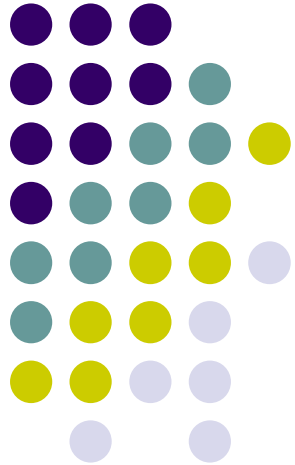
UD7: Tipi di dati in PHP

Fabio Cantaro



I tipi di dati

- 8 tipi di dati primitivi: 4 scalari, 2 composti, 2 speciali
- In php non occorre dichiarare esplicitamente le variabili => assegnazione IMPLICITA

- I tipi scalari:
  - Boolean
  - Integer
  - Float
  - String

Boolean

- Può assumere solo 2 valori TRUE o FALSE

```
<?php
  $variabile=true;
  if ($variabile==true)
            echo “ok”;
//similmente, non occorre per forza uguagliare a  true
 if ($variabile)
            echo “ok”;

?>
```

Conversione Implicita

- La variabile assume il TIPO in base al contenuto:
  - la 1° assegnazione determina il tipo
  - Successive assegnazioni possono modificare il tipo

```
<?php
  $a = “ciao”;                  //$a viene dichiarata STRINGA
  $a = $a +10;                  //$a diventa INTERO  (10)
  $a = $a +0.5;                 //$a diventa FLOAT  (10.5)
  $a = $a + “7  biscotti”;      //$a   resta FLOAT  (17.5)
  $a = 5 + “10ciambelle”;       //$a diventa INTERO (15)
?>
```

Fabio Cantaro

Il tipo INTEGER

- Int prende 4 byte (32 bit) => range $-2^{31} \div 2^{31}+1$
- In caso di overload => diventa automaticamente FLOAT

```
<?php                          //intero.php
      $n =  2147483647;     // in $n ottengo: int(2147483647)
      echo gettype($n);
      $n =  2147483648;     // in $n ottengo: float(2147483648)
      echo gettype($n);
      $m = 1000000;
      $n =  50000 * $m;     // in $n ottengo: float(50000000000)
      echo gettype($n);
?>
```

Integer (2)

- casting da float a integer => arrotondamento

```
<?php
      $n = 1.7;
      $numero=10;
      $numero=(int) ($n);      //Casting esplicito
?>
```

- casting da String a integer => se i primi caratteri sono numeri => Integer altrimenti la variabile diventa String

```
<?php
      $n = 3;
      $n="Ciao a tutti";  //$n diventa String
      $n= 5 + "ciao 35"; //$n diventa  5
      $n= $n + "10biscotti"; //$n diventa  15
      $n= (int) "20.3 ciambelle" //$n diventa String, poi CAST => diventa
      20
?>
```

Float - Double

- Float e Double sono sinonimi
- La dimensione dipende di una variabile float dipende piattaforma: max 64 bit (precisione alla quattordicesima cifra decimale)

```
<?php
      $a=1.234;
      $b=1.2e4;   //significa 1.2 * 10 elevato alla 3
      $c=7E-10;  //significa 7 * 10 elevato alla  -10
      $n = 1+”10.5”;     // $n: float (11.5)
      $n = 1+”-1.3e3”;  //$n: float (-1299)
      $n = 4+(float) ”10.2 abcdef”; //$n: float (14.2)
      $n = “10.0    “+1;                  //$n: Integer (11)
      $n = “20.0 ciambelle“+1.0;    //$n: float (21)
      $n = “ciambelle 20.0“;          //$n: String
      $n = “ciambelle 20.0“ + 2;    //$n: Integer (2)
?>
```

Le Stringhe

- Una stringa è un insieme di caratteri a 8 bit (1 byte) => PHP non supporta UNICODE come Java o Javascript
- Inizializzazione di una variabile Stringa:
  - Usando apice singolo (‘)
  - Usando virgolette (“)
  - Usando Heredoc (<<<) (lo vedremo in futuro)

Fabio Cantaro

Le Stringhe

- Le stringhe sono il tipo di dato **più importante** in PHP.
- Sono sequenze di caratteri
- Le stringhe possono contenere intere frasi e vengono definite utilizzando virgolette **doppie** o **singole**.

