Introduzione PHP

1 - Vediamo un File PHP

Un file php per essere pre-processato correttamente deve avere estensione .php e rispettare le seguenti regole:

- Sempre tag di apertura;
- Se il file è tutto php si può anche omettere il tag di chiusura;
- Ad ogni fine statement (Dichiarazione) va sempre il punto e virgola.

```
<!-- index.php -->

<?php

echo "Francesco";
```

2 - Apriamo un File PHP nel terminale

È possibile eseguire script PHP all'interno del terminale;

Basta verificare innanzitutto che PHP sia installato sul vostro sistema con:

```
php -v
```

E lanciare:

```
php index.php
```

3- Variabili & Costanti

Una variabile è una locazione di memoria in cui vengono salvati dei dati. Una variabile è formata dal nome e dal valore che contiene.

- Per definire una variabile in PHP si inizializza con il $ seguito dal nome della variabile.

```
<?php

$variabile = 'File Index';

const VERSION = 1.0;// Non è obbligatorio il maiuscolo ma fortemnte consiglia



echo $variabile;
echo VERSION;
```

- Ci sono delle regole però per dichiarare una variabile:

```
//1)Deve iniziare con il simbolo $

$age = 18;//Si
@age = 18;//No
£age = 18;//No

//2)Può contenere solo caratteri alfanumerici e l'underscore _

$actual_age2022 = 18;//Si
$actual.age2022 = 18//No
$actual-age2022 = 18//No
$age!2022 = 18//No

//3)Il nome della variabile deve iniziare con una lettera dell'alfabeto o con l'underscore. Non può iniziare con un numero e non può contenere spazi v uoti
$_actual_age = 18;//Si
$1age = 18//No
$actual age = 18//No

//4) Le variabili sono case sensitive, ciò significa che

$age = 18;
$AgE = 19;
```

```
$Age = 20;
$agE = 21;
$AGE = 22;

//5) Come detto ad inizio lezione, le variaili non sono tipizzate. La stessa va
riabile può cambiare durante il corso del programma (gettype):

$ageGreat18 = true;
$ageGreat18 = 1;
$ageGreat18 = 'Si';
```

4 - Data Types

Esistono 2 macro categorie:

- **Predefiniti o scalari**
- **Composti o user-defined**

Prima di iniziare introduciamo un comando utile per capire cosa stiamo facendo:

```
gettype($var);
```

4.1 Predefiniti o scalari

- **Integer**: Rappresentano tutti i numeri interi positivi e negativi;

```
$age = 18;
$year = 2022;
$points = -5;
```

- **Float (Double)**: Rappresentano tutti i numeri a virgola mobile positivi e negativi. Obbligo di inserire il punto e non la virgola.

```
$celsius = 45.6;
```

```
$price = 1022.89;
```

- **String**: Sequenza di caratteri (**ASCII Code 256 Caratteri**) alfanumerici, speciali e stringa vuota, delimitata da singoli o doppi apici;

```
//Stringa

$text = 'Ciao';
$string3 = Esempio di Testo;// Stringa non conforme
$name = 'Francesco';

//Concatenare una stringa

echo $text . ' ' . $name;



//Apici
$string1 = "Esempio di Testo";//Con i doppi apici PHP prova ad intepretare i
l contenuto della stringa ed effettua eventualemnte degli escape.

$string2 = 'Esempio di Testo';//Apici singoli vengono utilizzati per inserire u
n testo cosi com'è.




//Esempio

$a = 1;

$text1 = 'Esempio di Testo con valore di a: $a';
$text2 = "Esempio di Testo con valore di a: $a";
$text3 = 'Esempio di Testo con valore di a: '. $a; //Workaround
echo $text . "\n";


//Problema apici singolo
```

```
$problem1 = 'L'esempio di Testo con valore di a: $a';
$problem2 = 'L\'esempio di Testo con valore di a: $a';
```

- **Booleani**: Il tipo di dato booleano può rappresentare solo due valori: true e false (vero e falso).

```
$bool = true;
$status = false;
```

- **NULL**: Il tipo di dato booleano può rappresentare solo due valori: true e false (vero e falso).

```
$null = null;
```

4.1.1 - Type Coercion

```
//Somma Tra interi
$x = 10;
$y = 11;

echo $x + $y; //21

$x = false;
$y = 12;

echo $x + $y;//12

$x = "12sdhjsdhs";
//$x = "fra12";
$y = 12;

echo $x + $y;//12

//Alfabeto

$a = 'a';
$a++;
```

```
echo $a;
```

4.2 - Composti o user-defined

I tipi di dati composti sono diversi dai primitivi perché possono contenere più valori di tipi diversi.

