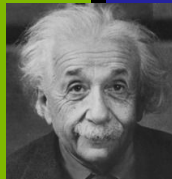


# B - PHP

## ESo1 - PHP Hello World

**FB - 09.2024 - v1.3**



**“La teoria è quando si sa tutto e niente funziona. La pratica è quando tutto funziona e nessuno sa il perché. Noi abbiamo messo insieme la teoria e la pratica: non c'è niente che funzioni... e nessuno sa il perché!”**

# Hello World

## Creazione di semplici applicazioni web con php ([sito2](#))

Sito in costruzione. Saluti dal prof. Filippo Bilardo.

[Vai al pannello di controllo](#)

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

PHP 8.0.26 - Optimized for AlterVista



Build Date	Dec 15 2022 04:05:54
Virtual Directory Support	disabled
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930,NTS
PHP Extension Build	API20200930,NTS
Debug Build	no

# Indice

- [Obiettivi didattici](#)
- [Introduzione teorica](#)
- [Esercitazione](#)
- [Valutazione](#)

# Obiettivi didattici

---

# Obiettivi didattici



- le variabili in php e i costrutti di base del linguaggio

# Introduzione teorica

---



# Il protocollo HTTP

# Il protocollo HTTP

HTTP (Hypertext Transfer Protocol) è un protocollo di comunicazione utilizzato per la trasmissione dei dati su Internet. Viene utilizzato per trasmettere le richieste e le risposte tra un client (ad esempio, un browser web) e un server (ad esempio, un sito web).

Il funzionamento di HTTP si basa sull'invio di messaggi di richiesta e di risposta tra il client e il server. Il client invia una richiesta al server utilizzando uno dei metodi HTTP (GET, POST, PUT, DELETE, ecc.), che indica l'azione da eseguire, e il server risponde inviando un messaggio di risposta contenente l'esito dell'azione richiesta.

HTTP utilizza un modello client-server in cui il client inizia la richiesta e il server risponde, il client può anche ricevere un cookie dal server per la gestione delle sessioni e l'autenticazione.

HTTP è un protocollo di livello applicazione, è stato creato per essere utilizzato insieme ad altri protocolli, come TCP (Transmission Control Protocol), che si occupa della trasmissione dei dati su una rete.



# Il protocollo HTTP

HTTP/2 è l'ultima versione del protocollo, essa introduce nuove funzionalità come la capacità di inviare più richieste contemporaneamente, la compressione dei dati e l'uso di un solo connessione per più richieste, migliorando le performance rispetto alla versione precedente, HTTP/1.1.

# Richiesta

Ecco un esempio di una richiesta HTTP inviata da un client (ad esempio, un browser web) al server (ad esempio, un sito web):

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/89.0.4389.82 Safari/537.36
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
```

La prima riga della richiesta contiene il metodo HTTP (GET), l'URL della risorsa richiesta (in questo caso, /index.html) e la versione del protocollo (HTTP/1.1).

Le righe successive contengono i campi dell'intestazione, che forniscono informazioni aggiuntive sulla richiesta. Ad esempio, il campo "Host" indica il nome del server a cui la richiesta è indirizzata, il campo "User-Agent" indica il browser utilizzato dal client e il campo "Accept-Language" indica la lingua preferita dal client.

# Risposta

Ecco un esempio di una risposta HTTP inviata dal server al client:

```
HTTP/1.1 200 OK
Server: Apache/2.4.37 (Unix)
Content-Type: text/html
Content-Length: 1234
Last-Modified: Tue, 12 Jan 2021 12:34:56 GMT
```

