

Introduzione al corso e al 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

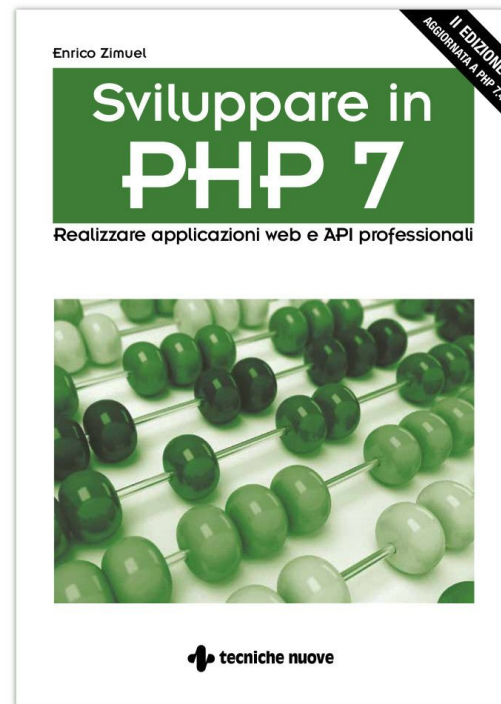
Enrico Zimuel

- Programmatore dal 1996
- Principal Software Engineer presso [Elastic](#)
- [Open source contributor](#)
- [PHP Certified Engineer](#) dal 2008
- Autore di [libri](#) tecnici (PHP, Javascript, Cloud, Open source, sicurezza informatica)
- Speaker [TEDx](#) e [relatore](#) internazionale
- Docente ITS ICT Piemonte dal 2018
- Docente a contratto Università di Torino
- Cofondatore [PUG Torino](#)
- Ulteriori informazioni: www.zimuel.it



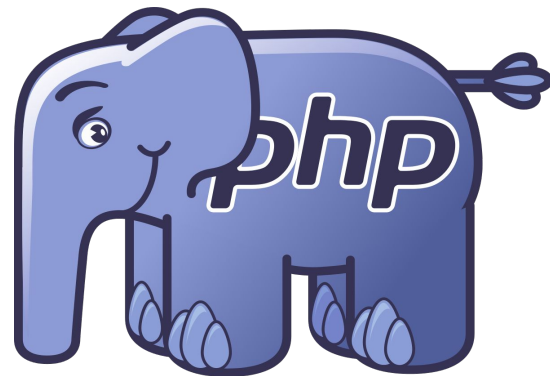
Libro su PHP

- **Sviluppare in PHP 7** (II edizione), realizzare applicazioni web e API professionali
- pp. 432, 2019, Tecniche Nuove editore, ISBN 978-8848140317
- Disponibile anche in formato e-book
- Informazioni: www.sviluppareinphp7.it



Programma del corso

- Introduzione al PHP
- I costrutti di base del linguaggio
- Programmazione a oggetti
- Sviluppo per il web (API, PSR-7)
- Utilizzo di librerie di terze parti con composer
- Accesso e gestione di file e database
- Unit testing con PHPUnit
- Utilizzo di un framework MVC (SimpleMVC)



Verifiche

- Le verifiche saranno degli esercizi/progetti da consegnare:
 - **Verifica intermedia** (metà corso), esercizio con consegna individuale
 - **Verifica finale** (fine corso), progetto individuale
- I voti della verifica intermedia e finale avranno un peso del **50%** sul voto finale