Esempio:

```
$citta = “Catania”;
$username = “Lisa”;
```

Le Stringhe

- Le stringhe possono essere concatenate utilizzando il simbolo punto “.”

Esempio:

```
<?php
$Nome = “Mario”;
$Cognome = “Rossi”;

$Nome_completo = $Cognome . “ “ . $Nome;
echo $Nome_completo;
?>
```

Fabio Cantaro

Le Stringhe

- Nelle stringhe possono anche essere inseriti i valori delle variabili.
- Se si inserisce un nome di variabile in una stringa, questo viene sostituito con il suo valore

Esempio:

```
$Nome = “Lisa”;
$Saluto = “Ciao $Nome !”;
echo $Saluto;
```

Ciao Lisa

Fabio Cantaro

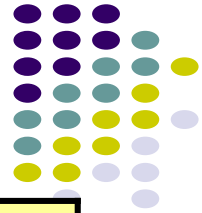
Esercitazione

- Scrivere una pagina HTML (esempio1_bis.htm) con un form in cui sia presente una casella di testo per trasformare la stringa inviata in **tutti i tipi possibili**
- Verrà richiamata una pagina PHP (esempio1_bis.php) che fa il casting della variabile nei 4 tipi

Soluzione(1) (esempio1_bis.htm, esempio1_bis.php)

esempio1_bis.htm

```
<HTML>
<BODY>
<FORM name="trasforma" method="post" action="esempio1_bis.php">
        Variabile da trasformare
    <INPUT type="text" name="variabile"><br>
    < INPUT type="submit" value="<- OK ->">
</FORM>
<BODY>
</HTML>
```



Soluzione(2) (esempio1_bis.htm, esempio1_bis.php)

esempio1_bis.php

```
<?php
$variabile=$_POST['variabile'];
if (isset($variabile))
 {
 echo "Trasformata in integer vale -->".(int) $variabile;
 echo "<BR>";
 echo "Trasformata in float vale   -->".(double) $variabile;
 echo "<BR>";
 echo "Trasformata in boolean vale -->".(boolean) $variabile;
 echo "<BR>";
 echo "Trasformata in string vale  -->".(string) $variabile;
 }
?>
```

Cast Numero -> Stringa

- È possibile convertire un numero in stringa e viceversa utilizzando una operazione di **cast**.
- Un cast si effettua scrivendo tra **parentesi tonde** prima della variabile, il tipo in cui si vuole trasformarla:

Esempio:

```
$Anno = “Anno “ . (string)$year;
$Quantita = (integer)$Qty;
```

- Raramente il casting è necessario. Nella conversione verso numeri può produrre risultati errati.

Fabio Cantaro

Esempio: Apice singolo

```
<?php
	echo 'stringa con apice singolo<BR>';		//stringhe.php

	// volendo inserire il simbolo ' basterà anteporre backslash.
	$frase= 'inserire l\'ennesimo numero';

	echo $frase."<BR>";
	$frase='vuoi cancellare c:\\*.*?';	//otterremo: vuoi cancellare c:\*.*?
	echo $frase;
		echo'<br>';
		echo '$frase';
?>
```

OUTPUT:

stringa con apice singolo
inserire l'ennesimo numero
vuoi cancellare c:*.*?
$frase

Problemi Singolo Apice

- Tutto il contenuto **tra singoli apici** resta INALTERATO
- => tra i singoli apici NULLA viene Interpretato => echo ‘$frase’ non stampa il contenuto della variabile $frase ma solo la scritta *$frase*
- **SOLUZIONE**:
  - **Usare le virgolette** (“)
- => PHP interpreta quanto contenuto tra le virgolette e effettua le eventuali sostituzioni

Caratteri di Escape

- caratteri di escape utilizzabili con le virgolette “

Caratteri di ESCAPE	
\n	newline: ritorno a capo
\r	Carriage return
\t	Tab orizzontale
\\	Backslash
\$	Simbolo del dollaro
\”	Virgolette
_	Simbolo dell’euro

Esempi (1)

```
<?php
   $Civico = 8;
   $Indirizzo = "Via Tespi, $Civico";
   echo 'Il mio indirizzo è $Indirizzo';
?>
```

Quale sarà l'output?