```
<!DOCTYPE html>
<html>
<head>
<title>Example Website</title>
</head>
<body>
<h1>Welcome to our website!</h1>
<p>Here is some sample content.</p>
</body>
</html>
```

La prima riga della risposta contiene la versione del protocollo (HTTP/1.1), il codice di stato (200) e una breve descrizione del codice di stato (OK).

Le righe successive contengono i campi dell'intestazione, che forniscono informazioni aggiuntive sulla risposta. Ad esempio, il campo "Server" indica il tipo e la versione del server che ha generato la risposta, il campo "Content-Type" indica il tipo di contenuto restituito e il campo "Content-Length" indica la lunghezza del contenuto restituito.

Infine, la parte successiva della risposta contiene il corpo della risposta, che è il contenuto effettivo restituito dal server. Nel caso di una pagina web, questa parte conterrà il codice HTML della pagina.



# Introduzione a PHP



- **PHP Hypertext Preprocessor** (acronimo ricorsivo)

- Linguaggio di **scripting interpretato**
- Principalmente usato per applicazioni web lato server

- **Caratteristiche**

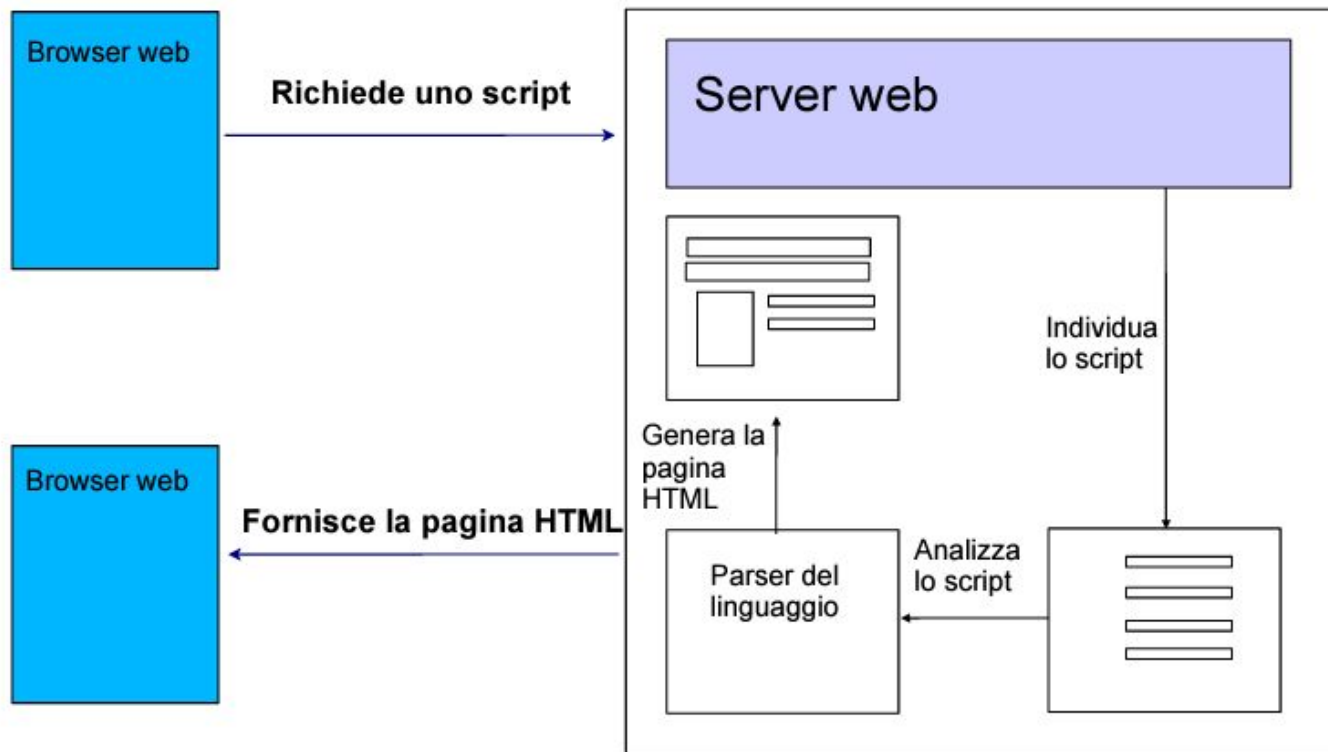
- open source
- numerose librerie (grafica, mail, ...)
- componenti per l'interazione con vari DB (MySQL, Oracle, Postgres, ...)
- ...



- La sintassi di PHP è simile a quella di C, C++ e Java (ma senza tipi)
- Tutto il codice PHP va messo all'interno di tag `<?php .... ?>`
  - ▶ il tag `<?php .... ?>` può essere usato più volte all'interno dello stesso file
  - ▶ tutto ciò che si trova al di fuori di tali tag viene dato in output così com'è
- Come C, C++ e Java:
  - ▶ I comandi sono separati da ;
  - ▶ PHP è case-sensitive (distingue tra maiuscole e minuscole)
- PHP supporta commenti a singola linea e multilinea con la stessa sintassi di C++ e Java:

```
<?php
// Questo e' un commento che verra' trascurato
// dall'interprete PHP....

