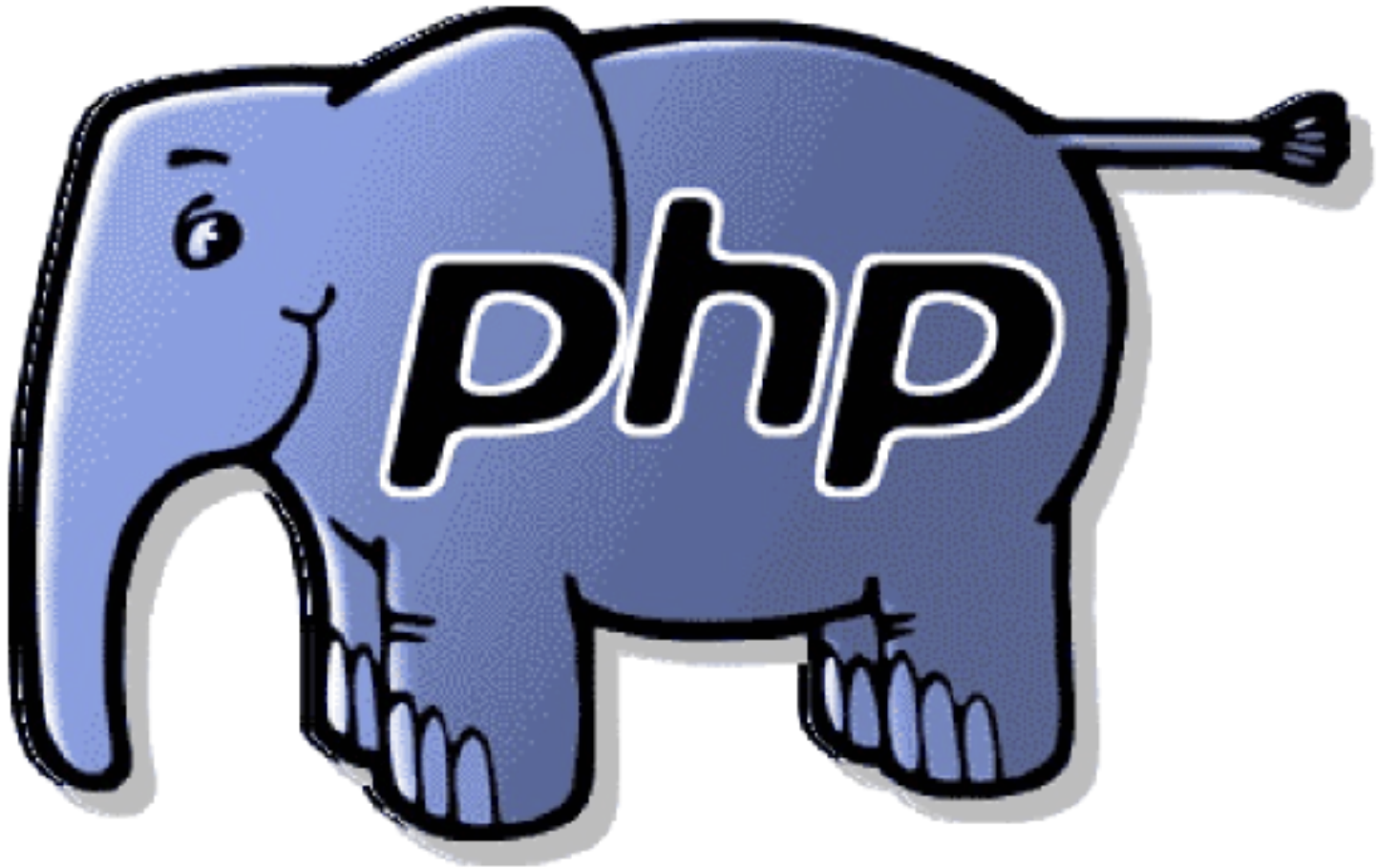
