

# Videocorso PHP



## Sessioni e variabili globali

gestiamo le sessioni degli utenti e le  
credenziali

*Alessandro Flora*

# Sessioni e variabili globali

## Come sarà impostata questa lezione

- All'interno delle slide ci saranno alcuni riferimenti utili per il successivo ripasso
- In particolare nelle slide sono riportati i concetti di base e la sintassi
- La seconda parte sarà un esercizio svolto; è vivamente consigliato provare a ragionare insieme creando uno script per provare in prima persona



# Il concetto di sessione

## Cos'è una sessione

- È una sorta di *colloquio* tra client e server e consente al server di identificare il client
- Il client salva sotto forma di cookie un ID di sessione assegnatogli dal server chiamato **PHPSESSID**
- Quando il client invia una richiesta al server usa il PHPSESSID per identificarsi
- Il server, mediante il **PHPSESSID**, accede a delle variabili denominate *di sessione* e può svolgere operazioni su queste

---

```
>> document.cookie
```

```
← "PHPSESSID=tnm5e2vci65h9eq07uasj2oome"
```

---

# Il concetto di sessione

## A cosa serve una sessione

- Può essere utilizzata per salvare i dati dell'utente una volta effettuato il login
- Può essere utilizzata per memorizzare le azioni dell'utente
- Può essere utilizzata per salvare le autorizzazioni (per esempio se un utente è autorizzato o no ad accedere a una pagina dopo la sua identificazione)



# Il concetto di sessione

## Come funzionano le sessioni in PHP

- Nelle pagine in cui vogliamo usare le variabili di sessione dobbiamo invocare la funzione `session_start()`
- Abbiamo ora a disposizione il vettore associativo `$_SESSION`

```
session_start();
```

```
$_SESSION['nome'] = 'Mario';
```

# Il concetto di sessione

## Il vettore \$\_SESSION

- È globale (lo possiamo usare in ogni pagina in cui abbiamo usato **session\_start()**)
- È salvato sul server ma ci si può accedere solamente quando il client effettua la richiesta usando il **PHPSESSID**
- È un vettore associativo che trattiamo come tale

```
session_start();
```

```
$_SESSION['nome'] = 'Mario';
```

# Il concetto di sessione

## Come si distrugge una sessione

- Quando l'utente termina la sua visita sul portale desideriamo che i dati dello stesso siano cancellati dal server (non dal DB) per liberare spazio in memoria
- Per farlo possiamo usare la funzione **session\_destroy()** dopo aver cancellato i campi del vettore **\$\_SESSION** con la funzione **unset()**

```
unset($_SESSION['nome']);
```

```
session_destroy();
```

# Il concetto di sessione

## Esempio d'uso delle sessioni

- L'utente si collega alla pagina di login e inserisce le proprie credenziali
- Tramite degli script senza interfaccia grafica (senza HTML) verifichiamo che l'utente esista e ne preleviamo i dati tramite una query **SELECT**
- Salviamo i suoi dati all'interno del vettore **\$\_SESSION**
- Nella pagina di logout inseriamo lo script di distruzione della sessione ovvero disallochiamo dal vettore **\$\_SESSION** i dati dell'utente e usiamo la funzione **session\_destroy()**



# Il concetto di sessione

## L'indirizzo IP dell'utente

- In alcuni casi potrebbe essere utile salvare l'indirizzo IP dell'utente (per esempio per richiedere una verifica dell'identità nel caso in cui questo cambi)
- Per farlo ci avvaliamo del vettore globale `$_SERVER` alla chiave `REMOTE_ADDR`
- L'indirizzo IP corrisponde all'indirizzo IP del client che sta ponendo la richiesta al server

```
$_SERVER['REMOTE_ADDR']
```

# Le variabili globali

## Visibilità delle variabili

- All'interno di uno script PHP le variabili sono visibili solamente nel loro *blocco* di appartenenza
- Ciò significa che le variabili dentro le funzioni non sono visibili al di fuori e viceversa

```
<?php
```

```
    $a = 3;
```

```
function prova() {  
    $b = 4;  
}
```

```
?>
```

Nello script a fianco la variabile **\$a** non è visibile all'interno della funzione **prova** e la variabile **\$b** non è visibile al di fuori della funzione **prova**

# Le variabili globali

## Il concetto di variabile globale

- Possiamo *promuovere* una variabile inizializzata al di fuori di una funzione per renderla accessibile dentro una funzione usando la keyword **global**

```
<?php  
  
    $a = 3;  
  
    function prova() {  
        global $a;  
        echo $a.'<br>';  
    }  
  
?>
```

In questo caso il codice funzionerà senza dare errori perché specifichiamo che la variabile **\$a** all'interno della funzione è la variabile **\$a** definita al di fuori della stessa

# Le variabili globali

## Quanto sono lecite?

- Il rapporto tra i programmatori e le variabili globali è molto ambiguo
- Spesso sono molto comode ma hanno il difetto di fondo di non far capire dove una variabile è *presa in prestito* (eventualmente in maniera poco legittima, come quando ne si modifica il valore)
- Tendenzialmente è meglio usarle il meno possibile per facilitare quanto più possibile il debug del codice scritto
- In alternativa usiamo il passaggio per riferimento o per valore della variabile a seconda del contesto in cui ci troviamo