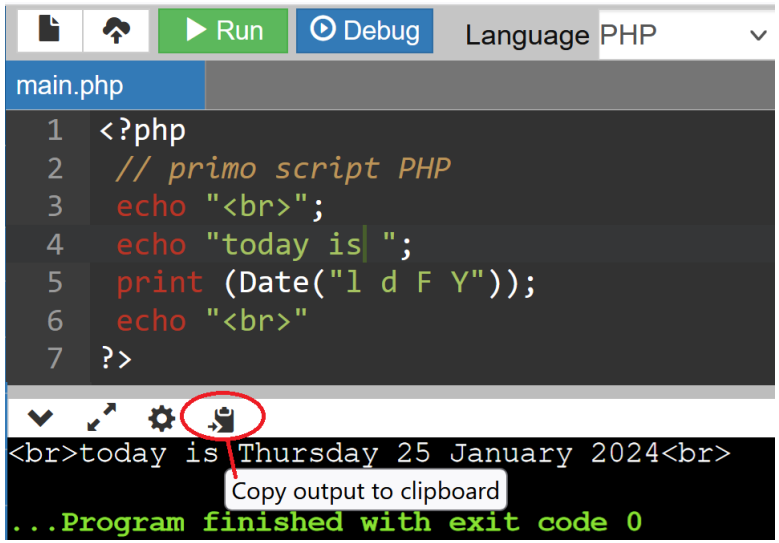
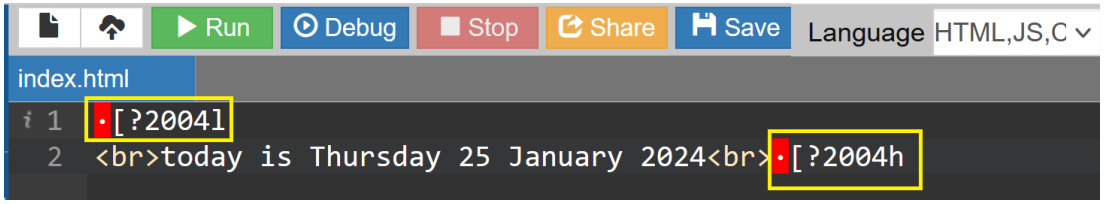
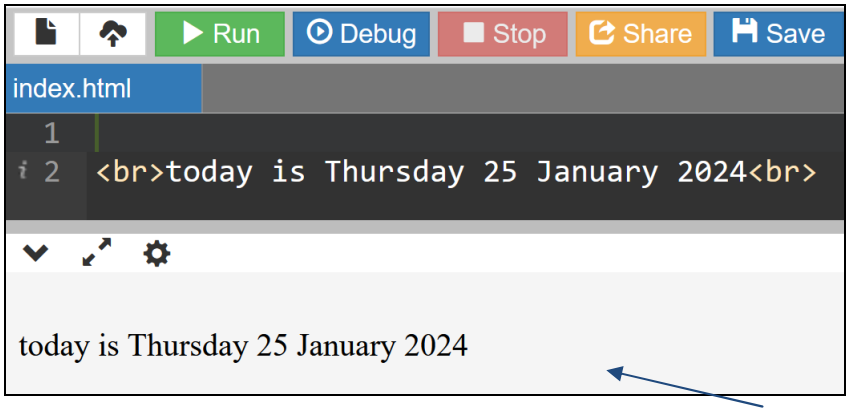
Selezionare in una finstra il linguaggio PHP e in un'altra finestra l'HTML-JS-CSS, scrivere ed eseguire il codice PHP nella prima finestra e poi copiare



l'output e **incollarlo sulla finestra HTML** ed eliminare i caratteri ad inizio e fine (nei rettangoli gialli)





Eseguito lo script HTML, in basso si ottiene la visualizzazione della **pagina HTML statica**

Parametro	Descrizione
Y	anno in 4 cifre
y	anno in 2 cifre
n	mese numero (1-12)
m	mese numero in 2 cifre (01-12)
F	mese testuale ('January' - 'December')
M	mese testuale su 3 lettere ('Jan' - 'Dec')
d	giorno su due cifre (01-31)
j	giorno (1-31)
w	giorno settimanale, numerico (0=dom, 1=lun ecc.)
l	giorno settimanale, testuale ('Sunday', 'monday', ecc.)
D	giorno settimanale in 3 lettere ('Sun', 'mon', ecc.)
H	ora in due cifre (00-23)
G	ora (0-23)
i	minuti (00-59)
s	secondi (00-59)