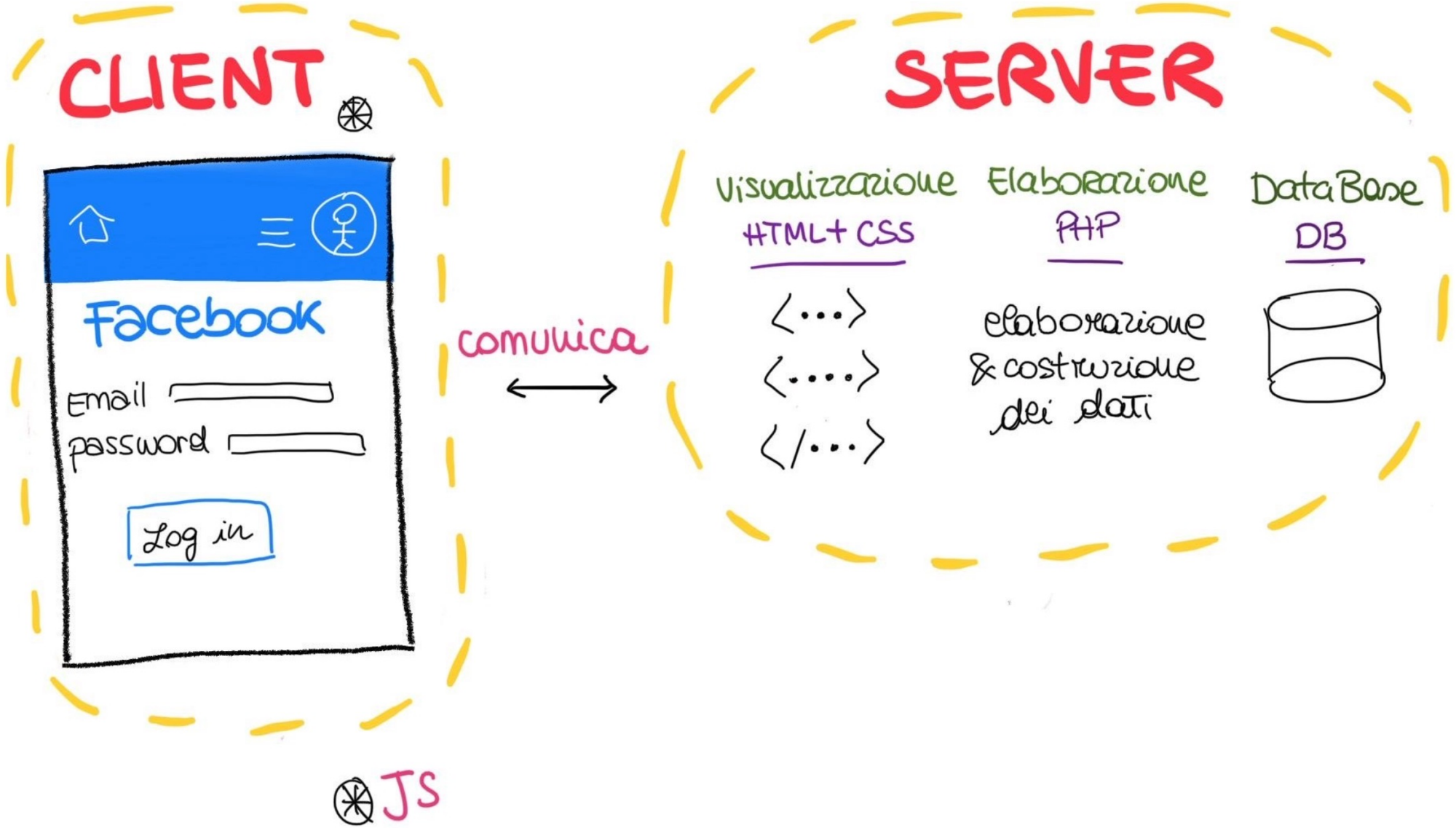
Introduzione al PHP