/* .... tanto quanto questo commento,
   che puo' occupare tranquillamente
   piu' linee
*/
?>
```





- I parametri di funzionamento di PHP sono definiti nel file **php.ini** che il server web legge ad ogni riavvio
- Tipicamente i valori di default sono ok `phpinfo()`
- Esempio:
  - **display\_errors**: impostato su 'on' mostra gli errori 'sul browser'.
  - **max\_execution\_time**: tempo concesso per l'esecuzione di uno script, dopo il quale si blocca (def. 30s). Utile se esistono loop su cicli errati.
  - **session\_save\_path**: Questo parametro indica la cartella nella quale PHP salva i files di sessione:
  - **short\_open\_tag**: default a 'on', ci permette di aprire e chiudere codice PHP con i tag `<? e ?>`.





Pagina con  
estensione .php

Parte statica  
(HTML)

Parte dinamica  
(PHP)

```
<html>
<head>
  <title>Esempio PHP</title>
</head>
<body>
<p>Buongiorno, mi chiamo
<?php
  $username="<b>Nessuno</b>";
  echo $username;
?>
signor Polifemo</p>
</body>
</html>
```

Polifemo.php  
Polifemo.html



- Il codice PHP si interviene al codice HTML, delimitato dai tag

`<?php ... ?>`

- I delimitatori permettono al server web di riconoscere il codice, che viene eseguito
- L'output viene combinato con la parte statica che sta al di fuori dai tag PHP, e il tutto è inviato al browser (client)
- Per "eseguire" lo script, devo richiederlo al web-server tramite il browser

<http://localhost/miapagina.php>

- Sintassi molto simile al C ...



# I file php e i tag <?php e ?>

1. `<?php echo 'if you want to serve PHP code in XHTML or XML documents, use these tags'; ?>`
2. You can use the short echo tag to `<?= 'print this string' ?>`.  
It's equivalent to `<?php echo 'print this string' ?>`.
3. `<? echo 'this code is within short tags, but will only work '  
'if short_open_tag is enabled'; ?>`

```
INFORMATICA_3_PHP > www > ES01 > 🐞 esempio01.php
```

```
1  <?php
2  echo '<h1>Ciao ragazzi ';
3  echo 'benvenuti nel mio corso PHP!</h1>';
4  ?>
5
```

204.216.213.176/inf3php/ES01/esempio01.php

**Ciao ragazzi benvenuti nel mio corso PHP!**



- L'istruzione di stampa di PHP è echo (o print)

- `<?php echo "Ciao"; ?>`

- oppure

- `<?php echo("Ciao"); ?>`

- oppure

- `<?php print "Ciao"; ?>`

- <https://www.php.net/manual/en/function.print.php>
- <https://www.php.net/manual/en/function.printf.php>



- Tre sintassi per i commenti:

- **/\* commento a  
riga multipla \*/**
- **// commento a riga singola**
- **# commento a riga singola**

# PHP è case sensitive e whitespace insensitive

PHP è un linguaggio di programmazione case sensitive, il che significa che le lettere maiuscole e minuscole sono considerate diverse. Ad esempio, la variabile `$nome` e la variabile `$Nome` sono due variabili diverse in PHP. Tuttavia, ci sono alcune eccezioni a questa regola. Ad esempio, i nomi delle funzioni in PHP non sono case sensitive, il che significa che la funzione `count()` e la funzione `COUNT()` sono equivalenti.

Inoltre, PHP è insensibile agli spazi bianchi, il che significa che gli spazi bianchi (come gli spazi, i tab e le nuove righe) non influiscono sulla semantica del codice PHP. Ad esempio, il seguente codice PHP è equivalente:

`$nome='Mario' ;` è equivalente a `$nome = 'Mario' ;`

Questa regola non vale per le stringhe. Gli spazi bianchi all'interno di una stringa sono considerati parte della stringa stessa.



# Esempio

# Esempio - file hello.php



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

Browser window showing the output of the PHP script `hello.php`. The page displays "Hello World!" and the PHP Version 8.2.5.

**PHP Version 8.2.5**

System	Linux 26bd8989bbcd 5.15.0-1035-oracle #41-Ubuntu SMP Wed May 3 14:18:23 UTC 2023 aarch64
Build Date	May 3 2023 04:22:48
Build System	Linux d59d02c02da5 5.10.0-19-cloud-arm64 #1 SMP Debian 5.10.149-1 (2022-10-17) aarch64 GNU/Linux
Configure Command	'./configure' '--build=aarch64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqld' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--disable-phpdbg' '--with-pear' '--with-libdir=lib/aarch64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=aarch64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-pdo_mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829,NTS





# Variabili e costanti



# Variabili e costanti

In PHP, le variabili sono contenitori di valori che possono essere modificati durante l'esecuzione del programma. Le costanti, invece, sono contenitori di valori che non possono essere modificati durante l'esecuzione del programma.

Per dichiarare una variabile in PHP, si utilizza il simbolo \$ seguito dal nome della variabile. Ad esempio:

```
$nome_variabile = valore;  
$miavar; $_ENV; $var34;
```

Per dichiarare una costante in PHP, si utilizza la funzione define() o la parola chiave const. Ad esempio:

```
define("NOME_COSTANTE", valore);  
const NOME_COSTANTE = valore;
```

Le costanti definite con define() possono essere utilizzate ovunque nel codice, mentre le costanti definite con const possono essere utilizzate solo all'interno della classe in cui sono definite



- **Nota:** distinzione tra nome della variabile ed accesso alla variabile ...
  - Permette di usare **variabili variabili**
    - `$a="b";`
    - `$b=1`
    - `echo $$a`
- ... produce 1
- Potente, ma pericoloso ...



- Lo scope di una variabile in PHP è la pagina stessa
- Ogni variabile esiste solo per lo script (pagina) in cui è definita e alla fine dell'esecuzione viene distrutta
- Si può verificare se una variabile è stata inizializzata con la funzione `isset`:

```
isset($var)
```

restituisce `true` se `$var` è stata precedentemente assegnata, `false` altrimenti

- E' possibile distruggere una variabile precedentemente assegnata tramite la funzione `unset`:

```
unset($var)
```



## Variabili predefinite

- PHP mette a disposizione un vasto numero di variabili (array) predefinite
- visibili globalmente, senza dichiarazione esplicita, non devono essere istanziate (**superglobali**)
  - \$GLOBALS: tiene traccia di tutte le var. globali
  - \$\_SERVER: variabili definite dal web server
  - \$\_ENV: variabili d'ambiente
  - \$\_GET, \$\_POST: variabili di un form HTML
  - \$\_REQUEST = \$\_GET unione \$\_POST
  - \$\_COOKIE: cookie
  - \$\_SESSION: variabili di sessione

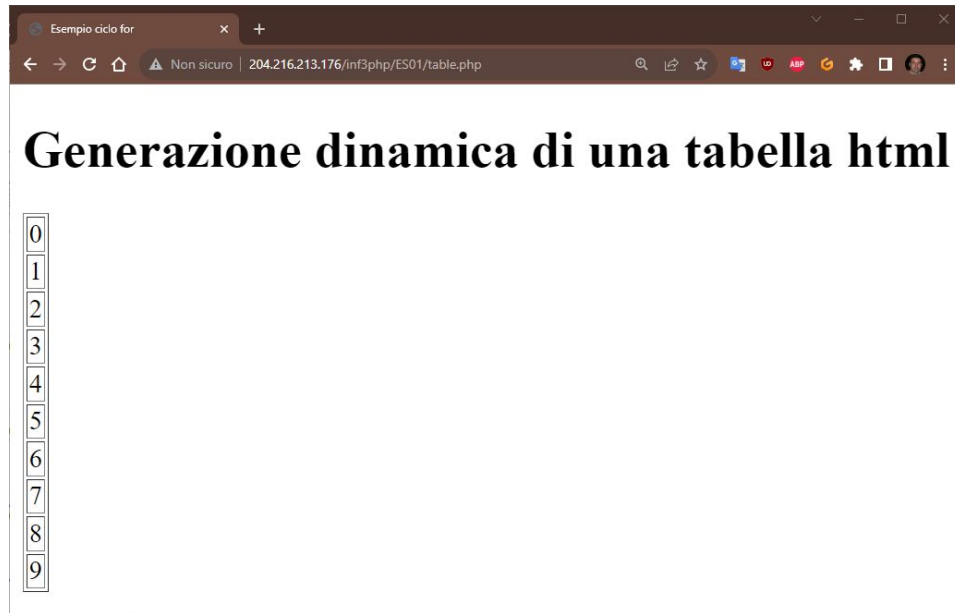


# Esempio

# Tabella html con php

INFORMATICA\_3\_PHP > www > ES01 > table.php

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Esempio ciclo for</title>
5 </head>
6 <body>
7   <h1>Generazione dinamica di una tabella html</h1>
8
9   <?php
10   echo '<table border="1">';
11   for($i=0; $i<10; $i++) {
12     echo "<tr><td>$i</td></tr>";
13   }
14   echo "</table>";
15   ?>
16
17 </body>
18 </html>
```



```
<body>
  <h1>Generazione dinamica di una tabella html</h1>

  <table border="1"><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>4</td></tr><tr><td>5</td></tr><tr><td>6</td></tr><tr><td>7</td></tr><tr><td>8</td></tr><tr><td>9</td></tr></table>
</body>
</html>
```