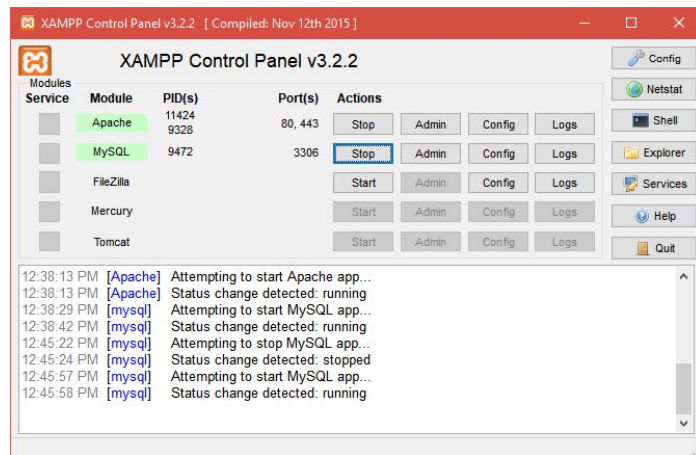
Materiale didattico

- Slide del corso
- Letture di approfondimento
- Esercizi
- Siti internet di riferimento

Installazione dell'ambiente PHP

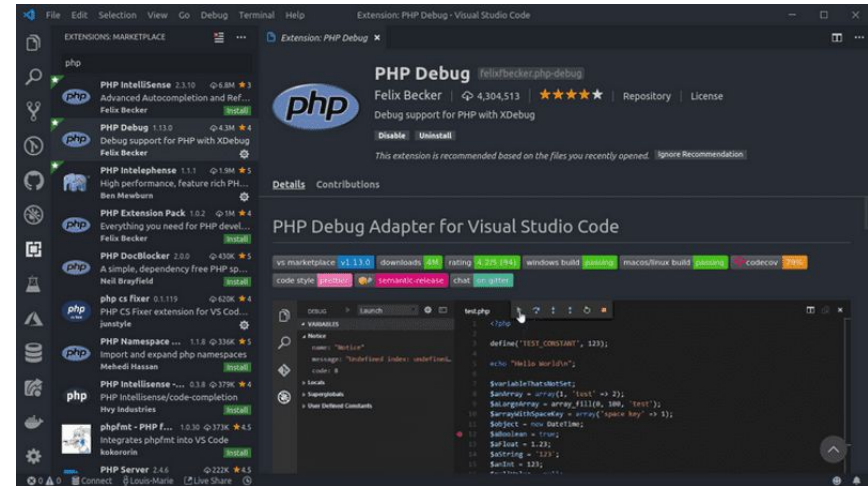
Installazione PHP + MySQL

- Per sistemi Windows/macOS:
 - [XAMPP](#)
- Per sistemi Linux:
 - [Install PHP + MySQL on Ubuntu](#)



Installazione IDE

- IDE = Integrated Development Environment
- Visual Studio Code
- PHP in Visual Studio Code



