

Dynamic content: programming the web servers



Step 1: let's install a (Apache) web server



Let's install a Web Server

We will, step by step:

1. install Apache (with a series of extra tools: DB and languages – Perl and PHP)
2. Customize it (e.g. by changing port)
3. Customize its response (static files)
4. Attack the dynamic content problem



XAMPP

<https://www.apachefriends.org/download.html>

What does XAMPP means? Generalization of WAMP, LAMP, MAMP

W = Windows, L=Linux, M=Mac, X=Anything

A = Apache Web Server

M = MySQL , MariaDB

P = PHP

P = Perl



XAMPP download

<https://www.apachefriends.org/download.html>

Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for Windows 7.2.27, 7.3.14 & 7.4.2

Version	Checksum	Size
7.2.27 / PHP 7.2.27	What's Included? md5 sha1 Download (64 bit)	147 Mb
7.3.14 / PHP 7.3.14	What's Included? md5 sha1 Download (64 bit)	147 Mb
7.4.2 / PHP 7.4.2	What's Included? md5 sha1 Download (64 bit)	148 Mb

[Requirements](#) [Add-ons](#) [More Downloads](#) »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the [Forums](#) or [Stack Overflow](#).

- [Linux FAQs](#)
- [Windows FAQs](#)
- [OS X FAQs](#)
- [OS X XAMPP-VM FAQs](#)

Add-ons and Themes



Bitnami provides a free all-in-one tool to install Drupal, Joomla!, WordPress and many other popular open source

Follow the documentation



XAMPP download



XAMPP for **Linux** 7.2.27, 7.3.14 & 7.4.2

Version		Checksum			Size
7.2.27 / PHP 7.2.27	What's Included?	md5	sha1	Download (64 bit)	148 Mb
7.3.14 / PHP 7.3.14	What's Included?	md5	sha1	Download (64 bit)	149 Mb
7.4.2 / PHP 7.4.2	What's Included?	md5	sha1	Download (64 bit)	151 Mb

[Requirements](#) [Add-ons](#) [More Downloads »](#)



XAMPP for **OS X** 7.2.27, 7.3.14, 7.4.2, 7.2.27, 7.3.14 & 7.4.2

Version		Checksum			Size
7.2.27 / PHP 7.2.27	What's Included?	md5	sha1	Download (64 bit)	159 Mb
7.3.14 / PHP 7.3.14	What's Included?	md5	sha1	Download (64 bit)	159 Mb
7.4.2 / PHP 7.4.2	What's Included?	md5	sha1	Download (64 bit)	160 Mb
7.2.27 / PHP 7.2.27	What's Included?	md5	sha1	Download (64 bit)	322 Mb

VM Installation (e.g. on Mac)...

LAMP is installed on a virtual machine.

We need to:

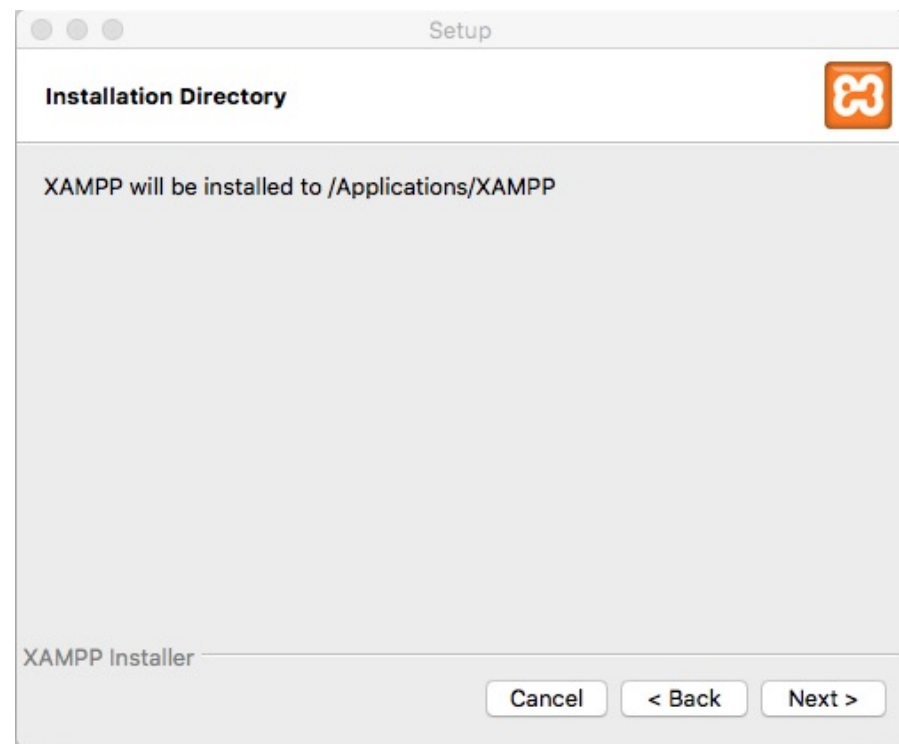
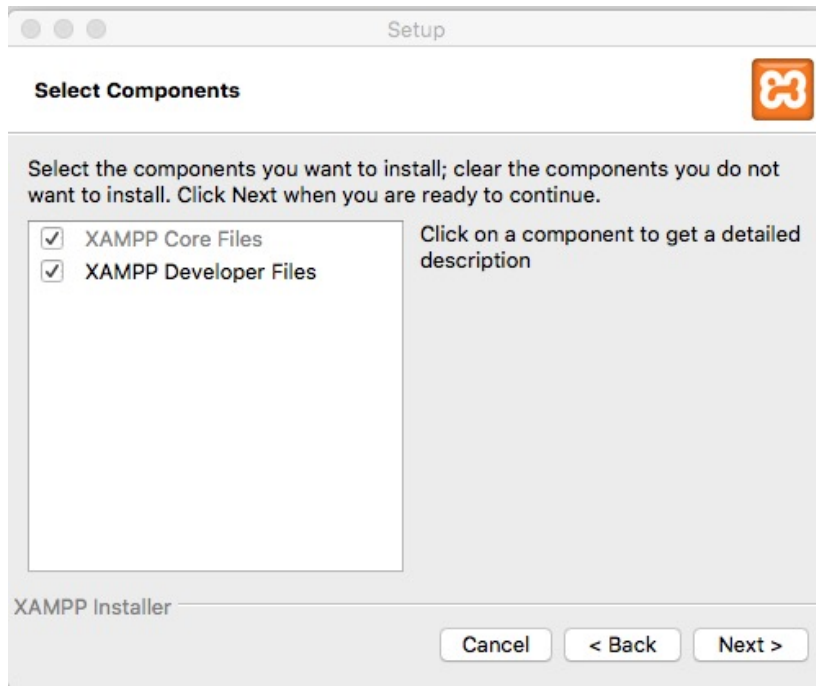
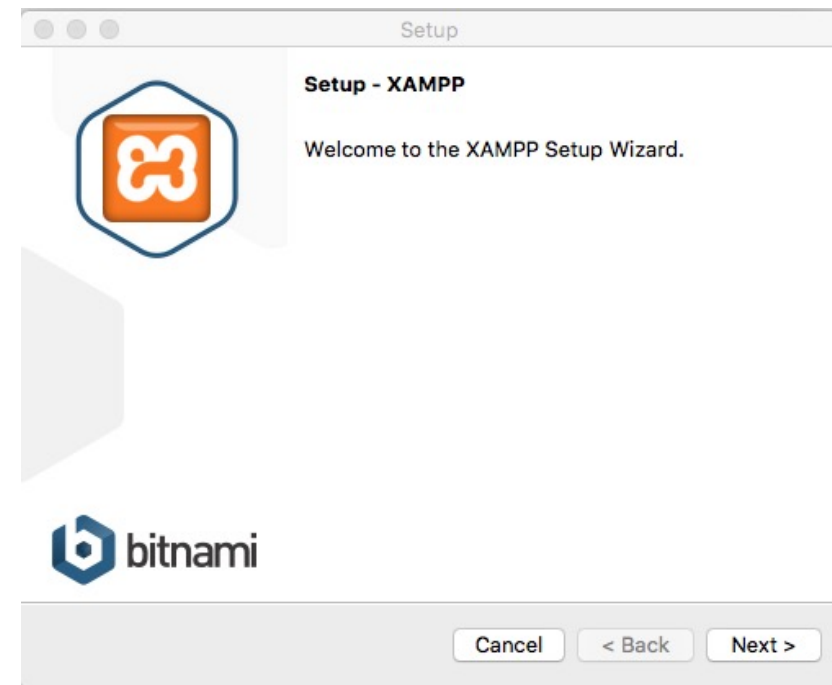
- Connect the port on our machine (e.g. 8080) to the 80 port on the VM
- Access the VM file system from our file system

Mac users: do not use the VM version, but rather use this:

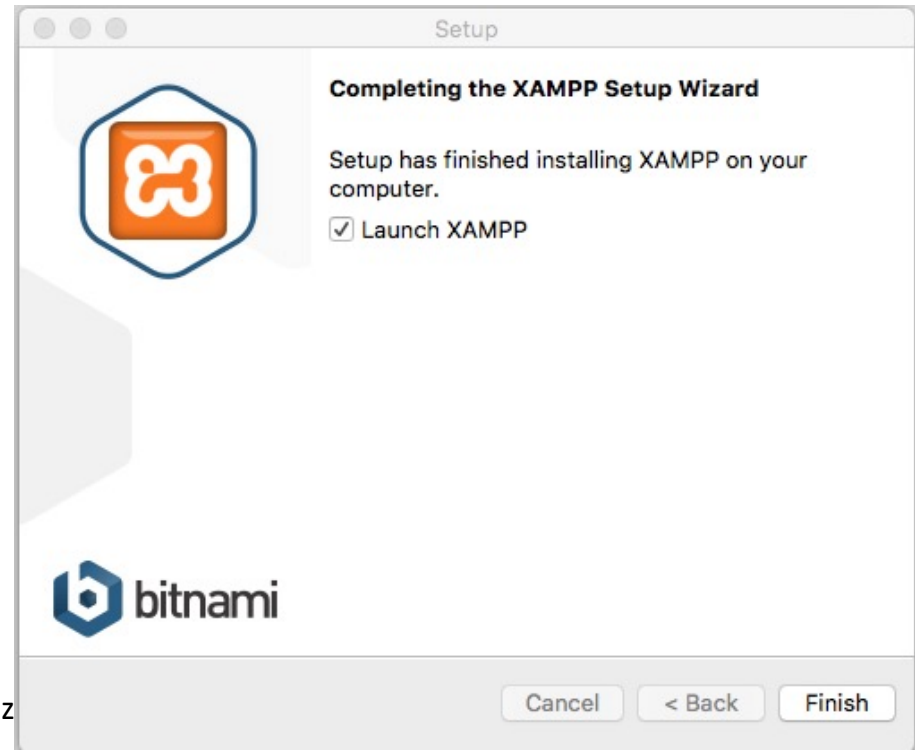
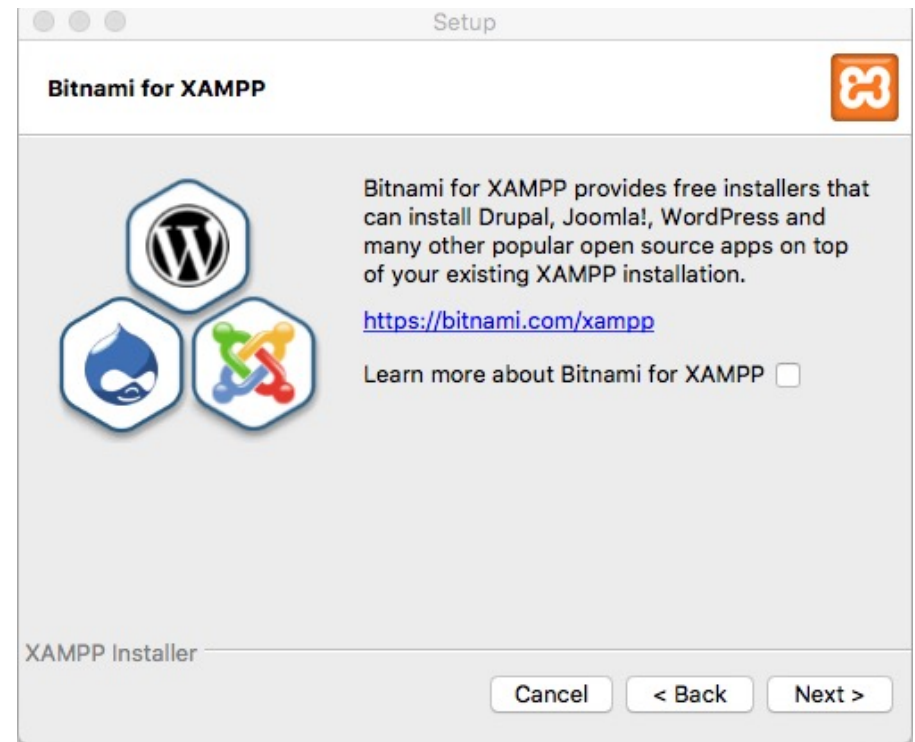
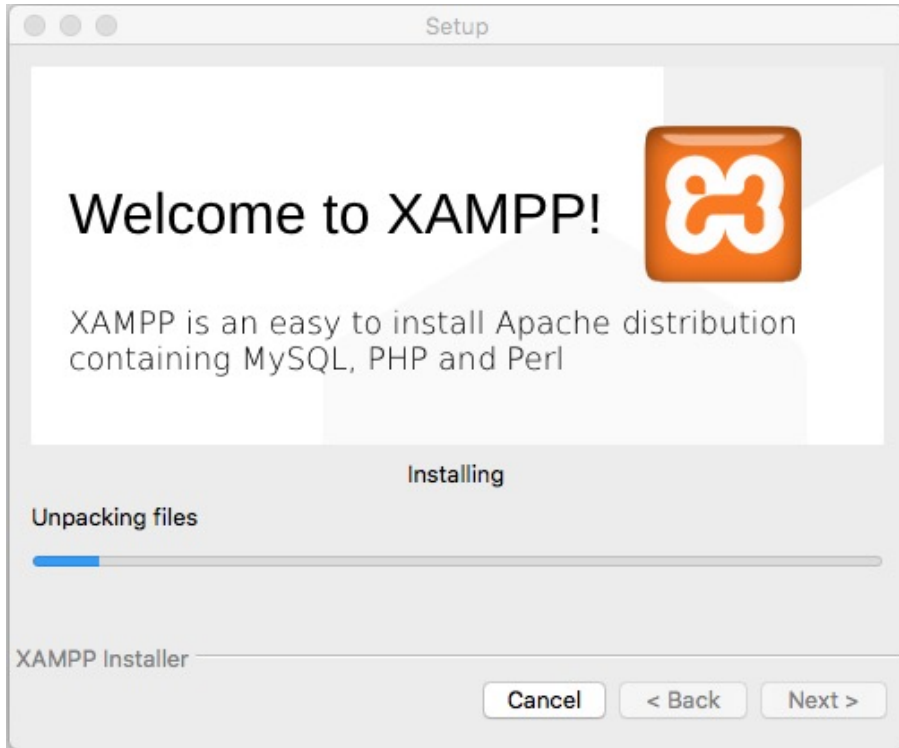
<https://sourceforge.net/projects/xampp/files/XAMPP%20Mac%20OS%20X/7.4.2/>



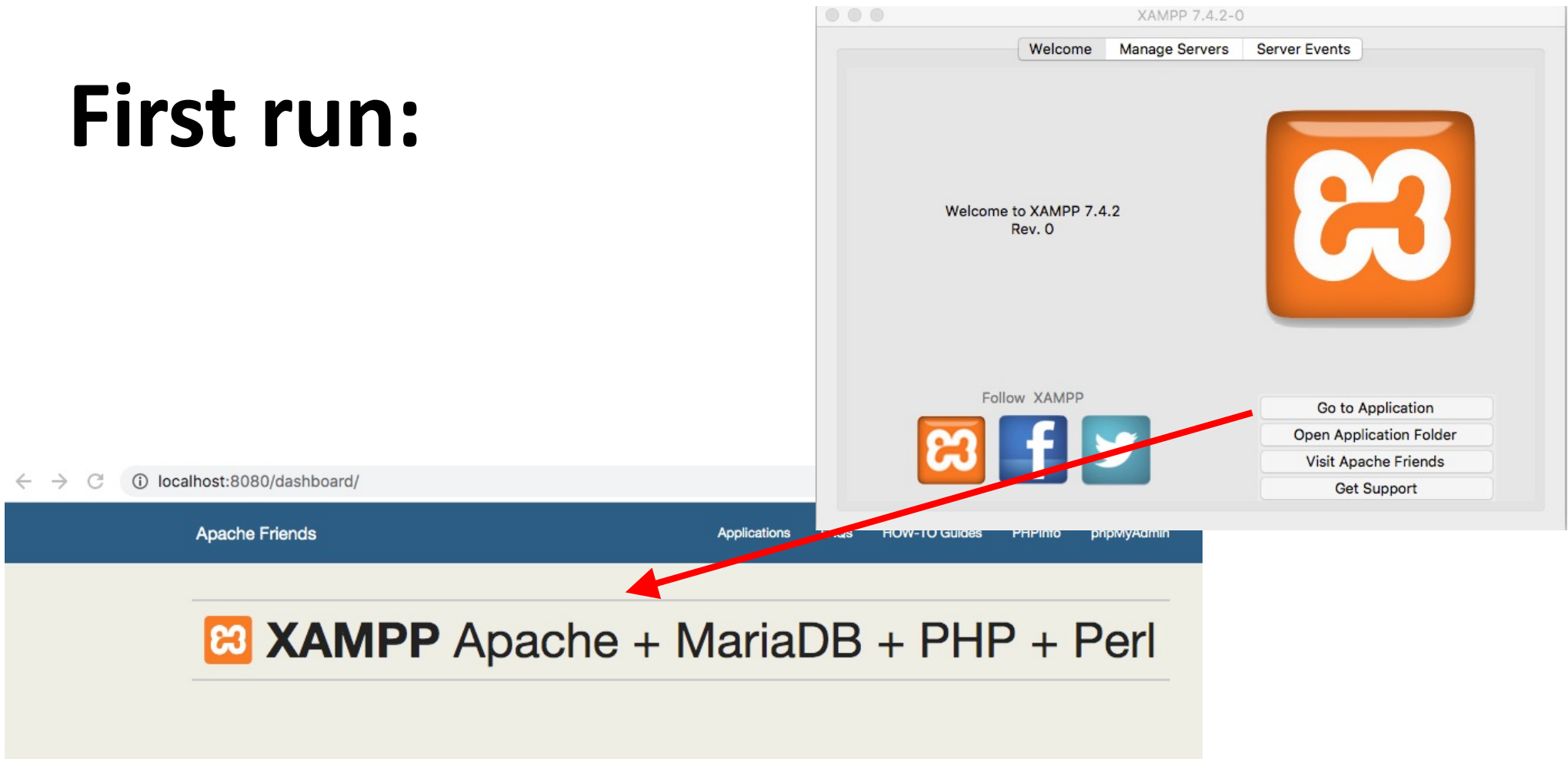
First run: setup - 1



First run: setup - 2



First run:



Welcome to XAMPP for 7.4.2-0

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the [FAQs](#) to learn how to protect your site. Alternatively you can use [WAMP](#), [MAMP](#) or [LAMP](#) which are similar packages which are more suitable for production.