Le istruzioni **echo** e **print** sono simili, ma print **non** è in grado di gestire più di una stringa per volta; **echo** ammette più stringhe separate da virgola (**,**):

*NOTA: dopo **print** è indicata 1 sola stringa ottenuta dalla concatenazione di 2 stringhe mediante l'operatore punto (**.**)*

Sia **print** che **echo** espandono i nomi di variabili presenti nella stringa (**$**nome) se questa è delimitata dai doppi apici (**"..."**) effettuando la sostituzione del nome con il valore della variabile:

Se invece la stringa è delimitata da apici singoli (**'...'**), sia **echo** che **print** scrivono il nome di variabile così come scritto della stringa.

Una variabile PHP generalmente **non deve essere dichiarata**.

Il nome di una variabile deve essere preceduto dal carattere **$** e può essere formato da lettere, cifre e il carattere underscore (_), ma non può iniziare con una cifra.

$a = 15; // --- valore numerico assegnato alla variabile $a

$a = "Anna"; // --- stringa di caratteri assegnata alla variabile $a

$a = TRUE; // --- costante boolenana assegnata alla variabile $a

NOTA: questi valori possono essere assegnati in sequenza perché la variabile $a non ha un tipo specifico

Per operare sulle variabili si usano operatori aritmetici, relazionali o logici.

operatore	codice php	tipo di variabile
assegnazione	$a = "pluto";	qualsiasi
addizione	$a = $b + $c;	numerico
sottrazione	$a = $b - $c;	numerico
moltiplicazione	$a = $b*$c;	numerico
divisione	$a = $b/$c;	numerico
modulo (resto divisione)	$a = $b%$c;	numerico
incremento di 1	$a++; *è come* $a = $a + 1;	numerico
decremento di 1	$a--; *è come* $a = $a -1;	numerico
uguaglianza	5==5	qualsiasi
disuguaglianza	5!=4	qualsiasi
maggiore, maggiore o uguale	$a>$b $a>=$b	qualsiasi
minore, minore o uguale	$a<$b $a<=$b	qualsiasi
and	($a<1) and ($b>0)	boolean
or	($a<1) or ($b>0)	boolean
not	!$a	boolean
concatenazione	$a="Bella"."monte"	stringa

Definizione di un **array indicizzato (o numerico)**:

$saluto = **array(**'buongiorno' , 'buonasera' , 'buonanotte'**)**;

Oppure:

$saluto[0] = 'buongiorno';

$saluto[1] = 'buonasera';

$saluto[2] = 'buonanotte';

main.php

```php
<?php
$saluto = array('buongiorno' , 'buonasera' , 'buonanotte');
print $saluto[0]."<br>\n";
print $saluto[1]."<br>\n";
print $saluto[2]."<br>\n";
?>
```

input

```
buongiorno<br>
buonasera<br>
buonanotte<br>
```

Contrariamente a quanto accade in altri linguaggi, in PHP gli elementi di un array possono essere di tipi diversi:

main.php

```php
<?php
  $Lista_Spesa = array("Pane", 0.5, "Latte", TRUE);
  $Lista_Spesa[] = "Uova";
  $Lista_Spesa[] = 5;
  $conta = count($Lista_Spesa);
  echo "elementi array: ", $conta, "\n";
  for($i=0; $i<$conta; $i++){
    echo $Lista_Spesa[$i]."<br>\n";
  }
?>
```

input

```
elementi array: 6
Pane<br>
0.5<br>
Latte<br>
1<br>
Uova<br>
5<br>
```

data con giorno della settimana e mese in italiano utilizzando array di stringhe indicizzati

data_ora.php (HTML interno al codice PHP)

```php
<?php
$giorni  = array ("domenica" , "lunedì" , "martedì",
          "mercoledì", "giovedì", "venerdì", "sabato");
$mesi   = array ( "gennaio" , "febbraio" , "marzo" , "aprile",
          "maggio"  , "giugno"  , "luglio" , "agosto" ,
         "settembre" , "ottobre" , "novembre" , "dicembre");
$gsett  = date("w");
$giorno = date("j");
$mese   = date("n");
$anno   = date("Y");
$ore    = date("H");
$min    = date("i");
$temp = $giorni[$gsett]; // elemento array $giorni di posto $gsett
echo "<html>", "\n ", "<body>", "\n";
echo "oggi &egrave; ", $temp, " ", $giorno." ";
$temp = $mesi[$mese];  // elemento array $mesi di posto $mese
echo $temp, " ", $anno, " ore ";
echo $ore,  ":", $min, " <br>\n";
echo "<br><p> data del   giorno </p>";
echo "</body></html>";
?>
```

data_oraH.php (codice PHP contenuto nel codice HTML)

