opakovaně provádět stejnou proceduru. Postupně ověřovat data, nebo generovat informace.

### Do while

Chceme zapsat čísla od 1 do 100. Je zbytečné zapsat 1,2,3... Poslouží nám jednoduchý skript:

```
$c = 1;
do {
echo($c . " ");
$c++;
}
while($c<101);
```

`c` je proměnná. Cyklus `do {proces} while(podmínka)` vždy vypíše hodnotu proměnné `c`. Zároveň k ní vždy přičte 1 `c++`. Nakonec jsme příkazem `while(podmínka)` cyklus omezili, cyklus se bude provádět, pouze když je proměnná menší než 101. V momentě kdy není menší se skript přeručí.

Stejným způsobem můžete vytvářet i jiné skripty, například pracovat s [polem](#) nebo databází. Projíždět postupně jednotlivé položky a vypisovat jejich obsah.

### While

Příkaz `while(podmínka)` je možné použít samostatně.

```
$c = 1;
while($c<101){
echo($c . " ");
$c++;
}
```

Výsledek je stejný, pouze pokud `c` přiřadíte hodnotu 101, příkaz `do while` vypíše 101, protože podmínka je definována až po procesem (skript nezná podmínku a tak napíše 101, teprve poté kontroluje podmínky). Příkaz `while 101` nezapíše, protože podmínka předchází proces. Příkaz `do while` jednou provede proces, když už podmínka neplatí. Na tento rozdíl je třeba si vždy dát dobrý pozor a ověřit skript, když se dostane k podmínce.

## For

Posledním příkazem je příkaz `for`. Syntaxe: `for(inicializace proměnné; podmínka; operace) {skript}`.

```
for($c=1; $c<101; $c++){  
echo("$c. " " ");  
}
```

Příkaz `for` se užívá především, když víme kolikrát chceme proces uskutečnit. Počet je konečný. U `do while` a `while` ověřujeme proměnnou, která se v procesu mění. U `for` zavádíme přímo proměnnou určenou k tomu, aby se určil počet procesů.

**Ukažte na příkladech (Flash či PHP)!!!**