

Il linguaggio PHP

Corso Backend System Integrator
Modulo **Programmazione PHP**

Docente: Dott. Enrico Zimuel

in collaborazione con:



REGIONE
PIEMONTE

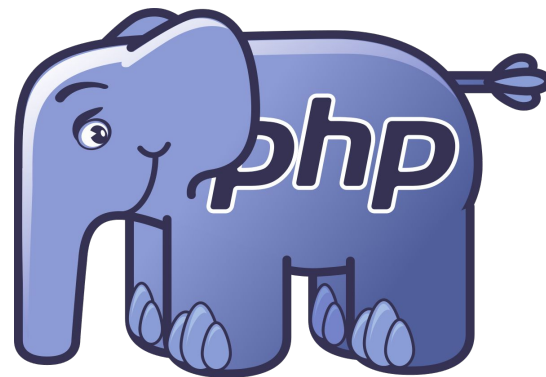
per una crescita intelligente,
sostenibile ed inclusiva

www.regione.piemonte.it/europa2020

INIZIATIVA CO-FINANZIATA CON FSE

Introduzione al PHP

- Hello World
- Sintassi di base: variabili, tipi di dati, commenti, costanti, costanti “magiche”
- Tipi di dati
 - integer
 - double (float)
 - string
- Basi numeriche, formule matematiche
- Concatenazione e funzioni sulle stringhe



Hello World

```
<?php  
echo "Hello World\n";  
?>
```

Non obbligatorio

Sintassi di base

- Tutti gli script PHP iniziano con **<?php**
- Possono terminare o meno con **?>** (di solito omissivo)
- L'istruzione **echo** stampa una stringa
- La stringa da stampare può essere specificata tra **apici** (') o **doppi apici** ("): con gli apici singoli la stringa non viene interpretata; nel nostro caso doppi per la presenza della sequenza di escape (**\n**)
- ogni istruzione PHP termina con il carattere di **punto e virgola** (;)

Apici singoli o doppi?

```
<?php  
echo "Hello World\n";  
echo 'Hello World\n';  
echo 'Hello World\n';
```

```
Hello World  
Hello World\nHello World\n
```

- ```
<?php
phpinfo();
```

- [illegible]

# Commenti

- Esistono diversi modi per commentare il codice PHP
  - Singola linea: Iniziando un commento con la stringa `//` oppure con il carattere `#`
  - Su più linee: utilizzando la stringa d'inizio `/*` e la stringa `*/` di fine commento

```
<?php
/*
Esempio di commento
*/

echo "Hello World\n"; // Esempio di commento
// Esempio di commento
echo "Hello World\n"; # Esempio di commento
Esempio di commento
```

# Sui commenti...

- Se da un lato un codice senza commenti può risultare difficile da comprendere, troppi commenti o commenti superflui possono risultare controproducenti
- Cercate di utilizzare i commenti solo dove servono!



# Tipi di dati in PHP

- boolean (true/false)
- integer
- double
- string
- array
- object
- resource
- NULL

# Variabili

- Le variabili in PHP hanno sempre il prefisso dollaro (\$)
- Ogni nome di variabile deve iniziare con una lettera dell'alfabeto o un underscore (\_) e può proseguire con qualsiasi lettera, numero o underscore
- Ad esempio sono validi \$a, \$\_a1, \$foo
- Non sono validi \$1a, \$!\_a, \$-a

## Variabili (2)

- I nomi di variabili sono *case-sensitive*, ad esempio `$foo` e `$Foo` sono due variabili distinte
- Le **variabili in PHP non sono tipizzate\***, ossia è possibile assegnare ad una stessa variabile un tipo di dato differente ([type juggling](#))

```
<?php
$a = 'ciao';
$a = 1;
$a = [7, 2, 3];
```

(\*) Dal PHP 7.0 stati introdotti i tipi per i parametri di ingresso e uscita delle funzioni.  
Dal PHP 7.4 sono stati introdotti i tipi per le proprietà di una classe.