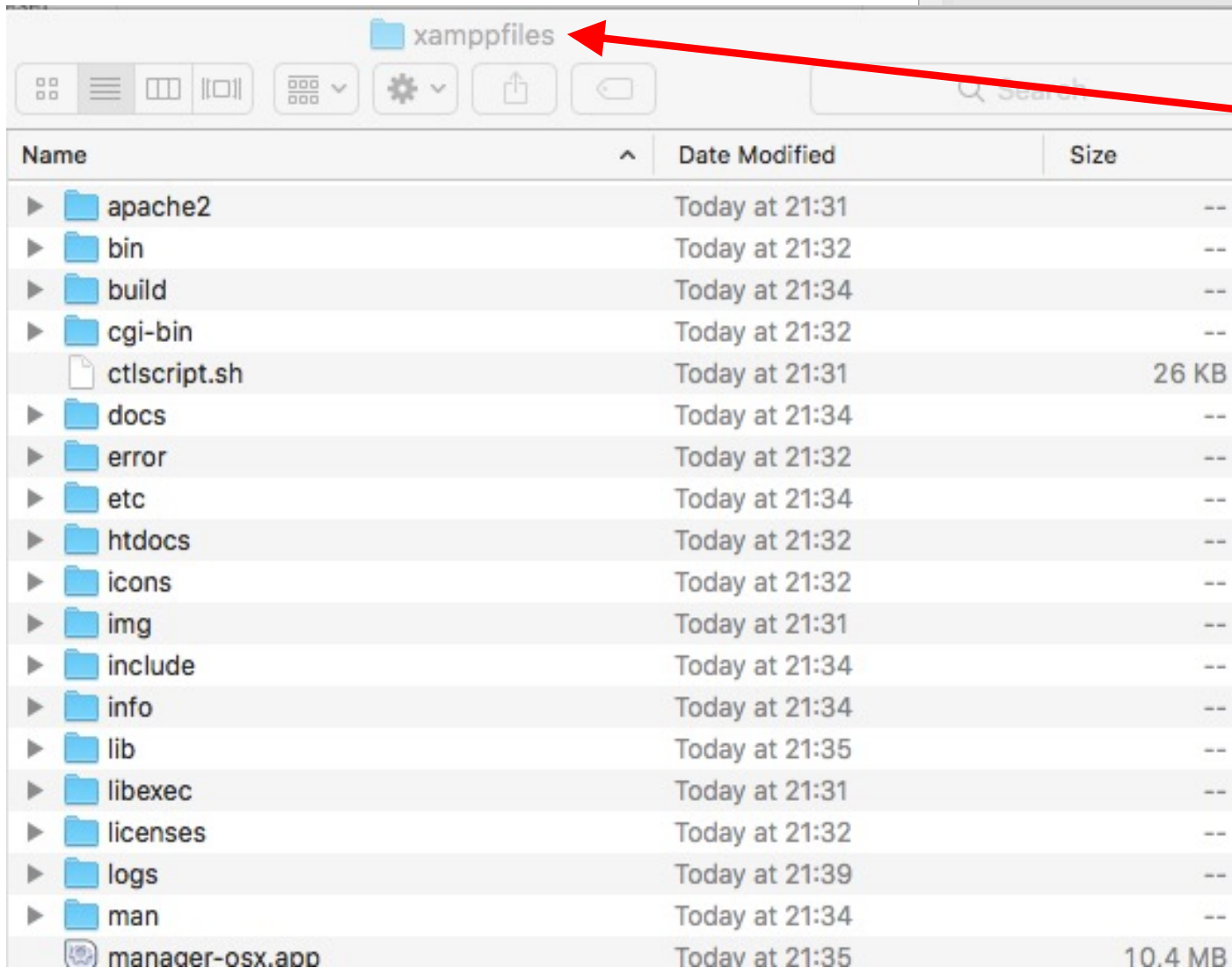
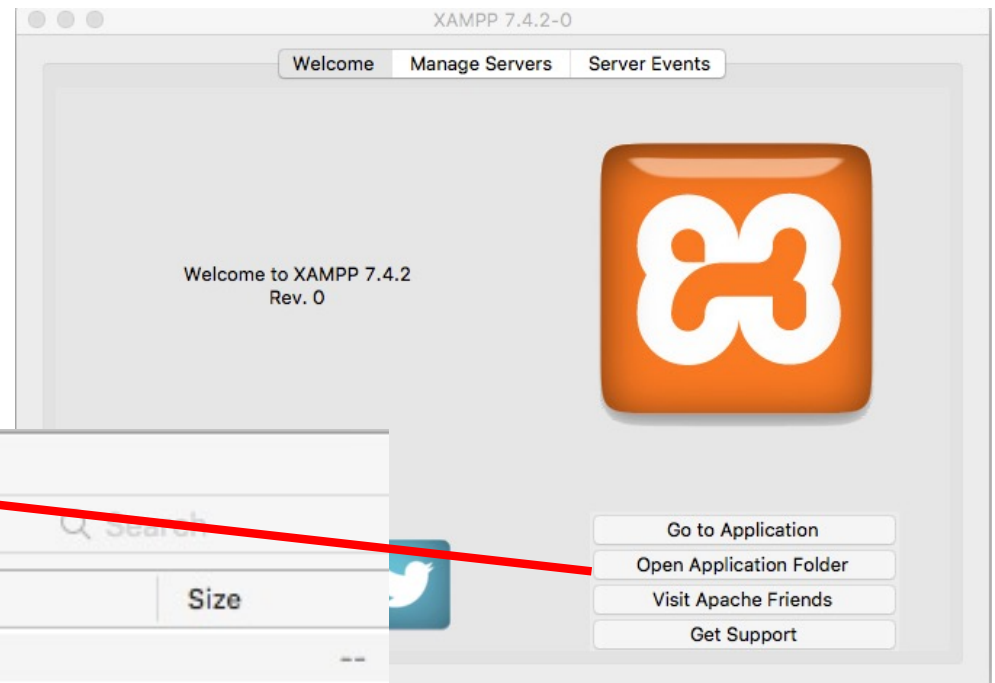
Start the XAMPP Control Panel to check the server status.

Community

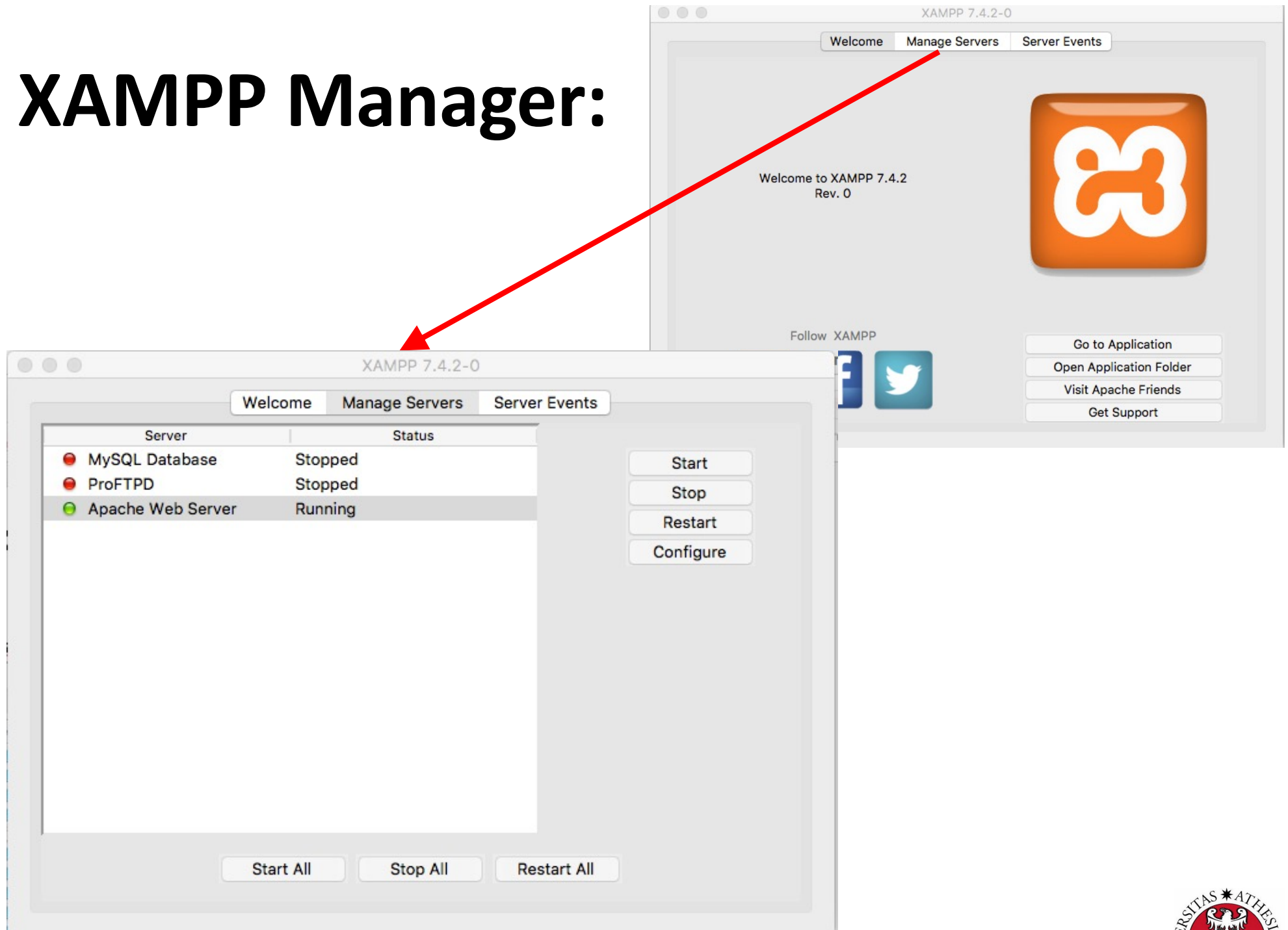
XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our [Forums](#).

XAMPP Manager:

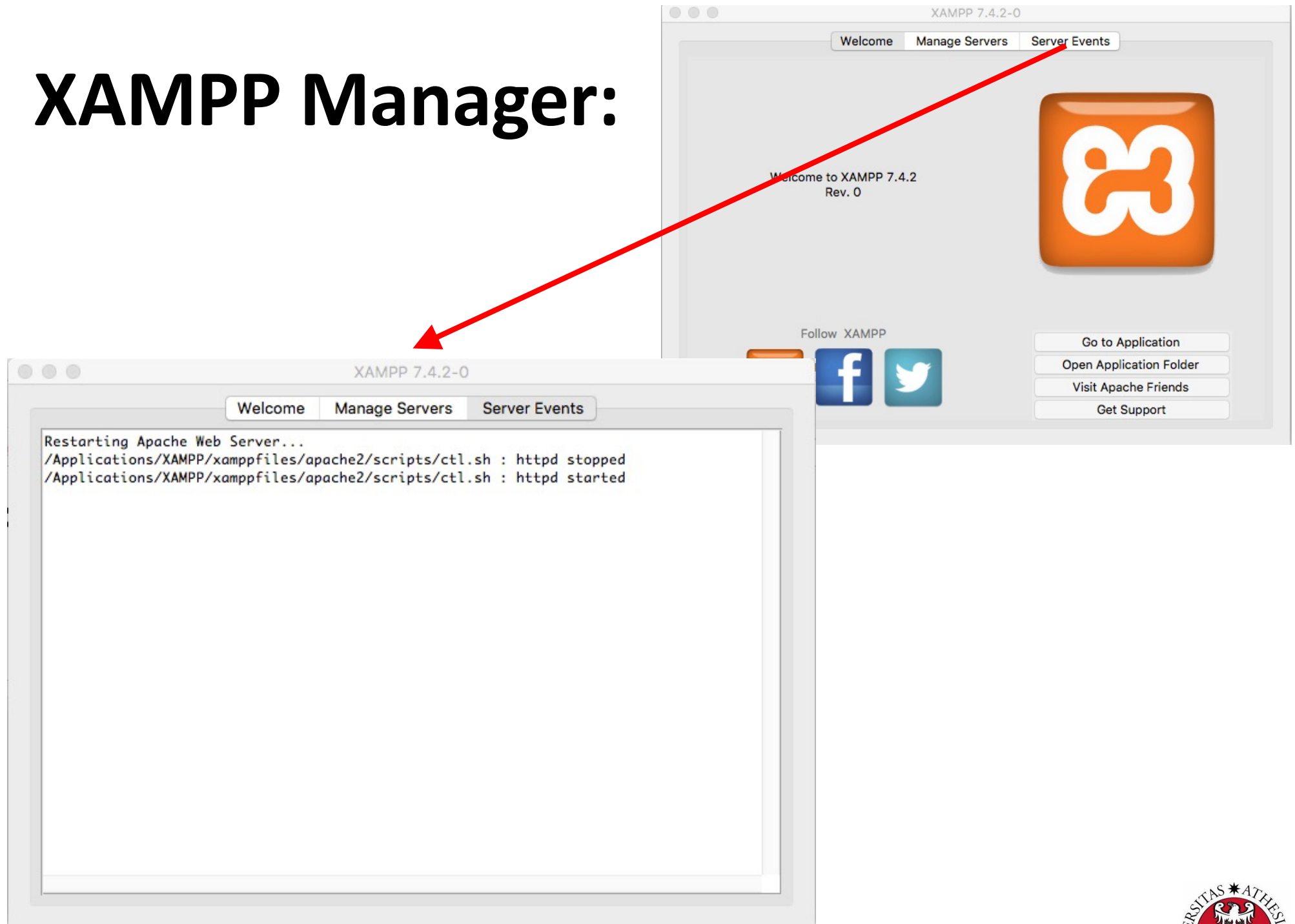
/MacintoshHD/Applications/XAMPP/xamppfiles



XAMPP Manager:



XAMPP Manager:



Main files e directories

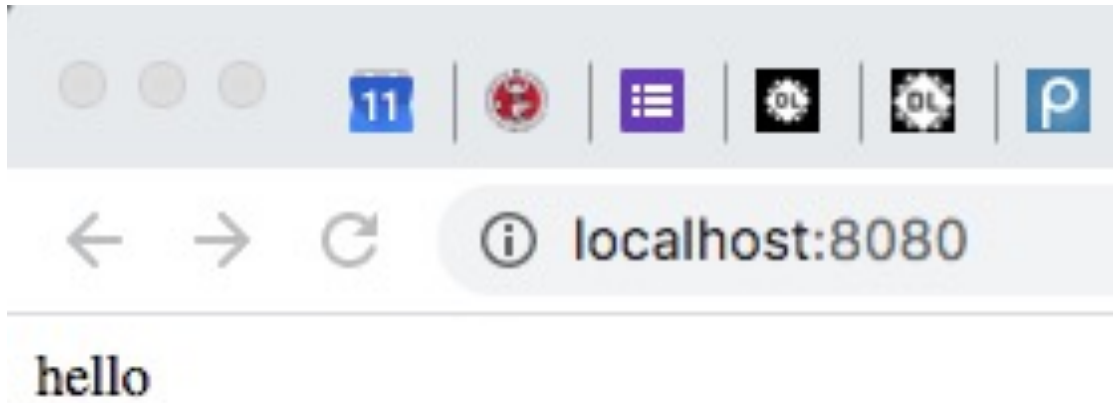
The screenshot shows the XAMPP directory structure. The following table summarizes the items and their functions as indicated by the red arrows:

Item	Function
bin	
cgi-bin	Executables for dynamic pages
etc	
htdocs	HTTP Server filesystem
logs	
manager-osx.app	XAMPP Manager
uninstall.app	
xamppfiles	

Modifying server content

```
MR-MBP-14955:local ronchet$ cd /Applications/XAMPP/xamppfiles/htdocs
MR-MBP-14955:htdocs ronchet$ touch index.html
MR-MBP-14955:htdocs ronchet$ vi index.html
MR-MBP-14955:htdocs ronchet$ cat index.html
hello
```

Create empty file
Edit file
Show content



APACHE DEFAULTS ARE: index.html, index.php

Apache configuration

See

http://www.cellbiol.com/bioinformatics_web_development/chapter-2-the-linux-operating-system-setting-up-a-linux-web-server/apache-web-server-configuration/

(has to be adapted to your locations)



Step 2:

let's use our web server to generate dynamic pages



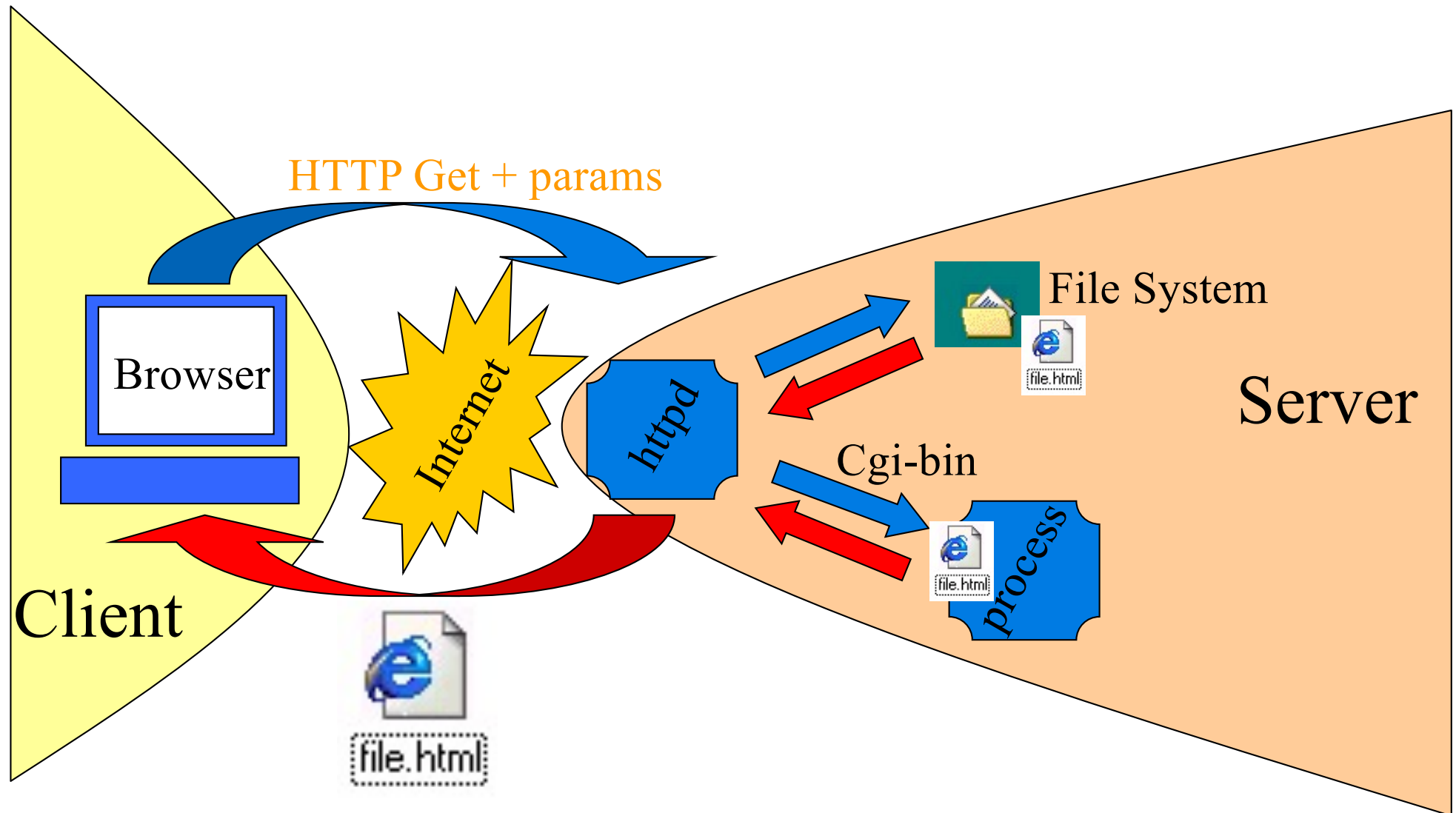
Dynamic pages: the main idea:

We want to obtain NON STATIC information from the server. This implies executing some code on it, and send the results to the user.

e.g.: what's the time?



The original web architecture: dynamic pages



Evolution 1: dynamically create (interlinked) documents

The first implementation: CGI

The Common Gateway Interface was (is) a way to tell the server to spawn a process, get its results and send them as HTTP response.

Reading:

- <https://computer.howstuffworks.com/cgi.htm>
- https://en.wikipedia.org/wiki/Common_Gateway_Interface

Follower: FastCGI

- <https://en.wikipedia.org/wiki/FastCGI>



Creating dynamic pages

```
MR-MBP-14955:local ronchet$ cd /Applications/XAMPP/xamppfiles/cgi-bin
MR-MBP-14955:htdocs ronchet$ touch getTime.sh
MR-MBP-14955:htdocs ronchet$ vi getTime.sh
MR-MBP-14955:htdocs ronchet$ cat getTime.sh
#!/bin/sh
echo `date`
MR-MBP-14955:cgi-bin ronchet$ ls -la getTime.sh
-rw-r--r-- 1 ronchet admin 22 Feb 11 22:26 getTime.sh
MR-MBP-14955:cgi-bin ronchet$ chmod 755 getTime.sh
MR-MBP-14955:cgi-bin ronchet$ ls -la getTime.sh
-rwxr-xr-x 1 ronchet admin 22 Feb 11 22:26 getTime.sh
MR-MBP-14955:cgi-bin ronchet$ ./getTime.sh
Tue Feb 11 22:28:56 CET 2020
```

Create empty file

Edit file

Show content

Show permissions

Make file executable

Show permissions again

Execute file

← → ↻ ⓘ localhost:8080/cgi-bin/getTime.sh

Server error!

The server encountered an internal error and was unable to complete your request.

Error message:

Premature end of script headers: getTime.sh

If you think this is a server error, please contact the [webmaster](#).

Error 500

localhost

Apache/2.4.41 (Unix) OpenSSL/1.1.1d PHP/7.4.2 mod_perl/2.0.8-dev Perl/v5.16.3

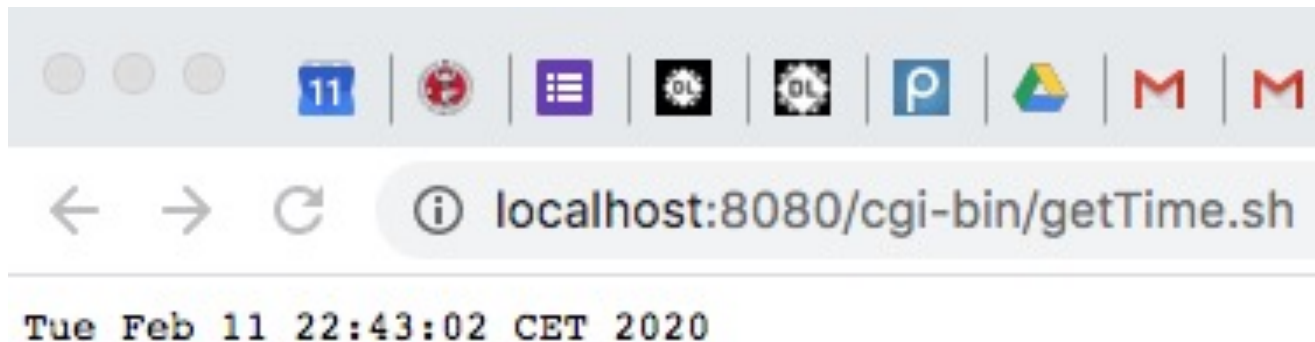


Creating dynamic pages

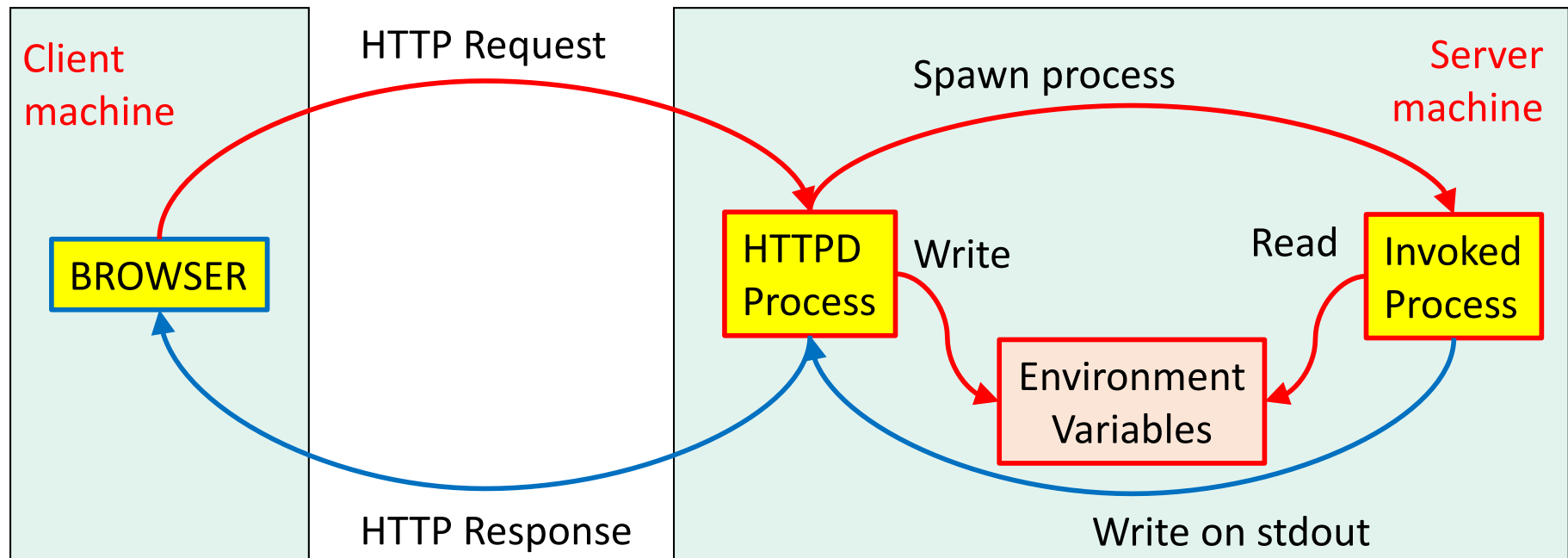
```
MR-MBP-14955:local ronchet$ cd /Applications/XAMPP/xamppfiles/cgi-bin
MR-MBP-14955:htdocs ronchet$ vi getTime.sh
MR-MBP-14955:htdocs ronchet$ cat getTime.sh
#!/bin/sh
echo "Content-type: text/plain; charset=iso-8859-1"
echo
echo `date`
```