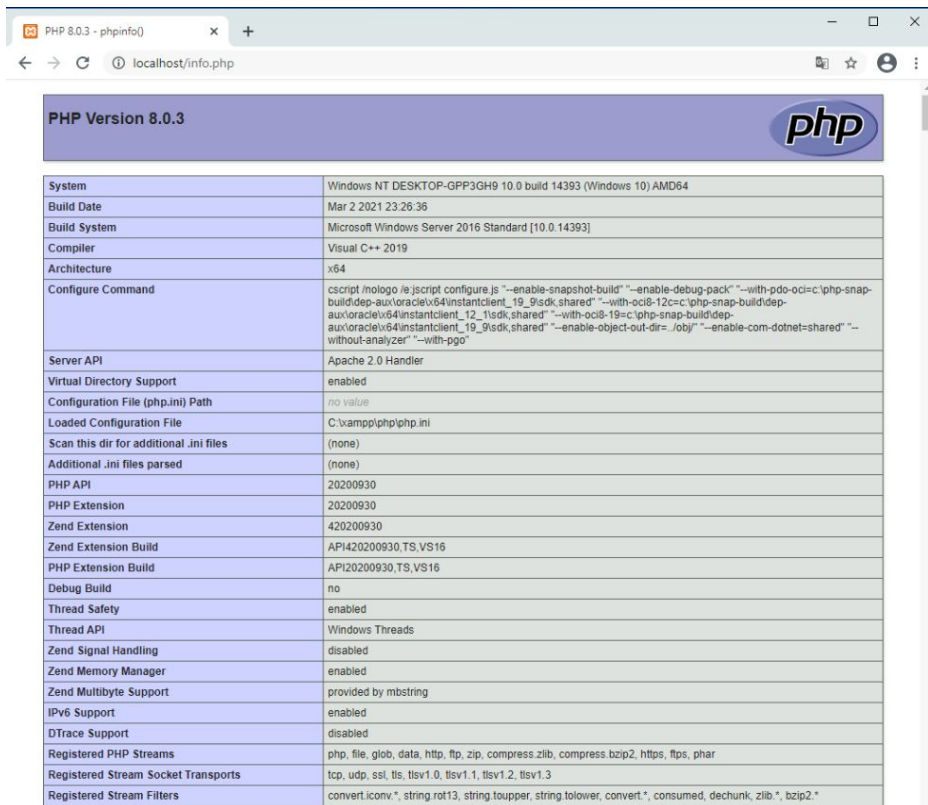
Esercizio

1. **Installare XAMPP su Windows/macOS** (o l'ambiente PHP su Linux)
2. **Installare Visual Studio Code**
3. Creare il file **info.php** con Visual Studio Code nella document root di Apache (C:\xampp\htdocs)

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

4. Aprire il browser all'indirizzo <http://localhost/info.php>

Soluzione



PHP Version 8.0.3	
System	Windows NT DESKTOP-GPP3GH9 10.0 build 14393 (Windows 10) AMD64
Build Date	Mar 2 2021 23:26:36
Build System	Microsoft Windows Server 2016 Standard [10.0.14393]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cmdscript mologo /e jsconfig configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build-dep-aux\oracle\v64\instantclient_19_9sdk,shared" "--with-oci8-12c=c:\php-snap-build-dep-aux\oracle\v64\instantclient_12_1sdk,shared" "--with-oci8-19=c:\php-snap-build-dep-aux\oracle\v64\instantclient_19_9sdk,shared" "--enable-object-out-dir=.obj" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930.TS.VS16
PHP Extension Build	API20200930.TS.VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*, bzip2.*



Il linguaggio PHP

- PHP: Hypertext Preprocessor
- Il linguaggio server-side più utilizzato al mondo: circa il **77%** di tutti i siti web*
- Siti che utilizzano il PHP: Facebook, Wikipedia, Yahoo, Etsy, Flickr, Digg, etc
- **28 anni** di utilizzo, dal 1995
- Completamente ad oggetti (OOP) dal PHP 5

* Fonte: w3techs.com

Storia del PHP

- Nato nel **1995** ad opera di [Rasmus Lerdorf](#)
- Nel **1997** due studenti dell'Università Technion di Haifa, [Andi Gutmans](#) e [Zeev Suraski](#) iniziano a lavorare sul PHP 3
- Nel **1998** viene rilasciato il PHP 3, inizia così il successo del linguaggio
- Nel **1999** Andi e Zeev fondano la società Zend Technologies
- Nel **2019** la società JetBrains sponsorizza [Nikita Popov](#) per lo sviluppo del PHP

Storia del PHP (2)

- Nel **2000** viene rilasciato il PHP 4, con un primo supporto alla OOP
- Nel **2005** viene rilasciato il PHP 5, pieno supporto OOP
- Il PHP 6 non ha mai visto la luce
- Nel **2015** viene rilasciato il PHP 7, rivoluzione del linguaggio!
- Nel **2020** viene rilasciato il PHP 8 con numerose novità: compilatore JIT, named arguments, attributes, etc
- Nel **2021** nasce la [fondazione PHP](#)

PHP 7

- Riscrittura dell'engine del linguaggio
- Performance notevoli rispetto al PHP 5 (2x più veloce)
- Introduzione dei tipi scalari e type hint strict
- Nuova gestione delle eccezioni (Throwable)