Soluzione:

Il mio indirizzo è $indirizzo

Esempi (2)

```
<?php
   $Civico = 8;
   $Indirizzo = "Via Tespi, $Civico";
   echo “il mio inidirizzo è $Indirizzo";
?>
```

Quale sarà l’output?

Soluzione:

Il mio indirizzo è Via Tespi, 8

Esempi (3)

- Problema: Voglio fare comparire a video una cosa del tipo:

Il valore della variabile $Indirizzo è Via Tespi, 8

Fabio Cantaro

Proposta 1

- Il seguente script cosa restituirebbe?

```
<?php
   $Civico = 8;
   $Indirizzo = "Via Tespi, $Civico";
   echo "Il valore della variabile $Indirizzo è $Indirizzo";
?>
```

Soluzione:

Il valore della variabile Via Tespi, 8 è Via Tespi, 8

=> NON VA BENE!!! Non è quello che mi aspettavo

Proposta 2

- Il seguente script cosa restituirebbe?

```
<?php
   $Civico = 8;
    $Indirizzo = "Via Tespi, $Civico";
    echo 'Il valore della variabile $Indirizzo è $indirizzo';
?>
```

Soluzione:

Il valore della variabile $Indirizzo è $indirizzo

=> NON VA BENE!!! Non è quello che mi aspettavo

Proposta 3

- Il seguente script cosa restituirebbe?

```
<?php
   $Civico = 8;
   $Indirizzo = "Via Tespi, $Civico";
   echo 'Il valore della variabile $Indirizzo';
   echo " è $Indirizzo";
?>
```

Soluzione:

Il valore della variabile $Indirizzo è Via Tespi, 8

⇒**OK!!! Questo funziona!!!**

⇒**PROBLEMA: UN PO' COMPLICATO!!**

Proposta 4

- Utilizzare il carattere di escape **\$** quando voglio visualizzare $indirizzo:

```
<?php
    $Civico = 8;
    $Indirizzo = "Via Tespi, $Civico";
    echo "Il valore della variabile \$Indirizzo è $Indirizzo";
?>
```

⇒ **$indirizzo non viene interpretata**

⇒**OK!!! Questo funziona!!!**

⇒**È Anche più semplice!!**

Esercizio

- Voglio fare comparire a video l'indirizzo tra VIRGOLETTE:

Il valore della variabile $Indirizzo è "Via Tespi, 8"

Soluzione: usare il carattere di escape **\"**

```
<?php
    $Civico = 8;
    $Indirizzo = "Via Tespi, $Civico";
    echo "Il valore della variabile \$Indirizzo è \"$Indirizzo\"";
?>
```

Fabio Cantaro

Backslash nelle stringhe

- se mettiamo un backslash:
  - davanti a Virgolette in una stringa delimitata da apici singoli
  - (o viceversa) davanti ad apici singoli in una stringa delimitata da Virgolette
- => anche il backslash entrerà a far parte della stringa stessa

```
ESEMPIO:
<?php
   echo "Torniamo un\'altra volta";
?>
```

Torniamo un\'altra volta

Fabio Cantaro

Altro esempio (1)

- può capitare che una stringa debba contenere a sua volta un **apice** o un paio di **virgolette**

```
<?php

echo 'Torniamo un\'altra volta';   //stampa: Torniamo un'altra volta
echo "Torniamo un'altra volta";    //stampa: Torniamo un'altra volta

echo "Torniamo un\'altra volta"; //stampa: Torniamo un\'altra volta
echo 'Torniamo un'altra volta'; /*causa un errore, perchè l'apostrofo viene scambiato per l'apice di chiusura*/

echo 'Anna disse "Ciao" e se ne andò';    /*stampa: Anna disse "Ciao" e se ne andò*/
echo "Anna disse \"Ciao\" e se ne andò"; /*stampa: Anna disse "Ciao" e se ne andò*/

echo 'Anna disse \"Ciao\" e se ne andò'; /*stampa: Anna disse \"Ciao\" e se ne andò*/
echo "Anna disse ""Ciao"" e se ne andò"; //errore

?>
```