Una visione più generale



Web Information Service (collegamento interdisciplinare)

- Un Web Information System (WIS) usa le tecnologie Web per permettere la fruizione di informazioni e servizi
- Le architetture moderne dei WIS (ERP) devono essere **estendibili**, per permettere l'impiego di nuove tecnologie e nuove forme di interazione (e.g., interazioni multimodali) + gestiscono **informazioni eterogenee**, come documenti, dati strutturati, risorse multimediali, informazioni semi-strutturate (XML) + permettono **l'interazione con diverse tipologie di sorgenti e di componenti** (architetture multi-tier)

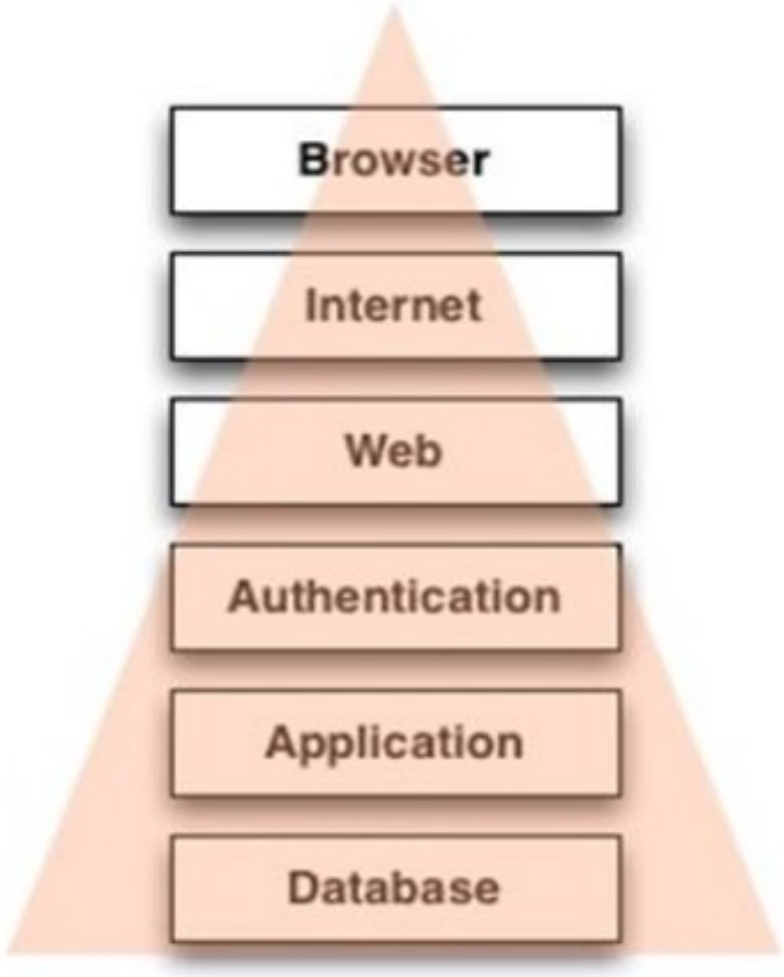
WIS: architettura a più livelli

Ogni *livello* ha un ruolo ben definito

Ogni livello è implementato da uno o più *server*

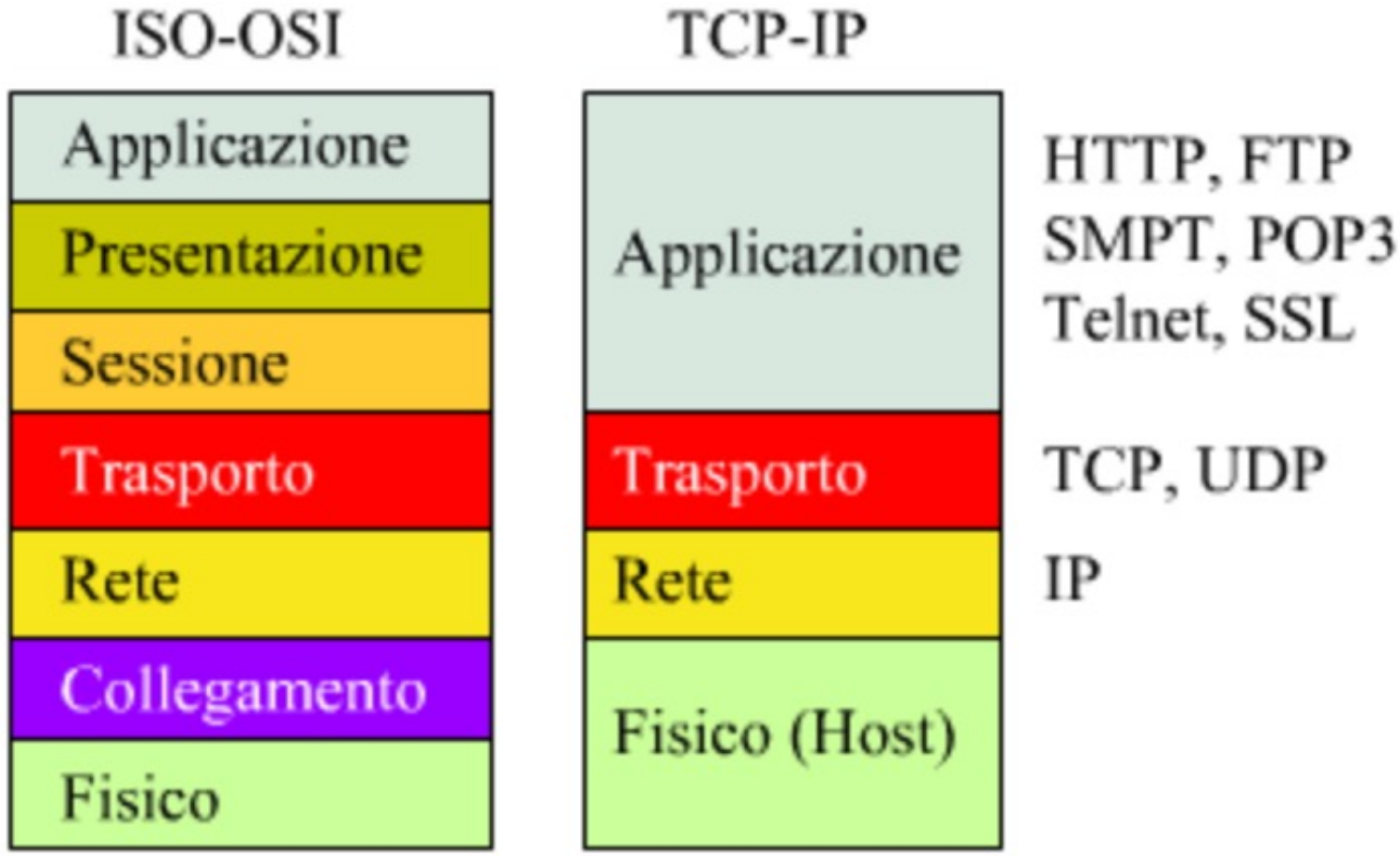
Più server possono condividere lo stesso HW o risiedere su dispositivi dedicati