# gettype()

- La funzione [gettype\(\)](#) consente di restituire il tipo di una variabile

```
<?php
$a = 'ciao';
echo gettype($a); // string
$a = 1;
echo gettype($a); // integer
```

# Costanti

- Le costanti in un linguaggio di programmazione sono dei valori che non possono essere modificati durante l'esecuzione
- In PHP una costante è definita con la funzione [define\(\)](#)
- Per convenzione, le costanti sono denominate in maiuscolo

```
<?php
define ('VERSIONE', '1.0.0');
echo VERSIONE; // utilizzando il nome
echo constant('VERSIONE'); // con la funzione constant()
```

# Costanti predefinite

- Il PHP ha alcune costanti predefinite, ad esempio:
  - `PHP_VERSION`, restituisce la versione del PHP con la sintassi "major.minor.release[extra]"
  - `PHP_OS`, il sistema operativo utilizzato
  - [Qui](#), l'elenco completo delle costanti predefinite del PHP

# Esercizio

- Cosa succede se provo a definire una costante utilizzando lo stesso nome di una predefinita del PHP?

```
<?php
define ('PHP_VERSION', '1.0.0');
echo PHP_VERSION;
```

PHP Notice: Constant PHP\_VERSION already defined

# Costanti “magiche”

- Il PHP ha alcune costanti dette "magiche" perchè variano a seconda del file e della linea d'utilizzo
- Sono denominate con dei doppi *underscore* **\_\_NAME\_\_**, dove NAME può essere ad esempio:
  - **\_\_FILE\_\_**, restituisce il percorso del file dove è memorizzato il codice PHP
  - **\_\_DIR\_\_**, la directory del file
  - **\_\_LINE\_\_**, il numero della riga corrente nel file
  - [Qui](#), l'elenco completo delle costanti magiche del PHP



# Esercizio

- Qual è il risultato dell'esecuzione del seguente codice?

```
<?php
echo "File : " . __FILE__ . "\n";
echo "Directory: " . __DIR__ . "\n";
echo "Linea : " . __LINE__ . "\n";
```

```
File : /home/enrico/ITS Piemonte/PHP Programming/2020-2022/test.php
Directory: /home/enrico/ITS Piemonte/PHP Programming/2020-2022
Linea : 4
```

# INTEGER

- Le variabili di tipo **integer** in PHP contengono valori numerici interi
- Le costanti **PHP\_INT\_MAX**, **PHP\_INT\_MIN** e **PHP\_INT\_SIZE** determinano rispettivamente il numero massimo supportato, quello minimo e la dimensione in byte del tipo integer

# Esercizio

- Qual è il numero **massimo**, **minimo** e la **dimensione** in byte supportata dalla vostra versione del PHP?

```
<?php
echo "Max : " . PHP_INT_MAX . "\n";
echo "Min : " . PHP_INT_MIN . "\n";
echo "Byte : " . PHP_INT_SIZE . "\n";
```

```
Max : 9223372036854775807
Min : -9223372036854775808
Byte : 8
```

# Basi numeriche

- Oltre al sistema decimale, si possono utilizzare numeri binari, ottali e esadecimali

```
$num = 255; // decimale
$num = 0b11111111; // binario
$num = 0377; // ottale
$num = 0xff; // esadecimale
```

# DOUBLE

- Il tipo **double** (o **float**) è utilizzato per la gestione dei numeri in virgola mobile

```
<?php
$a = 1.234; // 1.234
$b = 1.2e3; // 1200
$c = 7E-10; // 0.0000000007
```

# Operazioni matematiche

Operazioni matematiche elementari in PHP:

- somma (+)
- sottrazione (-)
- divisione (/)
- moltiplicazione (\*)
- modulo o resto della divisione (%)
- elevamento a potenza (\*\*)

