

SESSIONI

La sessione è come un contenitore nel quale posso mettere un numero indefinito di valori e richiamarli da qualsiasi punto del mio sito. A differenza dei cookies non posso impostare una scadenza prolungata. La sessione per sua stessa natura dura il tempo di una sessione, dunque il tempo che io passo su un sito. Se chiudo il browser e riaccio al sito, i dati di sessione non esisteranno più. La sessione inoltre viene salvata sul server, a differenza del cookie che viene scritto sul client. E per finire: 1 cookie = 1 valore, 1 sessione = indefiniti valori.

Con i cookie ha invece in comune la questione degli header che abbiamo trattato poco fa. Facciamo subito un'esempio.

```
// prima cosa dobbiamo avviare la sessione – PRIMA di qualsiasi output  
// utilizziamo la funzione session_start()  
  
session_start();  
  
// una volta avviata possiamo accedere all'array superglobale $_SESSION  
  
$_SESSION['nome'] = "Maurizio";  
$_SESSION['cognome'] = "Tarchini";
```

A questo punto i valori nome e cognome saranno disponibili in qualunque punto di qualunque pagina del mio sito, a condizione che venga richiamata la sessione con la funzione session_start().

Quindi se in un'altra pagina scriveremo

```
session_start();  
  
echo $_SESSION['nome'];  
  
// scriverà a video Maurizio
```

L'esempio classico di utilizzo delle sessioni è il carrello della spesa nei siti di e-commerce. Voi girate per il sito, ogni tanto mettete qualcosa nel carrello (che in pratica è una sessione) e poi alla fine cliccate su acquista. A questo punto la sessione viene analizzata e vengono estratti tutti gli articoli che avete memorizzato (messo nel carrello).

Un altro esempio potrebbe essere un semplice sistema di autenticazione. L'utente fa il login, nome utente e password vengono controllati, se esistono e corrispondono avvio una sessione impostando un parametro (qualsiasi). Ora nelle pagine che desidero proteggere, all'inizio, verificherò l'esistenza del parametro di sessione che imposto in caso di login corretto. Se esiste mostro la pagina, altrimenti reindirizzo l'utente alla pagina del login. Costruiremo più avanti un

sistema di autenticazione di questo tipo.

Per finire, se avete necessità di distruggere una sessione, ad esempio per fare il logout del nostro sistema di autenticazione basato sulle sessioni, utilizzeremo la funzione `session_destroy()`.