← **Stesso effetto**

Stesso effetto

Fabio Cantaro

Altro esempio

• Il backslash viene usato anche come 'escape di sè stesso', nei casi in cui vogliamo esplicitamente includerlo nella stringa

```
<?php

echo "<br>Questo: \"\\\" è un backslash"; /*stampa: Questo: "\" è un backslash*/
echo '<br>Questo: \'\\\' è un backslash';  /*stampa: Questo: '\' è un backslash*/
echo "<br>Questo: '\' è un backslash";   /*stampa: Questo: '\' è un backslash*/
echo "<br>Questo: '\\' è un backslash";   /*stampa: Questo: '\' è un backslash*/
?>
```

Questo: "\" è un backslash
Questo: '\' è un backslash
Questo: '\' è un backslash
Questo: '\' è un backslash

EREDOC(1)

- poco utilizzata se non in caso di stringhe molto lunghe
- La stringa deve essere delimitata dai caratteri **<<<** seguiti da un identificatore (per convenzione **EOD**) + Stringa + EOD;

<<< EOD Stringa EOD;

Esempio

```
<?php
    $nome = "Paolo";
    $stringa = <<<EOD
    Il mio nome è $nome
    EOD;
    echo $stringa;
?>
```

Il mio nome è Paolo

EREDOC(2)

- heredoc risolve i nomi di variabile così come le virgolette.
- Rispetto a queste ultime vantaggio:possiamo includere delle virgolette nella stringa senza farne l'escape

Fabio Cantaro

Esempio

```
<?php
        $frase = "ciao a tutti";
        $stringa = <<<EOT
        Il mio saluto è "$frase"
        EOT;
        echo $stringa;
    ?>
```

Il mio saluto è "ciao a tutti"

Esercizio

- Variabili:
  - $sito= www.unict.it
  1. $naviga="Questo è il sito dell'università: _____"

1. A video deve stampare:
   1. La variabile $naviga contiene: "Questo è il sito dell'università: www.unict.it"

Una Soluzione

Con le virgolette

```
<?php
$sito = "www.unict.it";
$naviga="Questo è il sito dell’università: $sito ";
echo "La variabile \$naviga contiene: \"$naviga\""
?>
```

Con gli apici

```
<?php
$sito = "www.unict.it";
$naviga="Questo è il sito dell’università: $sito ";
echo "La variabile \$naviga contiene: '$naviga' "
?>
```

Funzioni di esistenza

- importanti funzioni per la gestione delle variabili:
  - **isset()**: verifica se una variabile è definita
    (attenzione: definita, non valorizzata !)
  - **unset():** distrugge la variabile specificata
    (come avverte il manuale, in PHP4 questa non è più una vera funzione, pertanto non restituisce nulla)
  - **empty():** verifica **se una variabile è valorizzata**

ISSET()

- **ISSET()** Restituisce **TRUE** se la *variabile* esiste; **FALSE** se è NULL
- Le variabili non devono essere dichiarate prima dell'utilizzo.
  - => **IsSet()** è una funzione che esamina una variabile per vedere se le è stato assegnato un valore

Esempio

```
<?php
      $set_var = 0; // è stato assegnato un valore
      if(IsSet($set_var))
          print("set_var è impostata.<BR>");
      else
          print("set_var non è impostata.<BR>");
      if(IsSet($never_var))
          print("never_var è impostata.<BR>");
      else
          print("never_var non è impostata.<BR>");
?>
```

```
set_var è impostata.
never_var non è impostata.
```

UNSET()

- **UNSET()** distrugge la variabile specificata:
- L'atto di assegnare una variabile **non è irrevocabile => unset()** riporta una variabile ad uno stato di **non assegnamento**, senza badare all'assegnazione precedente

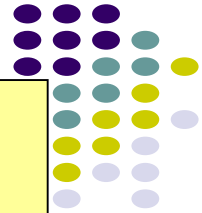
Esempio

```
<?php
$set_var = 0; // è stato assegnato un valore
unset($set_var); // revochiamo tutto!
if(IsSet($set_var))
    print("set_var è impostata.<BR>");
else
   print("set_var non è impostata.<BR>");
?>
```

set_var non è impostata.