PHP 5 vs PHP 7

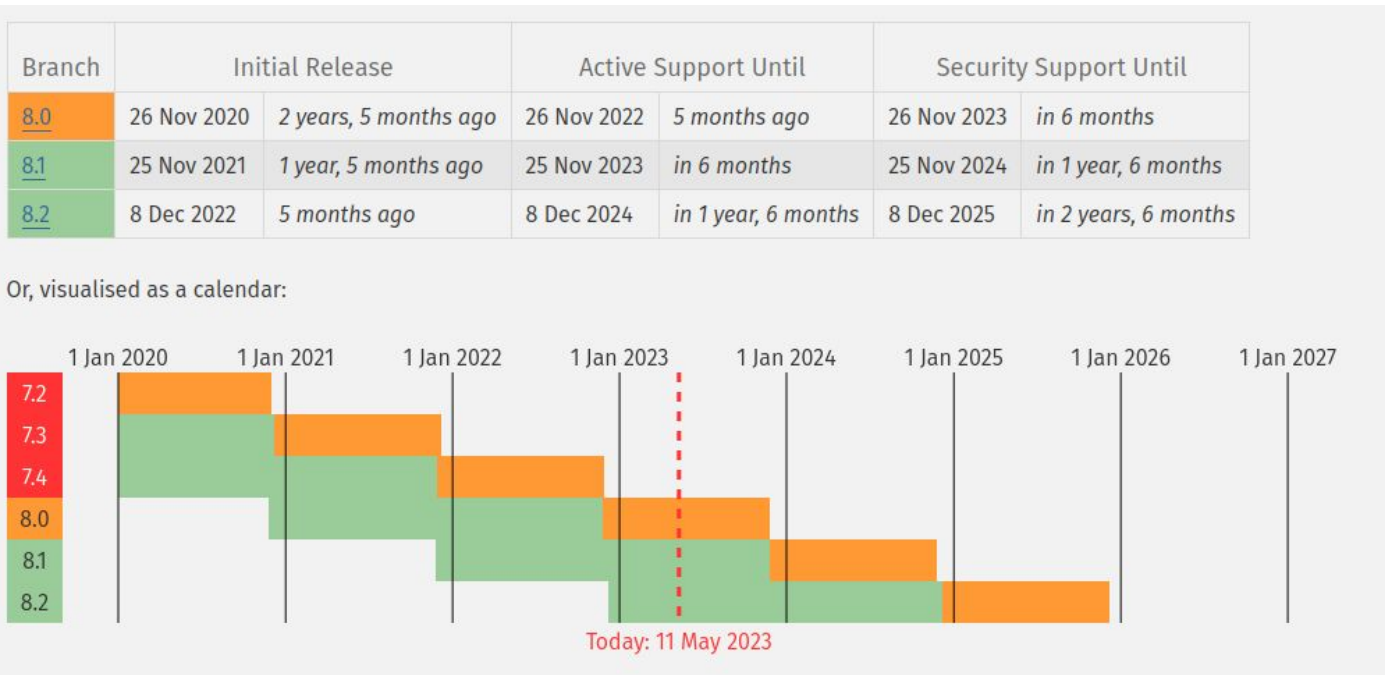
```
$a = [];  
for ($i = 0; $i < 1000000; $i++) {  
    $a[$i] = "hello";  
}  
echo memory_get_usage(true);
```

	PHP 5.6	PHP 7
Memoria	428 MB	33 MB
Tempi di esecuzione	0.49 sec	0.06 sec

PHP 8

- Rilasciato il 26 Novembre 2020
- Introduzione di un compilatore **Just In Time** (JIT)
- Named arguments
- Attributi
- Constructor property promotion
- Union types
- Performance migliorate rispetto al PHP 7

Versioni supportate



Fonte: php.net

Ecosistema PHP

- CMS: [Drupal](#) , [WordPress](#) , [Joomla](#) , etc
- E-commerce: [Magento](#), [OpenCart](#), [PrestaShop](#), etc
- CRM: [CiviCRM](#), [SugarCRM](#), [Vtiger](#), etc
- Wiki: [DokuWiki](#) , [MediaWiki](#) , [PmWiki](#) , etc
- LMS: [Moodle](#), [Forma.lms](#), etc
- Framework: [Laravel](#), [Symfony](#), [Zend Framework](#), etc