Array: Un array è lista ordinata di elementi. Possono essere dello stesso tipo o di tipi diversi.

```
$array = ['Lun','Mar','Mer','Gio','Ven'];//Omogenei

count($array);//5


echo $array;//gestire errore

print_r($array);


$array2 = ['Ciao',2,34.67,['Lun','Mar','Mer','Gio','Ven']];//eterogenei

var_dump($array2);
```

Aggiungere un elemento in un array

```
$array = [];

$array[] = 1;


print_r($array);
```

Ogni valore in un array è contrassegnato da un indice. Grazie all'indicizzazione, possiamo accedere agli elementi degli array. Gli array in

PHP sono 0-Based. Il primo elemento dell'array ha indice 0. L'ultimo elemento dell'array ha come indice n-1 dove n = lunghezza dell'array.

```
$array2 = ['Ciao',2,34.67,['Lun','Mar','Mer','Gio','Ven']];


print_r($array2[1]);//2

print_r($array2[3][3]);//Mer
```

Array Associativo: Gli array associativi sono coppie di elementi la cu chiave funge da identificativo. Esempio Dizionario.

```
$array = [
'w' ⇒ 3,
'l'⇒ 0];

$array = [
2022 ⇒ 'done',
2023⇒ 'loading'];



print_r($array['w']);//3


---

$array = [];

$array['elemento'] = 1;


print_r($array);
```

Oggetti: Li vedremo più avanti

5 - Operatori

Gli operatori servono ad effettuare delle operazioni sugli operandi
Si dividono principalmente in tre tipi:

- matematici;
- di confronto;
- logici.

5.1 - Operatori Matematici

Agiscono su uno o più operandi e restituiscono un valore numerico.

Aa Title	Descrizione	Column 2
$a + $b	Somma	Somma di $a e $b.
$a - $b	Sottrazione	Differenza of $a e $b.
$a * $b	Moltiplicazione	Prodotto of $a e $b.
$a / $b	Divisione	Quoziente of $a e $b.

5.2 - Operatori di Confronto

Gli operatori di confronto agiscono su due operandi e restituiscono un valore booleano: true o false.

Utilizziamo :

```
$a = 10;
$b = 2;
```

```
var_dump($a == $b);
```

Aa Title	Column 1	Column 2
$a == $b	Uguaglianza	true if $a is equal to $b after type juggling.
$a === $b	Identico	true if $a is equal to $b, and they are of the same type.
$a != $b	Diverso	true if $a is not equal to $b after type juggling.
$a <> $b	Dverso	true if $a is not equal to $b after type juggling.
$a !== $b	Completamente Diverso	true if $a is not equal to $b, or they are not of the same type.
$a < $b	Minore di	true if $a is strictly less than $b.
$a > $b	Maggiore di	true if $a is strictly greater than $b.
$a <= $b	Minore o uguale di	true if $a is less than or equal to $b.
$a >= $b	Maggiore o uguale di	true if $a is greater than or equal to $b.

5.3 - Operatori Logici

Agiscono su due operandi e restituiscono un valore booleano: true o false.

Aa Title	Column 1	Column 2
$a and $b	And	true if both $a and $b are true.
$a && $b	And	true if both $a and $b are true.
$a or $b	Or	true if either $a or $b is true.
$a || $b	Or	true if either $a or $b is true.
! $a	Not	true if $a is not true.

```
$a1 = true;
$b1 = true;
```

```
$a2 = true;
$b2 = false;

$a3 = false;
$b3 = true;

$a4 = false;
$b4 = false;


OR:
var_dump($a1 or $b1);
var_dump($a1 || $b1);

AND:
var_dump($a1 and $b1);
var_dump($a1 && $b1);

NOT:
var_dump(!$a1);
var_dump(!$b1);
```

6 - Condizione TRUTHY e valori FALSY

- **Truthy**: Un dato è **truthy** quando è quantificabile e contiene un valore non nullo o diverso da 0.
  - Numeri != da 0;
  - Stringhe != da "";
  - Array non vuoti.
- **Falsy**: Un dato è **falsy** quando contiene un valore nullo o non quantificabile.
  - 0;
  - Stringa vuota "";
  - Array vuoto [];
  - Il valore false.
  - Null