La *comunicazione* tra i livelli avviene attraverso la rete (HTTP)

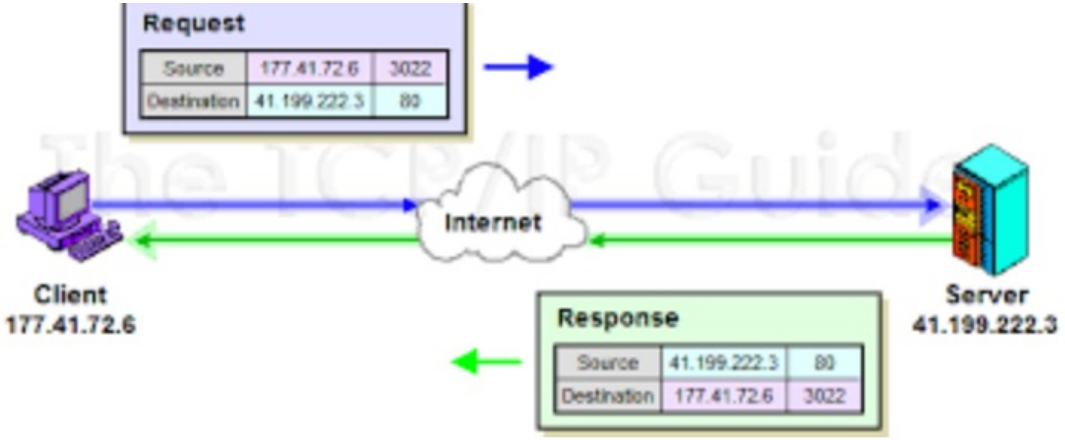


Architettura	N-tier
BROWSER	Chrome, Safari, Opera, Firefox…
WEB	HTML, CSS, JS…
INTERNET	IP, TCP, UDP, HTTP…
ACCESSO	Apache TomCat…
APPLICAZIONE	PHP…
DATABASE	Oracle, MySql…

Ricorda qualcosa?

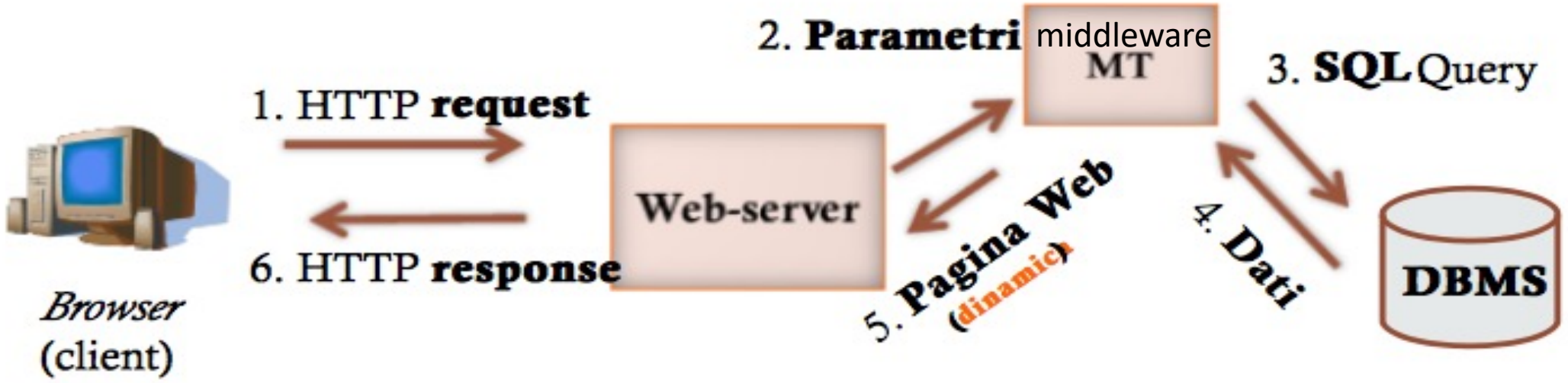


Le componenti:

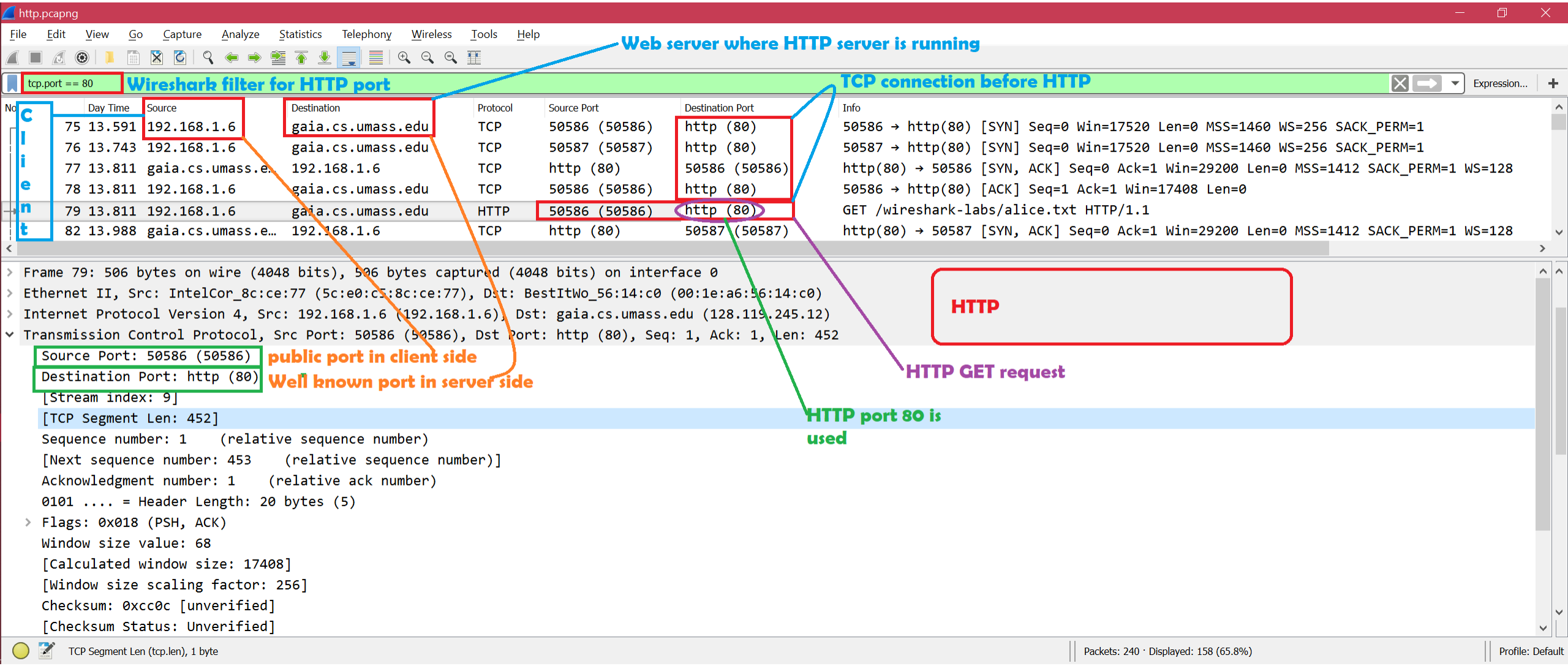


Componentidi un WIS:

- **Web-server**(HTTP-based)
- **DBMS** (relazionale)
- **Meta-tier**di collegamento