Edit file

Show content



Getting info about the request



Access to environment vars is supported in all programming languages!

- **Server specific variables:**

- `SERVER_SOFTWARE` : *name/version* of **HTTP** server.
- `SERVER_NAME` : **host name** of the server, may be **dot-decimal** IP address.
- `GATEWAY_INTERFACE` : *CGI/version*.

Getting info about the request

- Request specific variables:

- `SERVER_PROTOCOL` : HTTP/*version*.
- `SERVER_PORT` : TCP port (decimal).
- `REQUEST_METHOD` : name of HTTP method (see above).
- `PATH_INFO` : path suffix, if appended to URL after program name and a slash.
- `PATH_TRANSLATED` : corresponding full path as supposed by server, if `PATH_INFO` is present.
- `SCRIPT_NAME` : relative path to the program, like `/cgi-bin/script.cgi`.
- `QUERY_STRING` : the part of URL after `?` character. The query string may be composed of **name=value* pairs separated with ampersands (such as `var1=val1&var2=val2...`) when used to submit form data transferred via GET method as defined by HTML `application/x-www-form-urlencoded`.
- `REMOTE_HOST` : host name of the client, unset if server did not perform such lookup.
- `REMOTE_ADDR` : IP address of the client (dot-decimal).
- `AUTH_TYPE` : identification type, if applicable.
- `REMOTE_USER` used for certain `AUTH_TYPE` s.
- `REMOTE_IDENT` : see `ident`, only if server performed such lookup.
- `CONTENT_TYPE` : Internet media type of input data if PUT or POST method are used, as provided via HTTP header.
- `CONTENT_LENGTH` : similarly, size of input data (decimal, in octets) if provided via HTTP header.
- Variables passed by user agent (`HTTP_ACCEPT` , `HTTP_ACCEPT_LANGUAGE` , `HTTP_USER_AGENT` , `HTTP_COOKIE` and possibly others) contain values of corresponding HTTP headers and therefore have the same sense.

Testing it...

You need to modify test-cgi as described in the file itself



```
CGI/1.0 test script report:

argc is 0. argv is .

SERVER_SOFTWARE = Apache/2.4.41 (Unix) OpenSSL/1.1.1d PHP/7.4.2 mod_perl/2.0.8-dev Perl/v5.16.3
SERVER_NAME = localhost
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.1
SERVER_PORT = 8080
REQUEST_METHOD = GET
HTTP_ACCEPT = text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/test-cgi
QUERY_STRING =
REMOTE_HOST =
REMOTE_ADDR = ::1
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =
```

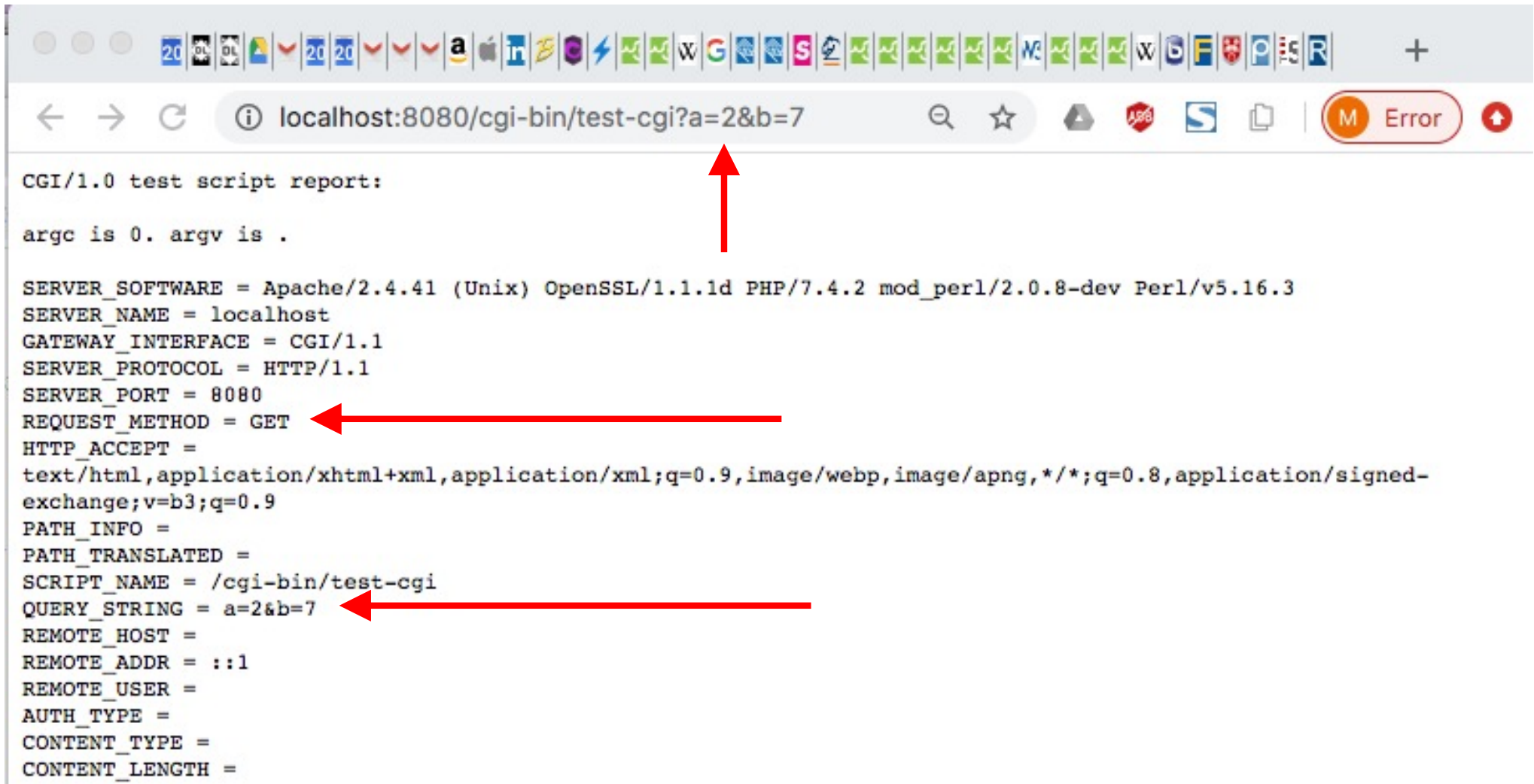


Step 3: let's pass parameters in our request

HTML Forms



Parameters passing



```
CGI/1.0 test script report:

argc is 0. argv is .

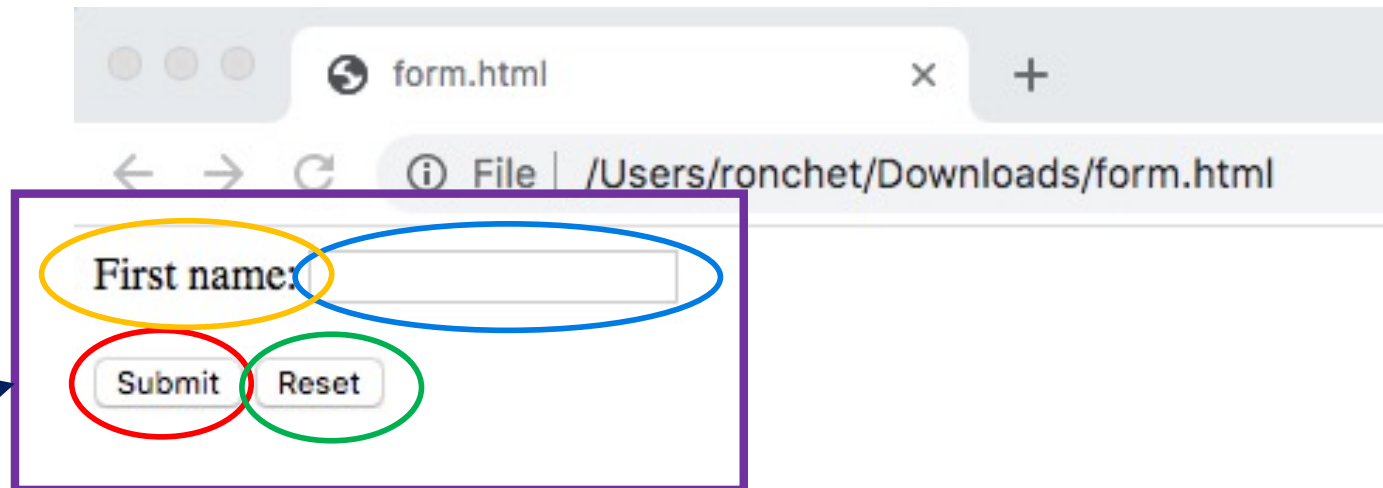
SERVER_SOFTWARE = Apache/2.4.41 (Unix) OpenSSL/1.1.1d PHP/7.4.2 mod_perl/2.0.8-dev Perl/v5.16.3
SERVER_NAME = localhost
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.1
SERVER_PORT = 8080
REQUEST_METHOD = GET
HTTP_ACCEPT =
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/test-cgi
QUERY_STRING = a=2&b=7
REMOTE_HOST =
REMOTE_ADDR = ::1
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =
```