# Esercizio

- Scrivere un programma in PHP per effettuare i seguenti calcoli:
  - $234 + 45.6$
  - $1234 - 23.12$
  - $6789 / 123$
  - $234 \times 3451$
  - il resto della divisione tra 3452 e 245
  - $345^{12}$
  - stabilire se questi numeri sono pari o dispari: 12, 45, 37, 24

# Incremento e decremento

Operatori di incremento (++) e decremento (--)

```
<?php
$a = 1;
echo ++$a; // 2
echo $a++; // 2
echo $a; // 3
$a = 3;
echo --$a; // 2
echo $a--; // 2
echo $a; // 1
```



# Funzioni matematiche

- Il PHP offre diverse funzioni matematiche
- Ad esempio:
  - **sqrt(\$x)**, radice quadrata
  - **sin(\$x)**, **cos(\$x)**, seno e coseno
  - **log(\$x)**, logaritmo naturale
  - **round(\$x)**, arrotondamento al numero intero, per eccesso se la parte decimale  $\geq 0.5$  per difetto altrimenti.  
Esempio: `round(3.4)=3`, `round(3.5)=4`
- [Qui](#) l'elenco completo delle funzioni matematiche

# Esercizio

- Scrivere un programma PHP per effettuare i seguenti calcoli:
  - Convertire il numero decimale **121** in binario
  - Convertire il numero decimale **34526** in esadecimale
  - Convertire il numero decimale **324** in ottale
  - Se il **seno** di un angolo  $\alpha$  è **0.9438**, quanto vale il coseno di  $\alpha$ ?  
Utilizzare la formula di trigonometria:  $\sin(\alpha)^2 + \cos(\alpha)^2 = 1$

# STRING

- Il tipo string viene utilizzato per la memorizzazione delle stringhe, ossia sequenze di caratteri [ASCII](#)
- Le stringhe vengono racchiuse tra apici singoli o doppi
- Gli **apici singoli** si utilizzano quando non ci sono variabili o caratteri di escape
- Gli **apici doppi** (o **virgolette**) vengono utilizzati per interpretare la stringa, con caratteri di escape (ad esempio il ritorno a capo `\n`) e sostituendo le variabili all'occorrenza

# Esempio: apici singoli e doppi

```
<?php
$a = 1;
echo "Il valore di a è $a\n";
echo 'Il valore di a è $a\n';
```

```
Il valore di a è 1
Il valore di a è $a\n
```

# Concatenazione

- L'operatore di concatenazione di stringhe in PHP è il **punto** (.)

```
<?php
$name = 'Enrico';
$surname = 'Zimuel';
$fullname = $name . ' ' . $surname;
echo $fullname; // Enrico Zimuel
```

## Concatenazione (2)

- E' possibile concatenare una stringa con un'altra utilizzando l'operatore di assegnamento `.=`

```
<?php
$name = 'Enrico';
$name .= ' Zimuel'; // equivale a $name = $name . ' Zimuel'
echo $name; // Enrico Zimuel
```

# Funzioni sulle stringhe

- Il PHP offre numerose funzioni per la manipolazione di stringhe

```
$text = 'Esempio di testo';
echo strlen($text) . "\n"; // 16
echo strpos($text, 'testo') . "\n"; // 11
echo substr($text, 11, 5) . "\n"; // testo
echo strtoupper($text) . "\n"; // ESEMPIO DI TESTO
echo strtolower($text) . "\n"; // esempio di testo
```

- [Qui](#) l'elenco completo delle funzioni sulle stringhe

# Esercizio

- Utilizzando le funzioni stringa del PHP scrivere un programma per convertire la stringa “**Nel mezzo del cammin di nostra vita**” in [camelCase](#)
- Risultato atteso:  
**nelMezzoDelCamminDiNostraVita**





# Grazie dell'attenzione!

Per informazioni:

[enrico.zimuel@its-ictpiemonte.it](mailto:enrico.zimuel@its-ictpiemonte.it)