Esecuzione di un codice in PHP

1. Il codice PHP viene tradotto in **bytecode** (codice oggetto)
2. Il bytecode viene interpretato (eseguito) dallo **Zend Engine** (la virtual machine del PHP)

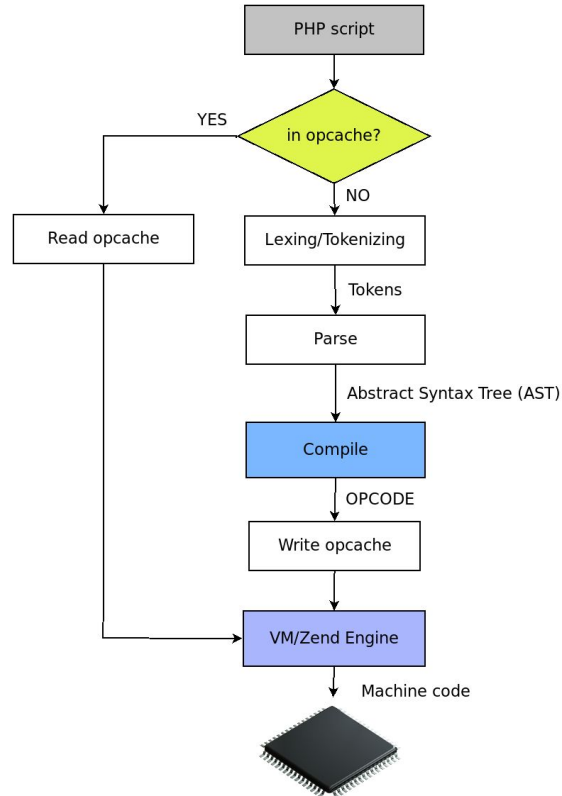
Esempio di bytecode

```
for ($i=0; $i<100; $i++) {  
    echo $i;  
}
```

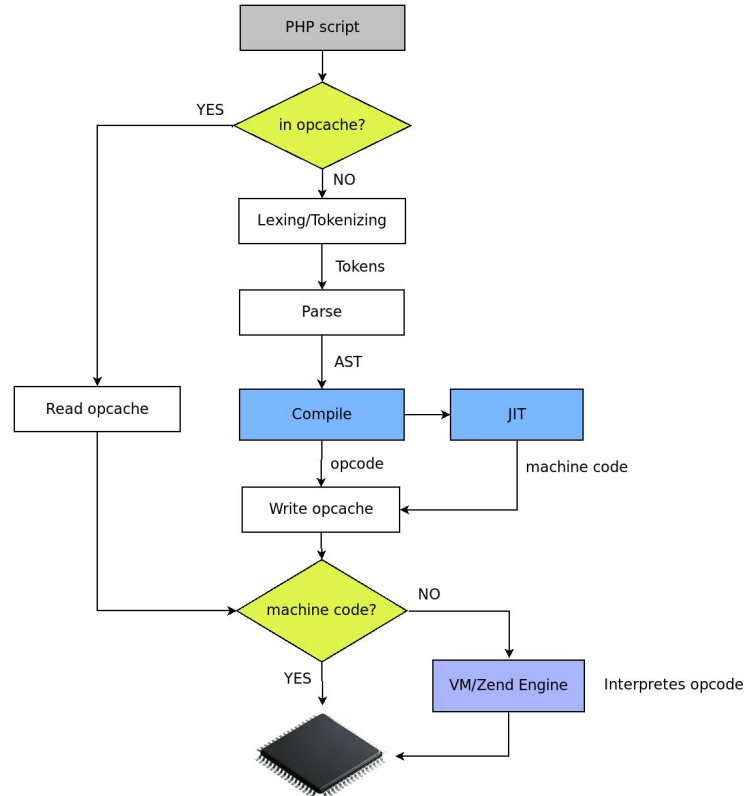
Opcode:

```
L0 (2):    ASSIGN CV0($i) int(0)  
L1 (2):    JMP L4  
L2 (3):    ECHO CV0($i)  
L3 (2):    PRE_INC CV0($i)  
L4 (2):    T1 = IS_SMALLER CV0($i) int(100)  
L5 (2):    JMPNZ T1 L2  
L6 (5):    RETURN int(1)
```

