

---

# PHP funzioni

---

Prof. Francesco Accarino

---

# Lavorare con le date

Con php è possibile mostrare la data corrente(Ora e/o giorno e/o mese e/o anno), sotto qualsiasi forma. La funzione chiave è **date()**; .

tabella di utilizzo:

**d** giorno del mese numerico 01-31

**D** giorno della settimana in abbreviazione di 3 caratteri

**m** mese numerico 01-12

**M** mese in abbreviazione di 3 caratteri

**F** mese in parola

**Y** anno a quattro cifre

**y** anno a due cifre

**H** ore 00-24

**h** ore 00-12

**i** minuti

**s** secondi

I valori che vedete sopra sono i valori da inserire nella funzione date, in questo modo `date("valore");`

Quindi, ad esempio se vogliamo stampare giorno, mese e anno corrente:

**Codice:**

```
echo date("d.m.Y");
```

---

---

# esempio1

```
<html>
  <head>
    <title>La mia prima pagina php</title>
  </head>
  <body>

    <br><br>

    <?php echo date("d-m-Y") ?>

    <br><br>

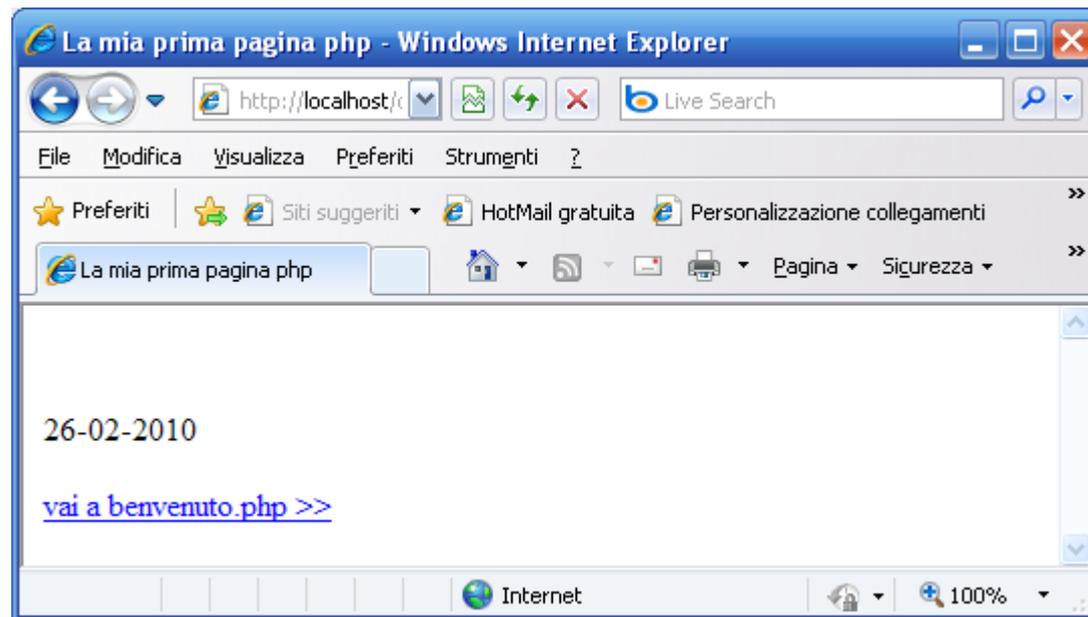
    <a href=" ../cap_2/benvenuto.php">vai a benvenuto.php &gt;&gt;</a>

  </body>
</html>
```

---

# Esempio 1

Ecco come appare la pagina HTML sul navigatore del visitatore.



