

Nepisana pravila dobre prakse programiranja podrazumijevaju da isti blokovi koda ne bi trebalo da se ponavljaju, odnosno kopiraju više puta. Ipak, veoma često imamo potrebu da određene blokove koda iznova izvršavamo. **For** petlja se koristi kada se unaprijed zna (ili se može izračunati) tačan broj **iteracija**, odnosno prolazaka kroz petlju.

for	(	\$i=0	;	\$i<10	;	\$i++	)	{ }
For petlje možete <b>gnijezditi</b> jednu u drugu. Vrijednost \$i ostaje i nakon izvršenja petlje!	<b>Inicijalizacija</b> -Osnovna i najčešća forma <i>for</i> petlje podrazumijeva da se u dijelu koda definiše jedna promjenljiva (najčešće promjenljiva \$i) koja će služiti kao brojač. U najvećem broju slučajeva je korisno da inicijalna vrijednost ove promjenljive bude 0, jer se često pomoću ove petlje prolazi kroz niz, čije indeksiranje počinje nulom. Pored inicijalizacije promjenljive koja će biti brojač, ovdje možemo definisati i druge promjenljive ukoliko su potrebne.		<i>Inicijalizacija, uslov i završna izjava</i> se odvajaju znakom tačka-zarez (;) i to je <i>obavezno</i> . U specijalnim slučajevima je dozvoljeno potpuno izostaviti bilo kakav kod u nekom od ovih dijelova (ili čak u sva tri dijela), ali dva znaka ; unutar zagrada ipak moraju postojati.		<b>Uslov</b> – U ovom dijelu se definiše izraz čiji rezultat izvršavanja treba da vrati boolean vrednost. Uslov se najčešće konstruiše tako da se neka vrijednost poredi sa promjenljivom definisanom u brojaču. U slučaju da je ovaj uslov ispunjen nastavlja se sa izvršavanjem <i>for</i> petlje, u suprotnom se izvršavanje petlje prekida.		<b>Završna izjava</b> - Kod definisan u ovom delu se izvršava na kraju svake iteracije. Najčešće se koristi za uvećanje (ili drugu izmenu) brojača.	
	Jedan prolaz kroz petlju naziva se <b>iteracija</b> . Kada testiranje uslova vrati vrednost <i>false</i> , izvršavanje petlje se prekida. Ponekad možemo u jednoj inicijalizaciji definisati <b>više parametara</b> . <code>for (\$i=0, \$a=5; \$i&lt;10; \$i++, \$a++)</code>						<pre>\$i = 0; for(;;){   if(\$i &gt; 9){ break;}   echo \$i;   \$i++; }</pre>	
KONTROLA TOKA PETLJE			1. Ručnom modifikacijom kontrolne promjenljive (brojača).					
			2. Upotreba ključne riječi, <b>break</b> , koja bezuslovno prekida blok koda. <code>break</code> može imati i <b>parametar</b> koji je brojčana vrijednost, a koji označava „dubinu“ ugniježdene petlje čiji blok koda prekidamo, <code>{ break 2; }</code>					
			3. <b>continue</b> –možemo prekinuti izvršavanje trenutne iteracije petlje i preći na sljedeću. <code>if(\$covjek != 20) {continue;}</code>					
foreach(\$niz as \$value) foreach(\$niz as \$key=>\$value)					foreach petlja je varijacija <i>for</i> petlje, namjenjena isključivo radu sa nizovima. Naročito asocijativnim. Kada radimo sa <i>foreach</i> -om, ne rukujemo članovima niza eksplicitno kroz njihove indekse, već samo dobijamo njegove vrijednosti, pa se tako za <i>foreach</i> može reći da je dobar kada želimo brzo i jednostavno da izvršimo čitanje nekog niza, ali ne i neke ozbiljnije intervencije na njemu.			