

```
print(rand(1,49)." ");
}
?>
```

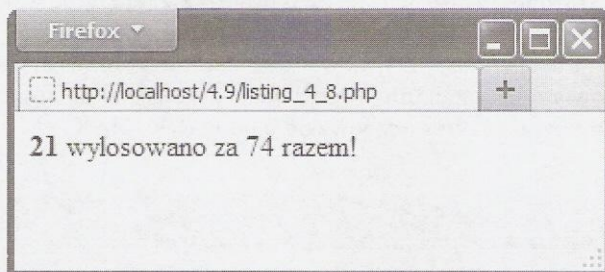
Innym rodzajem pętli jest pętla **while**. Jest ona wykonywana, dopóki warunek logiczny jest spełniony. Jest to najlepsza pętla, jeżeli chcemy wykonywać daną operację do czasu zajścia określonego warunku. Dlatego idealnie nadaje się do odczytywania plików. Pętla wówczas jest wykonywana tak długo, aż napotka koniec pliku.

```
while (warunek){
instrukcje;
}
```

Przykład z list. 21.2 przedstawia działanie pętli **while**. Warunkiem zakończenia działania pętli jest wylosowanie przez funkcję losującą **rand()** liczby 21. Wewnątrz pętli umieszczono licznik. Zlicza on liczbę wykonanych pętli. W rezultacie podawana jest informacja, za którym razem została wylosowana liczba 21. W tym przypadku przy każdym uruchomieniu skryptu liczba wykonanych pętli będzie inna.

Listing 21.2

```
<?php
$x=1;
while(rand(1,100)!=21) {
$x++;
}
print("<b>21</b> wylosowano za ".$x." razem!");
?>
```



Rys. 21.2. Wynik działania skryptu z list. 21.2 – pętla **while**

Pętla **do/while** stanowi odmianę pętli **while**, ale z tą różnicą, że ta pętla jest wykonywana raz przed sprawdzeniem warunku logicznego. Jest to jeden z rzadziej stosowanych rodzajów pętli.

```
do {
instrukcje;
}
while (warunek)
```

Przykład z list. 21.3 przedstawia działanie pętli **do/while**. Kolejne wykonania pętli powodują sumowanie się wylosowanych liczb z zakresu od 1 do 20. Pętla będzie wykonywa-