Esempio pratico con wireshark



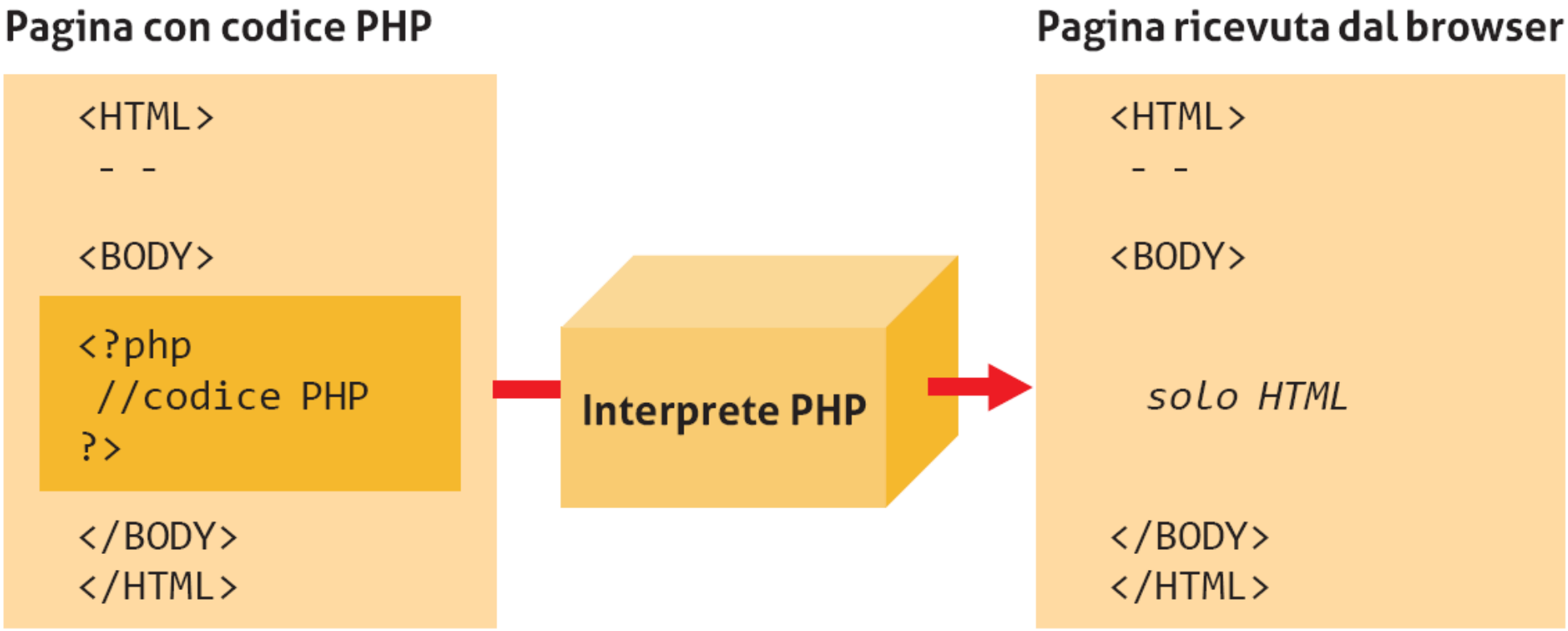
IL PHP:

```
<?php
    // elenco di istruzioni in PHP
?>
```

- PHP è un linguaggio che estende le funzionalità del Web server consentendo l'interpretazione di file con estensione *.php* contenenti il codice dell'applicazione.
- Quando il client richiede una pagina con estensione *.php*, il server Web non spedisce al browser direttamente il file.
- Prima interpreta le istruzioni scritte in PHP, recupera gli eventuali dati richiesti prelevandoli dai file o dai database del server e in seguito restituisce una pagina Web visualizzabile dal browser.
- Poiché questa pagina è costruita al momento della richiesta si chiama pagina dinamica oppure pagina **Web lato server**.

La pagina PHP

La pagina Web ricevuta dal browser non contiene codice PHP, ma solo il codice HTML generato dal server sulla base delle istruzioni PHP.



I vantaggi del php

- Rende più veloce la creazione e lo sviluppo di applicazioni Web
- Facilita la fase di manutenzione e di aggiornamento delle applicazioni
- I suoi script sono compatibili su diverse piattaforme
- Include la possibilità di accedere a vari tipi di database
- Appartiene alla categoria del software libero e viene continuamente controllato e aggiornato.

- Le variabili sono identificate da un nome preceduto dal segno del dollaro $.
- **I nomi delle variabili sono *case-sensitive.***

```
<?php
$eta = 19;                          // contiene un intero
$titolo = "I promessi sposi ";      // contiene una stringa
$prezzo= 31.2;                      // contiene un floating point
?>
```

I principali tipi di variabile

- Numero intero
- Numero a virgola mobile (floating point)
- Stringa
- Valori logici
- Array (numerici e associativi)
- Oggetto

```php
<?php
$nome = "Giovanni";
$saluto = "Buongiorno " . $nome; // contiene: Buongiorno Giovanni
?>
```

Usa il punto (.) come **operatore di concatenazione**

Le solite cose:

```
$X=10;
$Y=$X++;  // Operatore postfisso: si assegna prima il valore della variabile
          // $X alla variabile $Y e poi si incrementa la variabile $X
$X=10;
$Y=++$X;  // Operatore prefisso: prima si incrementa la variabile $X e poi
          // si assegna il valore alla variabile $Y
```

- **operatori aritmetici**

Operatore
+
–
*
/
%
++
––

- operatore di **assegnamento:** il segno di uguale (=)
- operatori di **assegnamento combinati:**