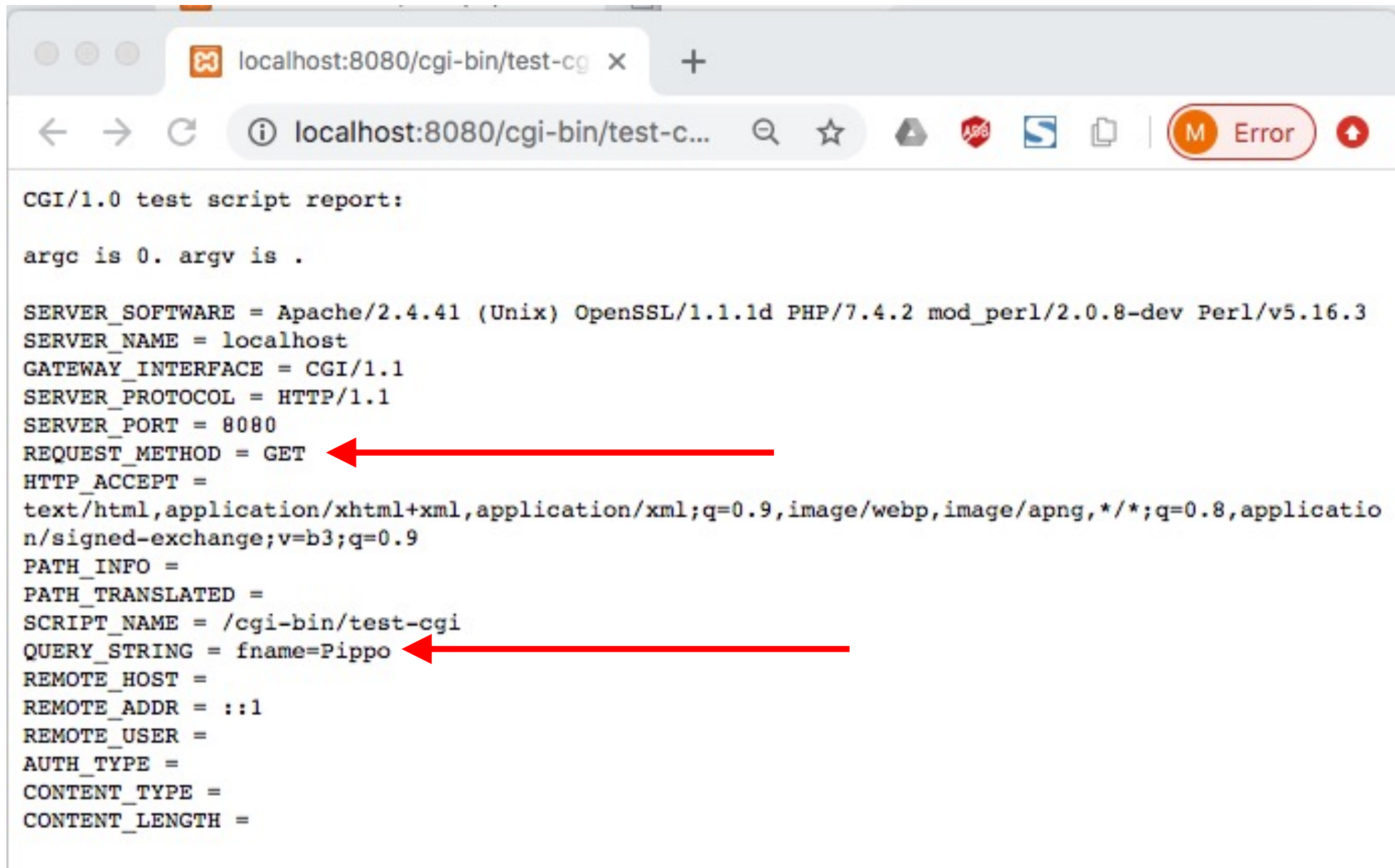
Parameters collection and passing

(HTML Forms)

```
<html>
<body>
<form action="http://localhost:8080/cgi-bin/test-cgi" method="GET">
  <label for="fname">First name:</label>
  <input type="text" name="fname"><br><br>
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
</body>
</html>
```



Parameters passing



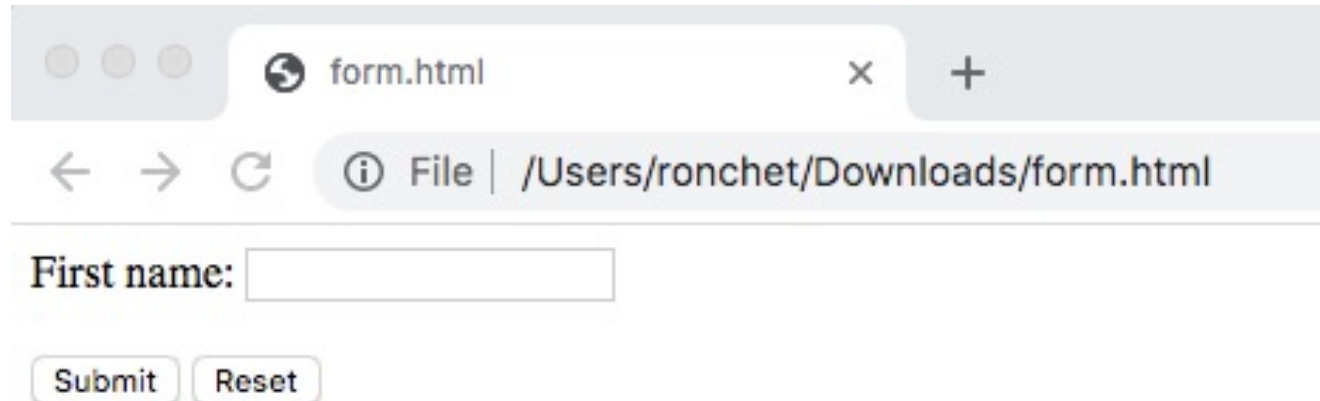
```
CGI/1.0 test script report:

argc is 0. argv is .

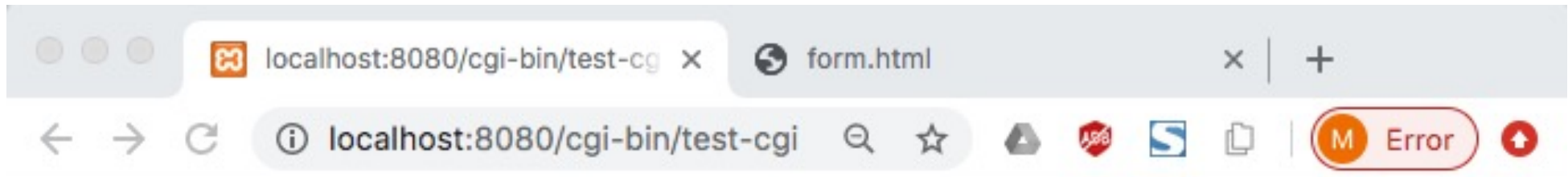
SERVER_SOFTWARE = Apache/2.4.41 (Unix) OpenSSL/1.1.1d PHP/7.4.2 mod_perl/2.0.8-dev Perl/v5.16.3
SERVER_NAME = localhost
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.1
SERVER_PORT = 8080
REQUEST_METHOD = GET
HTTP_ACCEPT =
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/test-cgi
QUERY_STRING = fname=Pippo
REMOTE_HOST =
REMOTE_ADDR = ::1
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =
```

Parameters passing

```
<html>
<body>
<form action="http://localhost:8080/cgi-bin/test-cgi" method="post">
  <label for="fname">First name:</label>
  <input type="text" name="fname"><br><br>
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
</body>
</html>
```



Parameters passing



```
CGI/1.0 test script report:
```

```
argc is 0. argv is .
```

```
SERVER_SOFTWARE = Apache/2.4.41 (Unix) OpenSSL/1.1.1d PHP/7.4.2 mod_perl/2.0.8-dev Perl/v5.16.3
SERVER_NAME = localhost
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.1
SERVER_PORT = 8080
REQUEST_METHOD = POST
HTTP_ACCEPT =
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/test-cgi
QUERY_STRING =
REMOTE_HOST =
REMOTE_ADDR = ::1
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE = application/x-www-form-urlencoded
CONTENT_LENGTH = 11
```

?? Where are the data ??

Let's read the POST DATA

readPost.sh

```
#!/bin/sh
```

```
read MYDATA
```

```
echo "Content-type: text/plain; charset=iso-8859-1"
```

```
echo
```

```
echo $MYDATA
```

This is just an arbitrarily chosen variable name

```
<html>
```

```
<body>
```

```
<form action="http://localhost:8080/cgi-bin/readPost.sh" method="post">
```

```
<label for="fname">First name:</label>
```

```
<input type="text" name="fname" "><br><br>
```

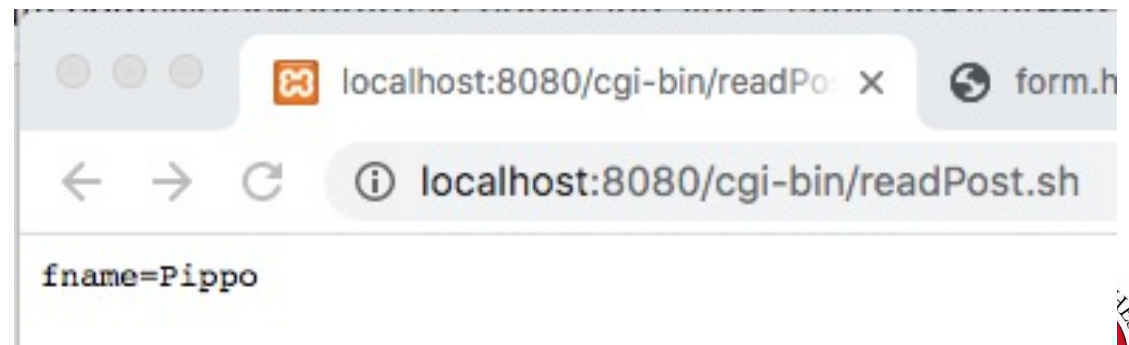
```
<input type="submit" value="Submit">
```

```
<input type="reset" value="Reset">
```

```
</form>
```

```
</body>
```

```
</html>
```



Let's read the POST DATA (full page)

readPost.sh

```
#!/bin/sh
read MYDATA
echo "Content-type: text/plain; charset=iso-8859-1"
echo
echo "<HTML>"
echo "<HEAD><TITLE>Showing Post Data </TITLE>"
echo "<BODY>"
echo "here are the post data:<br>"
echo $MYDATA
echo "</BODY></HTML> "
```

HTML is “printed out”. Can we do better?



Step 4: introducing “Web server languages”



Let's read the POST DATA (full page)

readPost.sh

```
#!/bin/sh
read MYDATA
echo "Content-type: text/plain; charset=iso-8859-1"
echo
echo "<HTML>"
echo "<HEAD><TITLE>Showing Post Data </TITLE>"
echo "<BODY>"
echo "here are the post data:<br>"
echo $MYDATA
echo "</BODY></HTML> "
```

HTML is “printed out”. Can we do better?



A simpler idea

Instead of embedding
HTML into the code, we
could embed the code into
HTML...

