PHP

Php a differenza di JS, Css e HTML non si svolge sul client, ma **sul server**.
Serve per rendere dinamiche le nostre pagine web, aggiungendo tutte le funzionalità che girano sul lato Server.

E' un linguaggio OOP (*Object Oriented Programming)*, ed e' un **linguaggio interpretato** quindi viene eseguito al momento.
Php sta per **Hypertext PreProcessor**, inizialmente era personal home page.
Il nome ci fa capire che serve a processare i contenuti di una pagina HTML prima che questa venga restituita al cliente.

Perche' lo usiamo?
- interagisce con il database e embeddato nell'HTML
- possiamo creare delle sessioni
- possiamo settare i cookies
- aiuta a criptare i dati e a creare validazioni
- supporta diversi protocolli come HTTP SMTP e POP3 ecc..
- può' gestire i form

Dalla *bash* creiamo un file ***index.php*** e apriamolo tramite il comando ***code index.php***

Il tag di apertura e chiusura di di ogni documento php è il seguente.
(il tag di chiusura è evitabile se nel documento viene utilizzato esclusivamente codice php)

```
<?php

?>
```

Per visualizzare il nostro contenuto in una pagina HTML come testo scriviamo sul terminale e click sul link:

```
php -S localhost:8000
```

SINTASSI e COMANDI

NB. *Il punto e virgola* ***;*** *è fondamentale per dichiarare la fine dell'istruzione data.*
Il linguaggio è a tipizzazione debole, quindi non dobbiamo come in JS specificare il tipo di dato da inserire al suo interno.

La ***variabile*** in php si dichiara con il simbolo **$**

```
$nomeVariabile = "Francesco";
```

Per ***stampare*** *stringhe o primitivi*:

```
echo $nomeVariabile;
```

Stampare con il *var_dump()* permette di avere piu' dati:

- il Tipo primitivo della variabile
- il numero di elementi da cui è composto
- il contenuto della variabile

```
$int = "35"
var_dump($int)
//result string(2) "35"
```

Per ***visualizzare a terminale*** scrivere al suo interno (shortcut: tasto SU + invio):

```
php index.php
```

Per dichiarare una ***costante*** (sempre in maiuscolo):

```
const NOME_COSTANTE = "Massimiliano";
```

Per **concatenare** due elementi:

```
$nome = "Massimiliano";
echo "Mi chiamo ".$nome;
echo "Mi chiamo $nome";
// result Mi chiamo Massimiliano
```

Gli apici doppi **" "** sono ***diversi*** dagli apici singoli **' '**
I ***doppi*** servono per inserire stringhe di testo al cui interno possono essere contenuti anche altri elementi (es. delle variabili come sopra), mentre i ***singoli*** trasformano in stringa tutto il contenuto al loro interno.

Un altro tipo di assegnazione è l'**IDT**:

```
$nomeVariabile = <<<IDT
Testo a caso con anche simboli e apici?>L>":<"?<
IDT;
```

L'***accapo*** si usa con il comando ***n*** e funziona solo se usato dentro agli apici doppi:

```
echo "Ciao sono Massimiliano\n"
```

readline() permette l'inserimento di una linea di testo da parte dell'utente in risposta a quello scritto nelle parentesi.

```
readline("Inserisci un colore: ")
//result Inserisci un colore: ____________
```

DATA TYPES

Tipi Primitivi

```
$string = "Francesco";
$bool = true;
$int = 35;
$float = 32.346;
```

Gli apici doppi non sono uguali agli apici singoli. gli apici singoli trasformano in stringa tutto cio' che c'e' all'interno.

Dati Compositi (User-Defined)

- Array

Tramite il *var_dump* possiamo visualizzare meglio il contenuto di un array.
L'indice dell'array **[0]** parte da 0, il Tipo Primitivo e il Contenuto.
E' possibile anche stampare con il ***print_r*** per avere un risultato simile senza il Tipo Primitivo.

```
$nomeArray = ["string", 34, 45, 43, true]
var_dump($nome_array)
//result array(4) {
//    [0] =>
//    string(6) "string"
//    [1] =>
//    int(34)
//    [2] =>
//    int(45)
//    [3] =>
//    int(43)
//    [4] =>
//    bool(true)
//    }
```

Per richiamare Array contenuti all'interno di altri Array si utilizza la stessa meccanica utilizzata in JS.

- Array Chiave-Valore

Negli *array chiave-valore* esplicito un valore di index ai dati al loro interno.

```
$nomeArr [
"nome" => "Massimiliano",
"cognome" => "Falcone",
"age" => 29
];
```

NB. Stiamo attenti a dare nomi giusti ai valori, soprattutto quando si danno dei numeri.