A sinistra il programma che costruisce tutto il codice HTML all'interno del codice PHP; anche l'organizzazione su righe diverse è costruita usando **"\n"** .

A destra il programma che ingloba nel codice HTML il codice PHP necessario a determinare data e ora corrente e a costruire le costanti ed eventuali tag da inglobare nel codice HTML; digitando il codice HTML va usato l'invio (a capo) per porre su righe diverse tag e frasi.

Lettere accentate e spazi nel testo si ottengono con le *html entity* ; es: è e

```php
<html>
 <body>
oggi &egrave;
<?php
$giorni = array ("domenica" , "lunedì" , "martedì",
       "mercoledì", "giovedì", "venerdì", "sabato");
$mesi   = array ( "gennaio" , "febbraio" , "marzo" ,
 "aprile","maggio"  , "giugno"  , "luglio" , "agosto" ,
 "settembre" , "ottobre" , "novembre" , "dicembre");
$gsett  = date("w");
$giorno = date("j");
$mese   = date("n");
$anno   = date("Y");
$ore    = date("H");
$min    = date("i");
$temp  = $giorni[$gsett];
echo $temp, " ", $giorno." ";
$temp = $mesi[$mese];
echo $temp, " ", $anno, " ore ";
echo $ore,  ":", $min, " <br>\n";
?>
<br><p> data del   giorno </p>
</body></html>
```

OUTPUT con con OnlineGDB PHP e poi HTML

```
<html>
 <body>
oggi &egrave; sabato 3 marzo 2024 ore 11:04 <br>
<br><p> data del   giorno </p></body></html>
```

index.html

```
1 <html>
2  <body>
3 oggi &egrave; sabato 3 marzo 2024 ore 11:04 <br>
4 <br><p> data del   giorno </p></body></html>
```

oggi è sabato 3 marzo 2024 ore 11:04

data del giorno

OUTPUT con con OnlineGDB PHP e poi HTML

```
<html>
 <body>
oggi &egrave;
sabato 3 marzo 2024 ore 11:14 <br>
<br><p> data del   giorno </p>
</body></html>
```

NOTE: la codifica speciale per lo spazio o per le lettere accentate viene interpretata dal browser alla ricezione della pagina HTML statica inviata dal server WEB

Per ottenere 3 spazi tra la parola del *e la parola* giorno *è necessario frapporre*

index.html

```
1 <html>
2  <body>
3 oggi &egrave;
4 sabato 3 marzo 2024 ore 11:14 <br>
5 <br><p> data del   giorno </p>
6 </body></html>
```

oggi è sabato 3 marzo 2024 ore 11:14

data del giorno

```php
<?php
 $saluto['mattina'] = 'buongiorno';
 $saluto['sera']    = 'buonasera';
 $saluto['notte']   = 'buonanotte';

  foreach ( $saluto  as  $key => $val ) {
     echo $key .  "\t=> "  . $val .  " <br>\n";
  }
  echo "<br>\n";

 $prov = array (   "Pa" => 'Palermo' ,
                   "Mi" => 'Milano'   ,
                   "To" => 'Torino'   );
  foreach ( $prov  as  $valore)
     echo $valore . "<br>\n";
  echo "<br>\n";

  echo $prov['Pa']  . "<br>\n";
  echo "$prov[To]<br>\n<br>\n";

  echo '<span style="color:green;">'  .  "\n";
  foreach ( $prov   as   $chiave => $valore ) {
     echo '$prov["'  . $chiave .  '"] = \'';
     echo $valore .  "'"  .  "<br>\n";
  }
  echo "</span><br>\n\n";

 $car = array(      "brand"=> "Ford"      ,
                    "model"=> "Mustang" ,
                    "year " => 1964       );

  foreach  ($car  as  $x => $y ) {
     echo "\n$x: $y <br>";
  }
?>
```