ISSET(), UNSET() Esempio

Fabio Cantaro



```php
<?php
$var = ' ';   // il carattere BLANK è un valore.
if (isset($var))
    echo "Questa variabile è valorizzata, pertanto scrivo.";

$a = "test";
$b = NULL;
echo "<BR>";
echo isset($a);      // TRUE stampa 1
echo isset($b);     // FALSE => non stampa niente

unset ($a);          //ora è FALSE
echo isset($a);  // => non stampa niente
?>
```

```
Questa variabile è valorizzata, pertanto scrivo.
1
```

EMPTY()

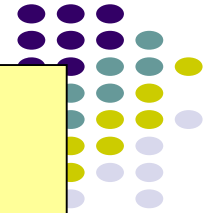
- Determina se una variabile ***var*** è da considerare vuota

Valori restituiti

- Restituisce **FALSE** se ***var*** non è vuota ed ha un valore diverso da zero.
- I seguenti valori sono considerati vuoti:
  - ***""*** *(stringa vuota)*
  - ***0*** *(0 come intero)*
  - ***"0"*** *(0 come stringa)*
  - ***NULL***
  - ***FALSE***
  - ***array()*** *(matrice vuota)*

Esempio

Fabio Cantaro



```php
<?php
$var1 = 0;

// Valutata come true perchè $var è vuota
if (empty($var1)) {
    echo '$var1 contiene 0';
}
$var2 = NULL;
if (empty($var2)) {
    echo '$var2 contiene NULL';
}
$var3 = "Ciao";
if (empty($var3)) {
    echo '$var3 contiene CIAO';
}
?>
```

Non viene stampato

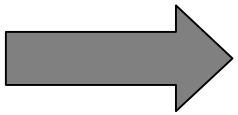
$var1 contiene 0$var2 contiene NULL

Funzioni di Verifica Tipo

- bool **settype** ($variabile, type): forza il tipo di "$variabile" secondo il tipo "Type"
- void **var_dump** ($expression): visualizza informazioni riguardanti una espressione (o variabile), tra cui il suo tipo e il suo valore
- string **gettype** ($variabile): Restituisce il tipo della variabile (boolean, Integer, Double, String, Object, NULL)

Esempio 1: settype, gettype

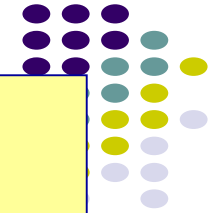
```
<?php
      $a = "50.5 Kg pane";
      $s1 = "800 Hg di farina";
      $s2 = "Ciambelle";
      $b = True;
      settype($a,"double");              // $a ora è float 50.5
      echo $a." tipo: ".gettype($a)."<BR>";
      settype($s1,"integer");            // $s1 ora è integer 800
      echo $s1." tipo: ".gettype($s1)."<BR>";
      settype($s2,"integer");            // $s2 ora è integer 0
      echo $s2." tipo: ".gettype($s2)."<BR>";
      settype($a, "integer");            // $a ora è 5   integer
      echo $a." tipo: ".gettype($a)."<BR>";
      settype($b, "string");             // $b ora è "1" stringa
      echo $b." tipo: ".gettype($b)."<BR>";
?>
```



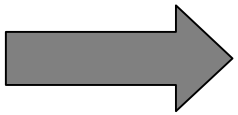
```
50.5 tipo: double
800 tipo: integer
0 tipo: integer
50 tipo: integer
1 tipo: string
```

Esempio 2: var_dump

Fabio Cantaro



```
<?php
        echo "ciao!";
        $a = "ciao!";
        $b=123;
        echo "il secondo carattere è: {$a{2}} ";
        echo "<BR><BR>";
        echo "<B>";
        var_dump($a);    //$a è stringa => suo valore è sua
    Lunghezza (5)
        echo "<BR>";
        var_dump($b); //$b è Integer => suo valore è suo
    Contenuto
?>
```



```
ciao!il secondo carattere è: a

string(5) "ciao!"
int(123)
```