# esempio1

L'interprete PHP ha fatto il suo dovere. Questo è il codice generato e fornito al navigatore tramite il servente HTTP.



```
http://localhost/cap_1/primo.php - Fonte originale
File Modifica Formatta
1 <html>
2   <head>
3     <title>La mia prima pagina php</title>
4   </head>
5   <body>
6
7     <br><br>
8
9     26-02-2010
10    <br><br>
11
12    <a href="../cap_2/benvenuto.php">vai a benvenuto.php &gt;&gt;</a>
13
14  </body>
15 </html>
```

---

# Funzioni

- Si dichiarano tramite la parola chiave function
- E' possibile indicare un valore predefinito per uno o più parametri: in tal caso la chiamata alla funzione può omettere il passaggio del parametro

```
function nome-funzione( ...parametri... ) {  
... Istruzioni ...  
}
```

```
/* Funzione somma */
```

```
function somma($a, $b) {  
return $a + $b;  
}
```

```
function saluta($utente, $msg = "Ciao") {  
return $msg . " " . $utente;  
}
```

```
$s1 = saluta("Pippo", "Salve"); // $s1 vale "Salve Pippo"
```

```
$s2 = saluta("Pippo"); // $s2 vale "Ciao Pippo"
```

---

---

# Funzioni predefinite

- `phpinfo()` genera dinamicamente un documento contenente informazioni sulla versione di PHP installata e sull'ambiente di esecuzione
    - Può essere utile per controllare quali variabili predefinite sono messe a disposizione del programmatore dall'ambiente PHP
  - `echo()` scrive una stringa nel documento che viene inviato al client
    - può essere usata anche senza parentesi, essendo non una funzione vera e propria ma un costrutto del linguaggio PHP
  - `exit()` e `die()` arrestano l'esecuzione dello script, con la differenza che `die()` consente la stampa di un messaggio (può essere usata per segnalare un errore)
-

---

# Funzioni predefinite

- `isset()` controlla se la variabile contiene o no un valore
- `unset()` cancella una variabile
- `isnull()` verifica se la variabile è di tipo NULL
- Il valore speciale NULL indica che una variabile non ha

valore. Una variabile si considera NULL se:

- le è stata assegnata la costante NULL
  - non le è stato ancora assegnato alcun valore
  - è stata cancellata mediante la funzione `unset()`
-

---

# Variabili globali

- Le variabili globali sono quelle definite all'esterno di una funzione, nella parte principale dello script
- Le variabili locali sono quelle definite all'interno di una funzione
- Per accedere ad una variabile globale dall'interno di una funzione bisogna dichiararla esplicitamente mediante la parola chiave `global`

```
/** $base e' una variabile globale */
```

```
$base = 14;
```

```
function normalizza( $val ) {
```

```
// si dichiara di utilizzare una variabile globale
```

```
global $base;
```

```
return $val/$base;
```

```
}
```

---

---

# Variabili globali

- In alternativa all'uso di `global` è possibile accedere ad una variabile globale dall'interno di una funzione utilizzando l'array associativo `$GLOBALS`

```
/** $base e' una variabile globale */
```

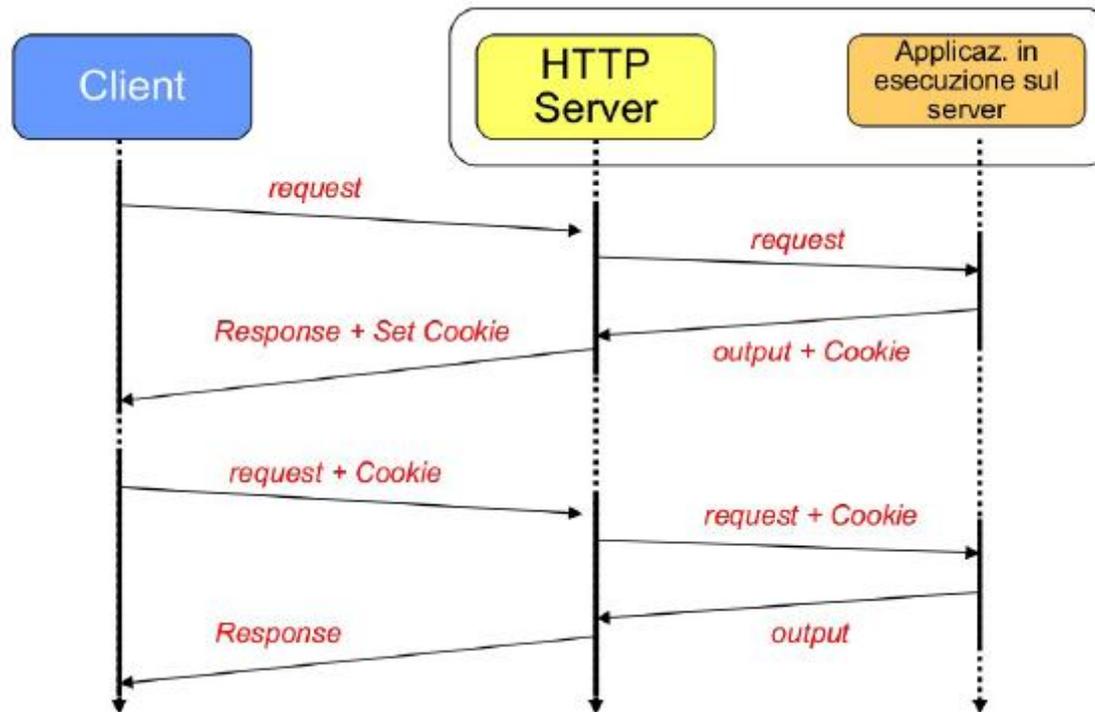
```
$base = 14;
```

```
function normalizza( $val ) {  
return $val/$GLOBALS["base"];  
}
```