Operatore
+=
–=
*=
/=
%=
.=

Operatore
==
!=
<
>
<=
>=

operatori di **confronto** e operatori **logici**

Operatore	
!	Not
&&	And
||	Or
xor	Xor

Operatori compattati

L'operatore di assegnamento può anche essere **compattato**, cioè **abbinato a un operatore aritmetico**, dando vita a espressioni scritte in forma abbreviata:

$X += $Y	equivale a	$X = $X +$Y	$X /= $Y	equivale a	$X = $X / $Y
$X -= $Y	equivale a	$X = $X – $Y	$X %= $Y	equivale a	$X = $X % $Y
$X *= $Y	equivale a	$X = $X * $Y	$X .= $Y	equivale a	$X = $X . $Y

Operatori relazionali

Gli operatori relazionali permettono di effettuare un confronto tra gli operandi. Richiedono operandi di tipo primitivo e producono sempre un risultato booleano.
Tutti gli operatori presenti in PHP sono identici a quelli del C++: ==, !=, >, >=, <, <=
In più è presente l'operatore:

- === identico a (si scrive con tre simboli "=" senza spazi tra di loro); il risultato è vero se i due operandi sono uguali tra loro e sono **dello stesso tipo**.

Ad esempio:

```
$X = 5; $Y = 5.0; $Z = 3;      // Assegniamo valori a tre variabili diverse
$X == $Y;                      // Vero hanno lo stesso valore
$X === $Y;                     // Falso, perché $X è intero mentre $Y è reale
$Z >= $X;                      // Falso, $Z è minore di $X
```

Finora abbiamo considerato solo variabili con valori numerici. Gli stessi operatori possono essere applicati anche ad operandi di tipo stringa. In questo caso il confronto viene fatto basandosi sull'ordine alfabetico dei caratteri: vale a dire che vengono considerati *minori* i caratteri che *vengono prima* nell'ordine alfabetico. Quindi "a" è minore di "b", "b" è minore di "c", e così via. Inoltre tutte le lettere minuscole sono *maggiori* delle lettere maiuscole, e tutte le lettere (maiuscole e minuscole) sono *maggiori* delle cifre da 0 a 9. Consideriamo i seguenti esempi:

```
$X = 'Matteo'; $Y = 'Marco'; $Z = 'Giovanni'; $V = 'alberto'; $W = '4
gatti';
$X < $Z;  // Falso, poiché la lettera 'G' precede la lettera 'M'
$Y < $X;  // Vero, la 'r' ('Mar') precede la 't' ('Mat')
$Z < $V;  // Vero, la 'a' minuscola è 'maggiore' di qualsiasi lettera
maiuscola
$Z > $W;  // Vero, ogni lettera è 'maggiore' di qualsiasi cifra
```

OSSERVA **COME SI FA**

- **Confrontare i tre costrutti iterativi**

Creare una tabella html contenente Quantità e Prezzo degli articoli presenti in magazzino

Creiamo tre differenti programmi PHP per risolvere il problema. I tre programmi producono lo stesso risultato finale e sono relativi all'utilizzo dei tre costrutti iterativi: *while*, *do…while* e *for*.

Analizziamo il codice con il costrutto *while*.

```
<?php
$PrezzoBase = 5;
$Contatore = 10;

echo "<table style=\"border: 1px solid black; text-align:center;\">";
echo "<tr><th>Quantita'</th>";
echo "<th>Prezzo</th></tr>";
while($Contatore <= 100)
{
   echo "<tr><td>";
   echo $Contatore;
   echo "</td><td>";
   echo $PrezzoBase * $Contatore;
   echo "</td></tr>";
   $Contatore = $Contatore + 10;
}
echo "</table>";
?>
```

Analizziamo il codice con il costrutto *do...while*.

```
<?php
$PrezzoBase = 5;
$Contatore = 10;
echo "<table style=\"border: 1px solid black; text-align:center;\">";
echo "<tr><th>Quantita'</th>";
echo "<th>Prezzo</th></tr>";
do {
   echo "<tr><td>";
   echo $Contatore;
   echo "</td><td>";
   echo $PrezzoBase * $Contatore;
   echo "</td></tr>";
   $Contatore = $Contatore + 10;
} while($Contatore <= 100);
echo "</table>";
?>
```

Analizziamo il codice con il costrutto *for*.