# Esercitazione

---





# Esercizio a

## Esercizio a

Scrivere una pagina PHP in grado di generare una tabella pitagorica

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100



# Esercizio b

## Esercizio b

Realizzare una pagina PHP che scriva: “Buongiorno/Buonasera/Buonanotte Paolo, benvenuta nella mia prima pagina PHP” la scelta tra Buongiorno/Buonasera/Buonanotte è legata all’ora attuale (8 – 12 Buongiorno, 12–20 Buonasera, 20–8 Buonanotte) Il nome “Paolo” deve essere contenuto in una variabile La pagina dev’essere chiusa da “stai usando il browser “ completata con il tipo di browser impiegato dall’utente.

```
<?php
$today = new DateTime("now", new DateTimeZone('Europe/Rome'));
echo $today->format('h:i:s');
$ora = $today->format('h');
echo "\nSono le $ora";
?>
```

 Output

```
08:36:26
Sono le 08
```




# Esercizio b

## Esercizio b

Realizzare una pagina PHP che scriva: “Buongiorno/Buonasera/Buonanotte Paolo, benvenuta nella mia prima pagina PHP” la scelta tra Buongiorno/Buonasera/Buonanotte è legata all’ora attuale (8 – 12 Buongiorno, 12–20 Buonasera, 20–8 Buonanotte) Il nome “Paolo” deve essere contenuto in una variabile La pagina dev’essere chiusa da “stai usando il browser “ completata con il tipo di browser impiegato dall’utente.

```
<?php
$today = new DateTime("now", new DateTimeZone('Europe/Rome'));
echo $today->format('h:i:s');
$ora = $today->format('h');
echo "\nSono le $ora";
?>
```

 Output

```
08:36:26
Sono le 08
```



# Esercizio c

Programma per stampare triangoli di asterischi

## Esercizio c - Stampare triangoli di asterischi

Scrivete un'applicazione PHP che visualizza i seguenti schemi di triangoli, uno sotto l'altro. Gli schemi devono essere prodotti utilizzando cicli `for`. Tutti gli asterischi `*` devono essere stampati usando una singola istruzione `echo ' * ' ;` che visualizzerà gli asterischi uno di fianco all'altro. Utilizzate `echo <br> ;` per andare alla riga successiva e `echo ' &nbsp; ; ' ;` per inserire uno spazio.

[illegible]

Usate il codice `<style>body {font-family: monospace;}</style>` per cambiare il tipo di carattere del testo nella pagina a un font monospaziato, questo serve per migliorare la leggibilità quando si devono mostrare contenuti come codice sorgente o dati tabulari. (vedi [risultato finale](#))