---

# COOKIE

- I cookie sono un meccanismo mediante il quale un'applicazione web può profilare l'utente, memorizzando delle informazioni sul client (il browser)



---

# Uso dei cookie

- Tutte le operazioni di scrittura, modifica o cancellazione di cookie in PHP avvengono mediante una stessa funzione, `setcookie( nome [, valore, scadenza] )`
  - Tale funzione deve essere invocata prima che qualsiasi contenuto venga inviato al browser, poichè i cookie vengono trasmessi al client all'interno dell'header della risposta HTTP
  - Il programmatore può accedere ai cookie inviati dal client usando la variabile `$_COOKIE`
-

---

# Uso dei Cookie

Per creare un cookie si usa una semplice funzione di nome `setcookie()`;

La funzione `setcookie` ha la seguente sintassi:

```
setcookie("nomecookie", "valorecookie", "tempocookie");
```

dopodichè per richiamare il cookie basterà stampare la variabile `$_COOKIE` (In questo modo)

## **Codice:**

```
<?php  
echo $_COOKIE['nomecookie'];  
?>
```

---

---

# Uso dei Cookie

```
<?php
$name="Matteo";
setcookie("nome", $name, time() + 3600);
/*Creo un cookie che durerà 3600 secondi, che
si chiama "nome" e che ha come valore il
valore della variabile $name, cioè Matteo*/
echo $_COOKIE['nome']; //Mostro il valore del
cookie
```

---

---

# Usare i Cookie

Per "distruggere" il cookie, cioè eliminarlo prima della scadenza.

Si può impostare un tempo di durata "negativo":

## **Codice:**

```
<?php  
setcookie("nome", "", time() - 3600);  
?>
```

---

---

# Usare i files (esempio conta accessi)

Con php è possibile creare file, eliminarli, modificarli, creare cartelle, eliminare files da cartelle e tante altre cose.

Le principali funzioni sono:

- `fopen($filedaaprire, $mode);` ( Apre un file )
  - `fread($filedaleggere,$numByte);` ( Legge un file )
  - `unlink($daeliminare);` ( Elimina un file )
  - `file_exists($dacontrollare);` ( Controlla se un file esiste)
  - `is_writable($dacontrollare)` (Controlla se un file è riscrivibile)
  - `is_readable($dacontrollare)` (Controlla se un file è leggibile)
  - `fwrite ($fileincuiscrivere , $testodascrivere);` (Scrive un file, eliminando pero' il vecchio contenuto )
  - `fclose($filedachiudere)` ( Chiude un file )
-

# Esempio Chat

The image shows a screenshot of a web browser displaying a chat application. The main window is titled "http://localhost/chat/multiframe.php - Windows Internet Explorer". The chat area shows two messages:

- **prova** : hei
- **franco** : ciao sono franco

Below the chat area, the text "utente: franco" is visible. At the bottom, there is a text input field with the text "Inserisci il messaggio: ciao sono franco" and a button labeled "invia".

A smaller window in the foreground is titled "http://localhost/chat/login.htm - Windows Inte...". It displays the text "Inserisci il tuo NickName" and a text input field containing "franco". To the right of the input field is a button labeled "entra nella chat".

# chat