Template page

```
<HTML>  
SOME HTML STUFF  
%SOME CODE%  
MORE HTML STUFF  
</HTML>
```



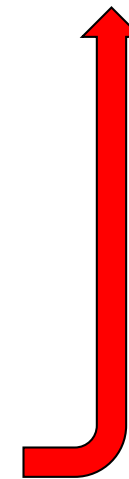
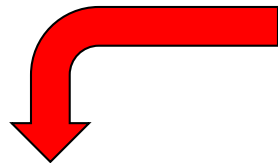
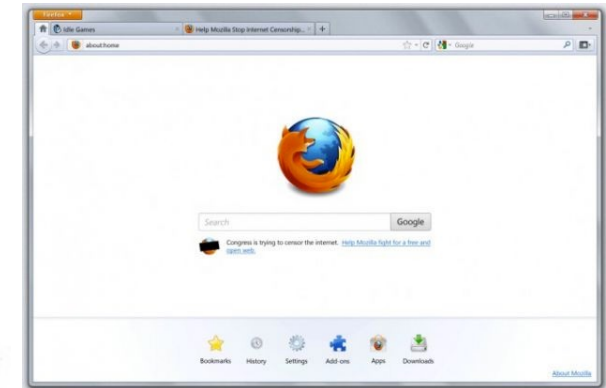
```
<HTML>  
SOME HTML STUFF  
CODE OUTPUT  
MORE HTML STUFF  
</HTML>
```

Augment the web server with an engine capable of parsing a web page, and executing code in it

A simpler idea

Retrieve template page from file system

Request URL



Response

```
<HTML>
SOME HTML STUFF
%SOME CODE%
MORE HTML STUFF
</HTML>
```

Evaluate code



```
<HTML>
SOME HTML STUFF
CODE OUTPUT
MORE HTML STUFF
</HTML>
```



PHP – PHP Hypertext Processor

originally Personal Home Page

whatsTheTime.php

```
<!DOCTYPE html>
<html>
<body>
<i>Sir, the current time is
<?php
echo date("h:i:sa");
?>
</i><br/>
(as far as I know!)
</body>
</html>
```

Sir, the current time is 04:20:11pm
(as far as I know!)

PHP: the language

Interpreted

Non-typed language

Case insensitive



PHP Elements

■ Variables

In PHP, a variable starts with the `$` sign, followed by the name of the variable:

Example

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

- String
 - Integer
 - Float (floating point numbers - also called double)
 - Boolean
 - Array
 - Object
 - NULL
 - Resource
- ## ■ Data types

■ Comments

```
<!DOCTYPE html>
<html>
<body>

<?php
// This is a single-line comment

# This is also a single-line comment
?>

</body>
</html>
```

- Type is implicitly assigned by interpreter
- PHP 7 introduces type declaration and enforcement

■ output

PHP Elements

```
<?php
print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

```
<?php
$x = 100;           int(100)
var_dump($x);
?>
```

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```

echo and print are more or less the same.
They are both used to output data to the screen.

echo has no return value
print has a return value of 1 so it can be used in expressions.

echo can take multiple parameters (although such usage is rare)
print can take one argument.



Gym, testing and deployment

Drop your .php files in your htdocs directory, and invoke them from a browser

Alternatives (quicker for learning):

- 1) <https://www.jdoodle.com/php-online-editor/>
- 2) <https://www.w3schools.com/php/default.asp>



jdoodle

← → ↻ <https://www.jdoodle.com/php-online-editor/>

This site is ad supported! Please consider disable

Online PHP IDE

```
1 - <html>Hello<br>
2   <?php
3
4     $x=10;
5     $y=25;
6     $z=$x+$y;
7
8     $msg = 'Sum of x+y = ';
9
10    print($msg.$z);
11
12   ?>
13 </html>
```

Execute Mode, Version, Inputs & Arguments

7.3.10

Interactive

Stdin Inputs

CommandLine Arguments

▶ Execute



Result

CPU Time: 0.01 sec(s), Memory: 21832 kilobyte(s)

```
<html>Hello<br>
Sum of x+y = 35</html>
```

ento



PHP Tutorial

PHP HOME

- PHP Intro
- PHP Install
- PHP Syntax
- PHP Comments
- PHP Variables
- PHP Echo / Print
- PHP Data Types
- PHP Strings
- PHP Numbers
- PHP Constants
- PHP Operators
- PHP If...Else...Elseif
- PHP Switch
- PHP Loops
- PHP Functions
- PHP Arrays
- PHP Superglobals

PHP Forms

- PHP Form Handling
- PHP Form Validation
- PHP Form Required
- PHP Form URL/E-mail
- PHP Form Complete

PHP 7 is the latest stable release.

Easy Learning with "PHP Tryit"

With our online "PHP Tryit" editor, you can edit the PHP code, and click on a button to view the result.

Example

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

Try it Yourself »

Click on the "Try it Yourself" button to see how it works.

[Start learning PHP now!](#)



PHP Language constructs

■ Flow control

- `if` statement - executes some code if one condition is true
- `if...else` statement - executes some code if a condition is true and another code if that condition is false
- `if...elseif...else` statement - executes different codes for more than two conditions
- `switch` statement - selects one of many blocks of code to be executed

https://www.w3schools.com/php/php_if_else.asp

https://www.w3schools.com/php/php_switch.asp

■ Iteration

- `while` - loops through a block of code as long as the specified condition is true
- `do...while` - loops through a block of code once, and then repeats the loop as long as the specified condition is true
- `for` - loops through a block of code a specified number of times
- `foreach` - loops through a block of code for each element in an array

https://www.w3schools.com/php/php_looping.asp



PHP - Arrays

- Indexed arrays

```
$cars = array("Volvo", "BMW", "Toyota");  
  
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";
```

- Associative arrays

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
  
$age['Peter'] = "35";  
$age['Ben'] = "37";  
$age['Joe'] = "43";
```

PHP - Arrays

- Example

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " .
$cars[1] . " and " . $cars[2] . ".";
echo count($cars);
?>
```

- GLOBALS

```
<?php
$x = 75;
$y = 25;
function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z;
?>
```



PHP Functions

Syntax

```
function functionName() {  
    code to be executed;  
}
```

A function name must start with a letter or an underscore.
Function names are NOT case-sensitive.

```
<?php  
function addNumbers(int $a, int $b) {  
    return $a + $b;  
}  
echo addNumbers(5, "5 days");  
?>
```

// since strict is NOT enabled "5 days" is changed to int(5)

Output: 10

PHP Functions

Syntax

```
function functionName() {  
    code to be executed;  
}
```

A function name must start with a letter or an underscore.
Function names are NOT case-sensitive.

- Strict data types

```
<?php declare(strict_types=1); // strict requirement  
  
function addNumbers(int $a, int $b) {  
    return $a + $b;  
}  
echo addNumbers(5, "5 days");  
?>
```

Output: since strict is enabled and "5 days" is not an integer, an error will be thrown

To specify strict we need to set `declare(strict_types=1);`. This must be on the very first line of the PHP file.



PHP Operators

- String operators

Same as in Java:

- Arithmetic operators +,/,%,*,**
- Increment, decrement operators ++, --
- Assignment operators +=, -= ...
- Logical operators &&, ||, ! but also and, or, not

Operator	Name	Example	Result
.	Concatenation	\$txt1 . \$txt2	Concatenation of \$txt1 and \$txt2
.=	Concatenation assignment	\$txt1 .= \$txt2	Appends \$txt2 to \$txt1

- Array operators

Operator	Name	Example	Result
+	Union	\$x + \$y	Union of \$x and \$y
==	Equality	\$x == \$y	Returns true if \$x and \$y have the same key/value pairs
===	Identity	\$x === \$y	Returns true if \$x and \$y have the same key/value pairs in the same order and of the same types
!=	Inequality	\$x != \$y	Returns true if \$x is not equal to \$y
<>	Inequality	\$x <> \$y	Returns true if \$x is not equal to \$y
!==	Non-identity	\$x !== \$y	Returns true if \$x is not identical to \$y



PHP Comparison Operators

Operator	Name	Example	Result
==	Equal	<code>\$x == \$y</code>	Returns true if <code>\$x</code> is equal to <code>\$y</code>
===	Identical	<code>\$x === \$y</code>	Returns true if <code>\$x</code> is equal to <code>\$y</code> , and they are of the same type
!=	Not equal	<code>\$x != \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<>	Not equal	<code>\$x <> \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
!==	Not identical	<code>\$x !== \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code> , or they are not of the same type
>	Greater than	<code>\$x > \$y</code>	Returns true if <code>\$x</code> is greater than <code>\$y</code>
<	Less than	<code>\$x < \$y</code>	Returns true if <code>\$x</code> is less than <code>\$y</code>
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if <code>\$x</code> is greater than or equal to <code>\$y</code>
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if <code>\$x</code> is less than or equal to <code>\$y</code>
<=>	Spaceship	<code>\$x <=> \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if <code>\$x</code> is less than, equal to, or greater than <code>\$y</code> . Introduced in PHP 7.

See https://www.w3schools.com/php/php_operators.asp

Example

```
<!DOCTYPE html>
<html>
<body>
<?php
function isEqual($x1,$s1,$y1,$result) {
    var_dump($x1);
    echo $s1;
    var_dump($y1);
    echo " : ";
    var_dump($result);
    echo "<BR>";
}
$x = 100;
$y = "100";
$z = 100;
isEqual($x," == ",$y,$x == $y);
isEqual($x," === ",$y,$x === $y);
isEqual($x," == ",$z,$x == $z);
isEqual($x," === ",$z,$x === $z);?>
</body>
</html>
```

■ Output

```
int(100) == string(3) "100" : bool(true)
int(100) === string(3) "100" : bool(false)
int(100) == int(100) : bool(true)
int(100) === int(100) : bool(true)
```

PHP Variable scope

A variable declared **outside a function** has a **GLOBAL SCOPE** and can only be accessed outside a function.

A variable declared **within a function** has a **LOCAL SCOPE** and can only be accessed within that function

```
<?php
$x = 5;
$y = 25;
print $x+$y;
function f() {
    $z = 2;
    print $x;
}
print $z;
f();
print $z;
?>
```

\$GLOBALS marks variables
as global vars

=> 30

=> *(nothing)*

=> *(nothing)*

=> *(nothing)*

30 <=

(nothing) <=

5 <=

2 <=

OUTPUT

30

3052