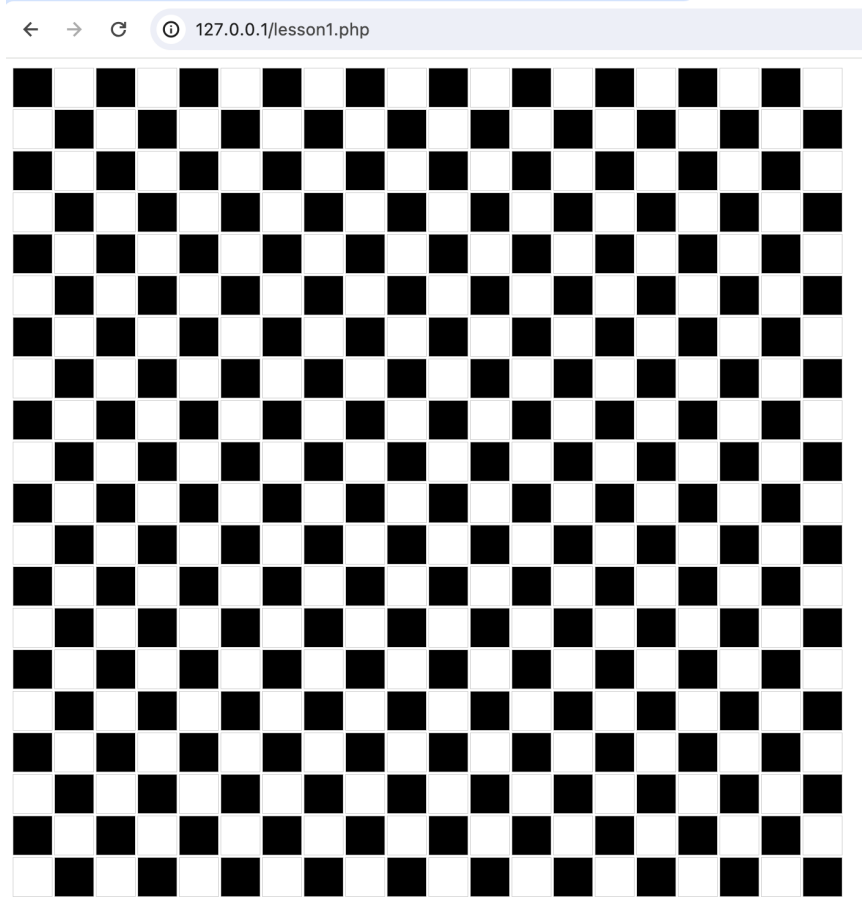
```
<?php
$PrezzoBase = 5;

echo "<table style=\"border: 1px solid black; text-align:center;\">";
echo "<tr><th>Quantita'</th>";
echo "<th>Prezzo</th></tr>";
for($Contatore = 10; $Contatore <= 100; $Contatore += 10)
{
   echo "<tr><td>";
   echo $Contatore;
   echo "</td><td>";
   echo $PrezzoBase * $Contatore;
   echo "</td></tr>";
}
echo "</table>";
?>
```

Il ciclo crea una nuova tabella, con due colonne (una per la *Quantità*, l'altra per il *Prezzo*) finché la variabile *Contatore* non arriva al valore di 100. Quando il valore della variabile *Contatore* supera 100, la condizione fallisce e il ciclo termina. Vediamo in dettaglio che cosa succede:

- vengono impostate le due variabili $*PrezzoBase* e $*Contatore* ai valori iniziali;
- vengono impostati i *tag* iniziali della tabella html e dei suoi *header*;
- viene verificata la condizione del *while*, che è minore di 100, poiché $*Contatore* è pari a 10 (valore iniziale);
- viene eseguito il codice all'interno del *while*, creando una nuova riga per il prezzo pari a 10;
- viene aggiunto 10 alla variabile $*Contatore* per portare il suo valore a 20;
- il ciclo riparte dal passo 3, finché la variabile $*Contatore* sarà maggiore di 100;
- subito dopo il ciclo vengono inseriti i *tag* di chiusura della tabella.

Da notare gli *slash* posti prima dei doppi apici nella prima istruzione *echo* (quelli a sinistra di *border*). Se questi fossero omessi, i secondi doppi apici (quelli posti vicino a *border*) sarebbero interpretati come la chiusura della stringa che l'istruzione *echo* deve visualizzare.

Scacchiera 20x20



lesson1.php

```php
<!DOCTYPE html>
<html>
<head>
    <title>Scacchiera 20x20 </title>
</head>
<body>
    <table >
        <?php
            for ($row = 0; $row < 20; $row++) {
                echo "<tr>";
                for ($col = 0; $col < 20; $col++) {
                    //coloro le celle
                   if(($row + $col) % 2 !== 0)
                    {
                        echo "<td style=\"width: 40px; height: 40px; border: 1px solid #ccc; text-align: center; vertical-align: middle; background-color: white; color: black;\"> </td>";
                    } else {
                        echo "<td style=\"width: 40px; height: 40px; border: 1px solid #ccc; text-align: center; vertical-align: middle; background-color: black; color: white;\"> </td>";
                    }
                }
                echo "</tr>";
            }
        ?>
    </table>
</body>
</html>

```