Anche gli **array associativi** possono essere costruiti **assegnando valore ai singoli elementi** e indicando la stringa **indice** (detta **chiave**) tra parentesi quadre
oppure
utilizzando il costrutto **array()** in cui vanno elencati i diversi elementi separati da virgola (**,**) indicando per ognuno la **chiave** e il **valore** dell'elemento **associato** mediante l'**operatore di assegnamento =>**

Per scorrere gli elementi di un array associativo si usa l'istruzione **foreach** indicando il **nome dell'array** (esempio **$prov**), seguito da **as** e il nome della variabile che deve contenere il valore dell'elemento (es: **$valore**) eventualmente preceduto dal nome della variabile che deve contenere la **chiave** e dall'**operatore =>**

Il singolo elemento dell'array può essere esposto indicando il valore della **chiave** tra parentesi quadre delimitandolo con apici singoli oppure con doppi apici. ATTENZIONE:
se si vuole scrivere il valore dell'elemento mediante la costruzione di una stringa a doppi apici **" "**, **il valore della chiave non va delimitato**

*NOTA: Nella costruzione di codice HTML all'interno del codice PHP, per le stringhe occorre utilizzare gli **apici singoli** ogni volta che va indicato il **delimitatore HTML del doppio apice***
*oppure se si deve scrivere un **doppio apice** (altrimenti va usata la sequenza di escape **\"*** echo "<span style=\"color:green;\">\n";
*Si usa **\'** per scrivere un apice singolo in una stringa delimitata da **' '** apici singoli oppure l'apice singolo **'** in una stringa delimitata da doppi apici* **"'"**

Gli **array associativi** possono essere utilizzati anche per contenere una **struttura dati** (di tipo diverso: un record) ponendo nella chiave il significato dell'elemento (nome dell'attributo).

riga	codice HTML statico generato dal codice PHP alla pagina precedente
1	mattina => buongiorno
2	sera => buonasera
3	notte => buonanotte
4	
5	Palermo
6	Milano
7	Torino
8	
9	Palermo
10	Torino
11	
12	
13	$prov["Pa"] = 'Palermo'
14	$prov["Mi"] = 'Milano'
15	$prov["To"] = 'Torino'
16	
17	
18	
19	brand: Ford
20	model: Mustang
21	year : 1964

```
foreach ( $saluto as $key => $val )          scrive le prime 3 righe HTML
foreach ( $prov as $valore)                  scrive le righe 5, 6 e 7
echo $prov['Pa'] ."<br>\n";                  scrive la riga 9
echo "$prov[To]<br>\n<br>\n";                scrive la riga 10
foreach ( $prov as $chiave => $valore )      scrive le righe 13, 14 e 15
foreach ($car as $x => $y )                  scrive le righe 19, 20 e 21
```

index.html

```
11  <br>
12  <span style="color:green;">
13  $prov["Pa"] = 'Palermo'<br>
14  $prov["Mi"] = 'Milano'<br>
15  $prov["To"] = 'Torino'<br>
16  </span><br>
```

mattina => buongiorno
sera => buonasera
notte => buonanotte

Palermo
Milano
Torino

Palermo
Torino

$prov["Pa"] = 'Palermo'
$prov["Mi"] = 'Milano'
$prov["To"] = 'Torino'

brand: Ford
model: Mustang
year : 1964

NOTA: per le righe 13, 14 e 15 si sono concatenati nome dell'array, parentesi quadre, i delimitatori doppio apice per i valori delle chiavi, l'uguale, i delimitatori apici singoli per i valori degli elementi dell'array, usando per le stringhe da scrivere con **echo** *i delimitatori compatibili con quelli da usare all'interno.*

Eseguendo il codice HTML si ottiene la pagina qui esposta dove hanno effetto:

- i tag
 che vanno a linea successiva incolonnando i valori
- il tag sulle righe 13, 14 e 15
- l'attributo style="color:green;"

ma **non hanno effetto** le righe 17 e 18 che sono state scritte nel testo mediante

echo "
\n\n";

che va a capo 2 volte per \n\n dopo aver scritto

la riga vuota è effetto del
 così come il corretto incolonnamento degli elementi degli array scritti in precedenza.