```
<?php
$GLOBALS['x'] = 5;
$y = 25;
print $x+$y;
function f() {
    $GLOBALS['z'] = 2;
    print $GLOBALS['x'];
}
print $GLOBALS['z'];
f();
print $GLOBALS['z'];
?>
```

PHP Superglobals variables

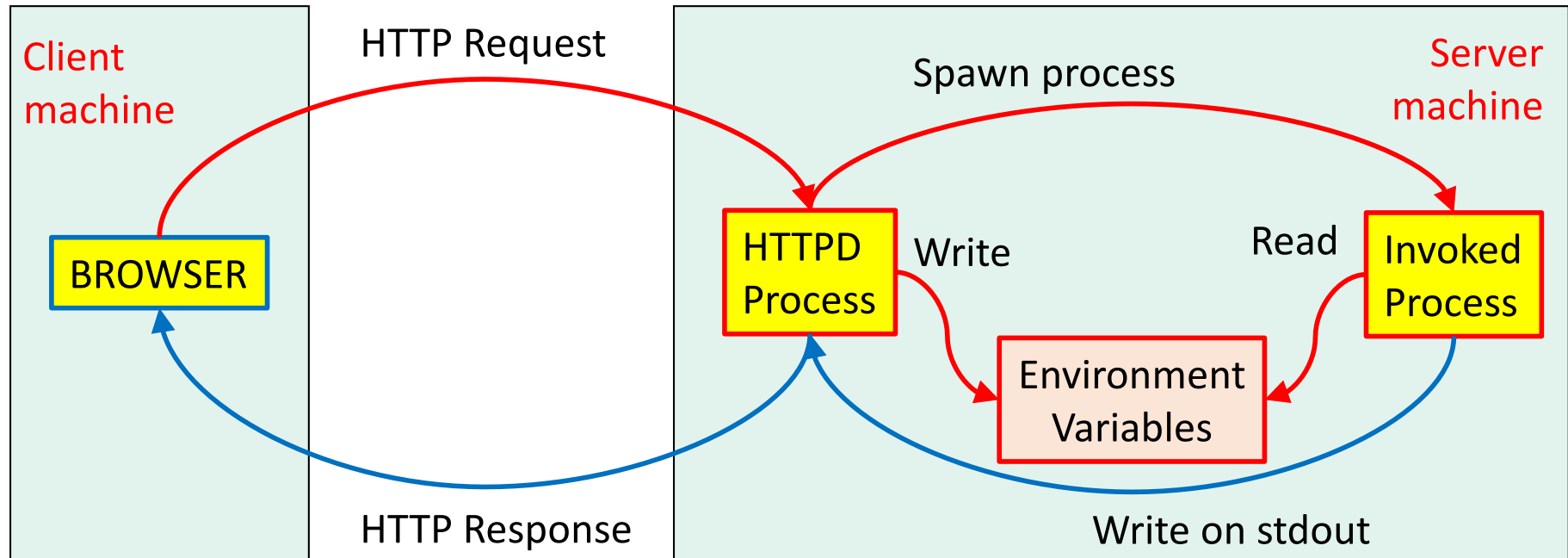
A **superglobal** variable is **visible everywhere**.

The PHP superglobal variables are:

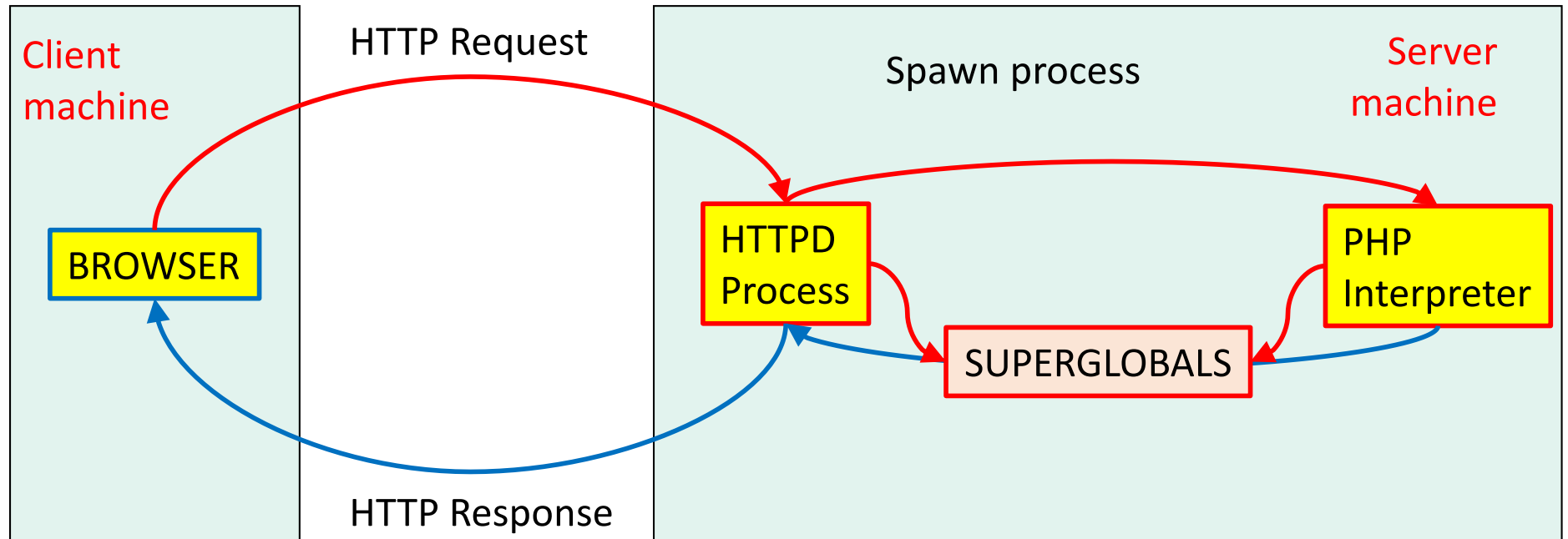
- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`
- `$_FILES`
- `$_ENV`
- `$_COOKIE`
- `$_SESSION`



Getting info about the request



Getting info about the request



Some superglobals:

- SERVER -> Info mostly extracted from http request headers
 - https://www.w3schools.com/php/php_superglobals_server.asp
- POST
- GET
 - Associative arrays, with key=name



PHP Predefined functions

- Lots!

Array	Calendar	Date	Directory	Error
Filesystem	Filter	FTP	Libxml	Mail
Math	Misc	MySQLi	Network	SimpleXML
Stream	String	XML Parser	Zip	Timezones

See: https://www.w3schools.com/php/php_ref_overview.asp



PHP OOP

PHP has:

- Classes (with properties and methods)
- Constructors - Destructors
- Inheritance
- Abstract classes
- Access modifiers
- Static methods and properties
- Generalization of Interfaces (Traits)

See

https://www.w3schools.com/php/php_oop_classes_objects.asp



Homework

Go through the following part of the w3school tutorial and exercises

PHP HOME
PHP Intro
PHP Install
PHP Syntax
PHP Comments
PHP Variables
PHP Echo / Print
PHP Data Types
PHP Strings
PHP Numbers
PHP Constants
PHP Operators
PHP If...Else...Elseif
PHP Switch
PHP Loops
PHP Functions
PHP Arrays
PHP Superglobals

PHP Forms
PHP Form Handling
PHP Form Validation
PHP Form Required
PHP Form URL/E-mail
PHP Form Complete



Getting deeper with HTML Forms



Forms

Give to the user the possibility to **send information** to the Web server

The **FORM** tag defines a form and has the following attributes:

- **ACTION** identifies the processing engine
- **ENCTYPE** specifies the MIME type used to pass data to the server (Es. Text/html)

FORM contains the sub-tag (inner tags):

- several tags for collecting data
- An **INPUT** tag **must be** of type **SUBMIT** for sending the data
- An **INPUT** can be of type **RESET** to cancel all the gathered data



Form - input

```
<FORM method="POST" action="/cgi-bin/elabora">
```

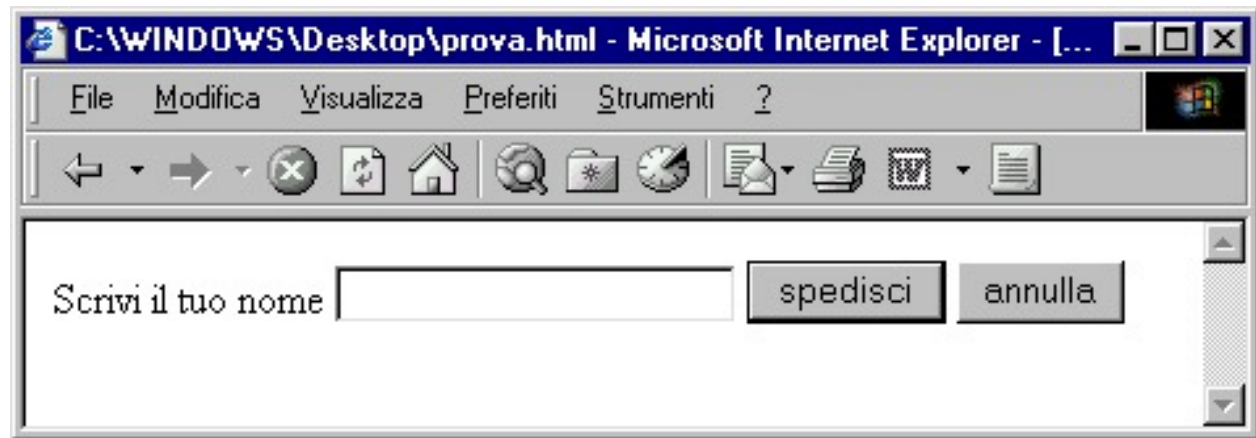
Scrivi il tuo nome

```
<Input type="text" size="25" maxlength="15" name="a">
```

```
<Input type="submit" value="spedisci">
```

```
<Input type="reset" value="annulla">
```

```
</FORM>
```



Sends a url of type

<http://.../cgi-bin/elabora?a=MarcoRonchetti>



Form – input type="radio"

```
<FORM method="POST" action="/cgi-bin/elabora">
```

Fai la tua scelta:

```
<LI><Input type="radio" name="tipo"  
value="auto" checked>Auto
```

```
<LI><Input type="radio" name="tipo"  
value="bus">Bus
```

```
<LI><Input type="radio" name="tipo"  
value="camion">Camion
```

```
<P><Input type="radio" name="colore"  
value="rosso">Rosso
```

```
<Input type="radio" name="colore"  
value="argento" checked>Argento</P>
```

```
<Input type="submit" value="spedisci">
```

```
</FORM>
```



Form – input type="checkbox" - select

```
<FORM method="POST" action="/cgi-bin/elabora">
```

Fai la tua scelta:

```
<LI><Input type="checkbox"
name="tipo" value="auto" checked>Auto
```

```
<LI><Input type="checkbox"
name="tipo" value="bus">Bus
```

```
<LI><Input type="checkbox"
name="tipo" value="camion">Camion
```

```
<P><Select name="colore">
```

```
<option>Rosso
```

```
<option selected>Argento
```

```
</select></P>
```

```
<Input type="submit" value="spedisci">
```

```
</FORM>
```



Form – textarea

```
<FORM method="POST" action="/cgi-bin/elabora">
```

Scrivi i tuoi commenti:

```
<Textarea
```

```
name="commenti" rows="4" columns="14">
```

Spiega in questo spazio la tua opinione

```
</TEXTAREA>
```

```
<Input type="submit" value="via!">
```

```
</FORM>
```



HTML5: many more types!

See https://www.w3schools.com/html/html_form_input_types.asp



Form – more input types

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

https://www.w3schools.com/html/html_form_input_types.asp



Forms – how many buttons?

- Up to HTML 4:
 - At most 2: “exec” and “cancel”
- HTML 5: as many as you want!

```
<form action="/action_page.php">  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"><br><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formaction="/action_page2.php" value="Submit as Admin">  
</form>
```



HTML Form: attributes

Various attributes allow customizing Input and forms, e.g.:

- formaction
- formenctype
- formmethod
- formtarget

W3schools:

HTML Forms
HTML Forms
HTML Form Elements
HTML Input Types
HTML Input Attributes
HTML Input Form Attributes

https://www.w3schools.com/html/html_form_attributes_form.asp



HTML Form, with (some) validation

- https://www.w3schools.com/html/html_forms.asp



HTML Form, with (some) validation

Attribute	Description
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

https://www.w3schools.com/html/html_form_attributes.asp



Forms – restrictions: patterns

```
<form>
```

```
<label for="phone">Enter your phone number:</label>
```

```
<input type="tel" id="phone" name="phone"  
      pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
```

```
</form>
```

The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.