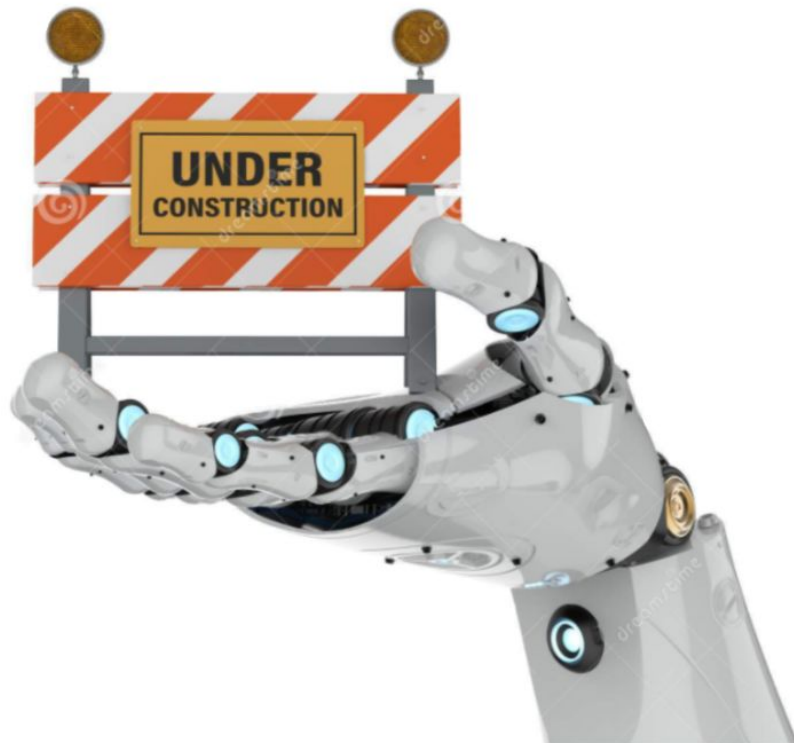
# Valutazione

---

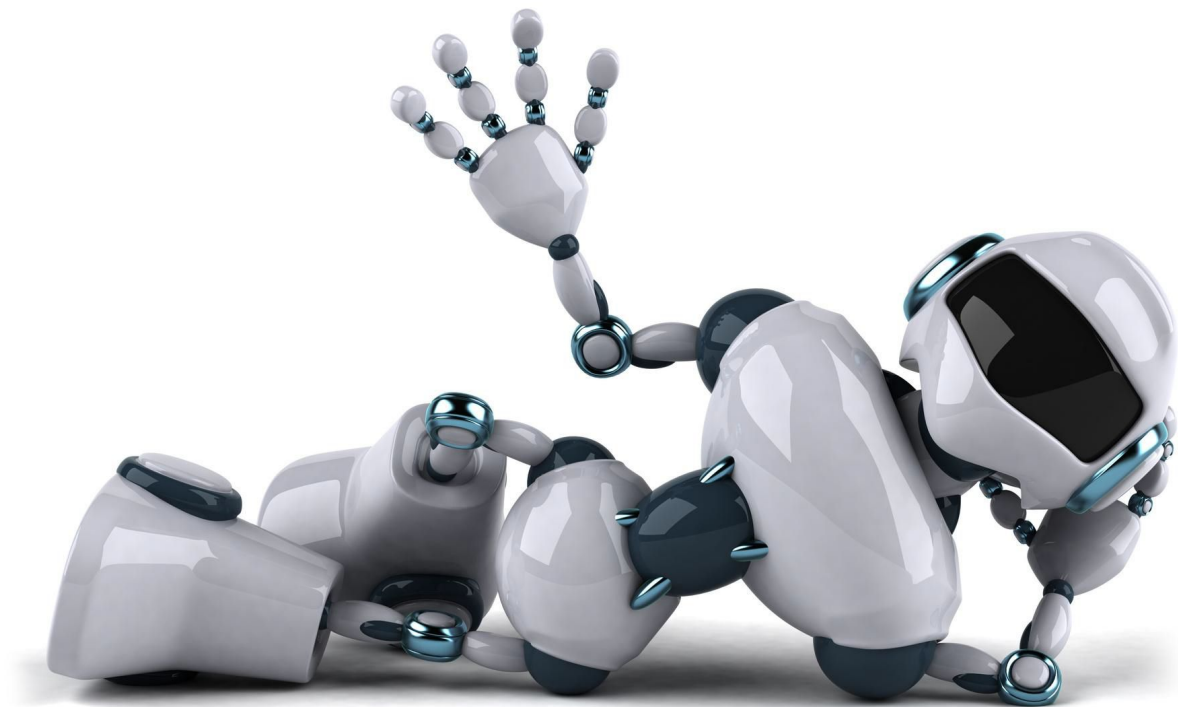
# Consegna del lavoro



Si ricorda l'importanza di svolgere e consegnare i progetti assegnati.



*Grazie per l'attenzione*



# Link/Riferimenti



- <https://fb-labs.blogspot.com/>
- <https://github.com/filippo-bilardo>
- <https://www.youtube.com/channel/UCoBNbHeKNdgeXjMiWhXuH8A/playlists>
- 
- <https://www.php.net/manual/en/>

# Revisioni



[v1.1 13/10/22](#) - versione iniziale

[v1.2 13/09/23](#) - aggiunti gli esempi

v1.3 08/09/24 -