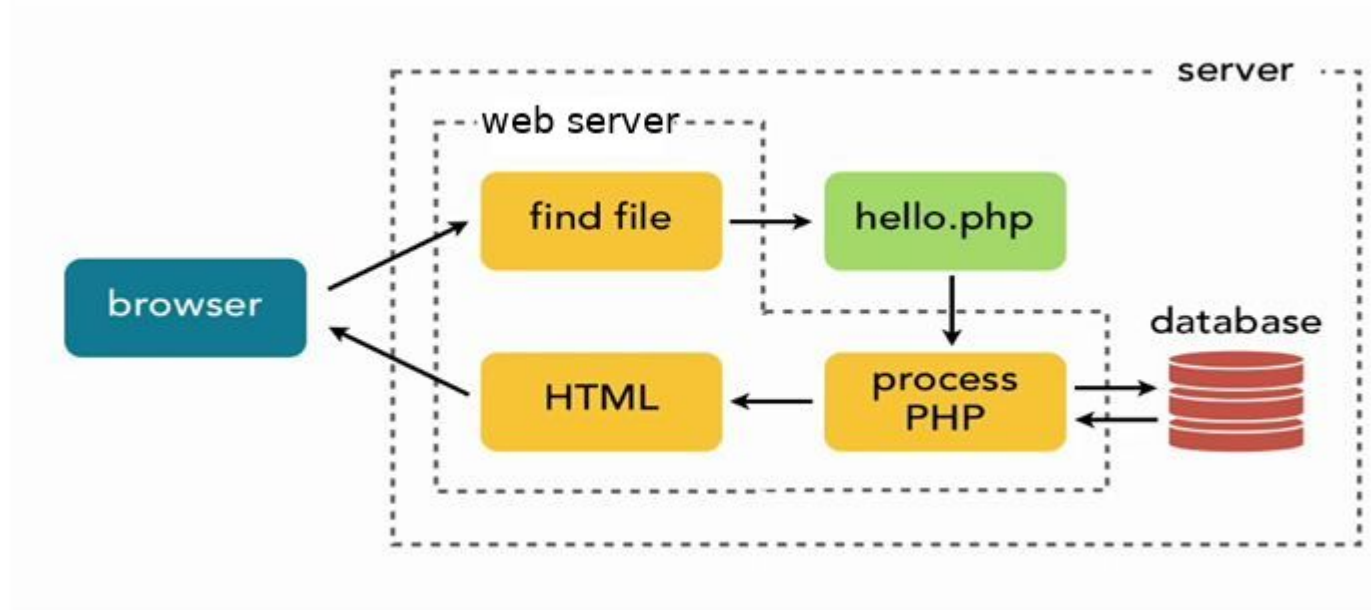
Ciclo di vita del PHP



Il JIT del PHP 8



Architettura client/server



LAMP/LEMP stack

- Linux **A**pache **M**ySQL **P**HP (**LAMP**)
- Linux **N**ginx (pronuncia Engine-x) **M**ySQL **P**HP (**LEMP**)



Esercizio

- Leggere l'articolo ["The business of PHP"](#) reperibile nella sezione **Lecture di approfondimento** su its.mrooms.net
- Leggere la storia dei 25 anni del linguaggio: [PHP turns 25](#)
- Cercare annunci di lavoro su PHP in [LinkedIn](#)

Grazie dell'attenzione!

Per informazioni:

enrico.zimuel@its-ictpiemonte.it