Step 4:
introducing
“Web server languages”

Let’s read the POST DATA (full page)

readPost.sh

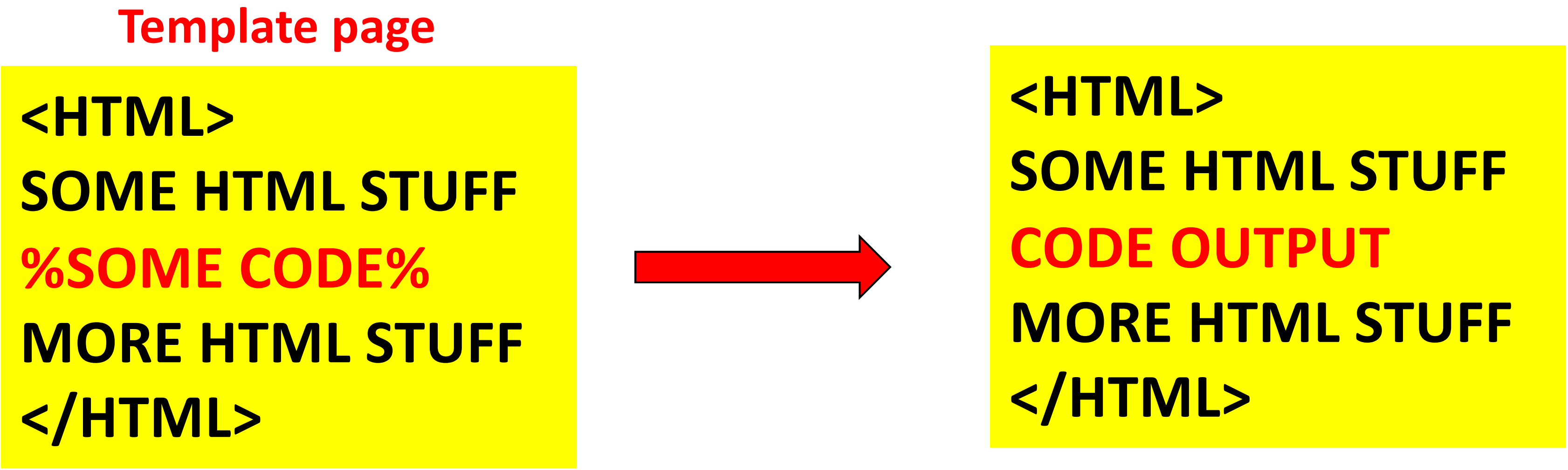
```
#!/bin/sh
read MYDATA
echo "Content-type: text/plain; charset=iso-8859-1"
echo
echo "<HTML>"
echo "<HEAD><TITLE>Showing Post Data </TITLE>”
echo "<BODY>"
echo "here are the post data:<br>"
echo $MYDATA
echo "</BODY></HTML> "
```

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

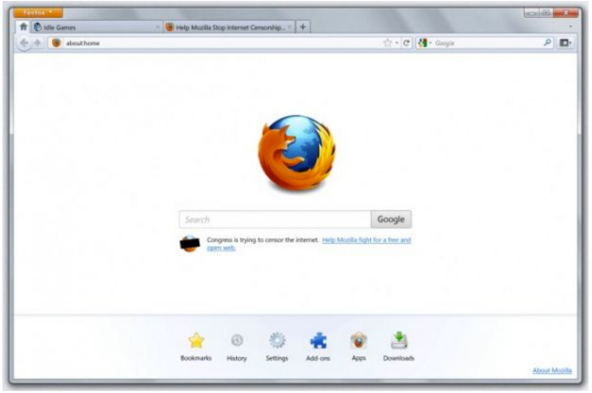
<HTML>
SOME HTML STUFF
%SOME CODE%
MORE HTML STUFF
</HTML>

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

A simpler idea

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

Request URL



Retrieve template page from file system

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

Evaluate code

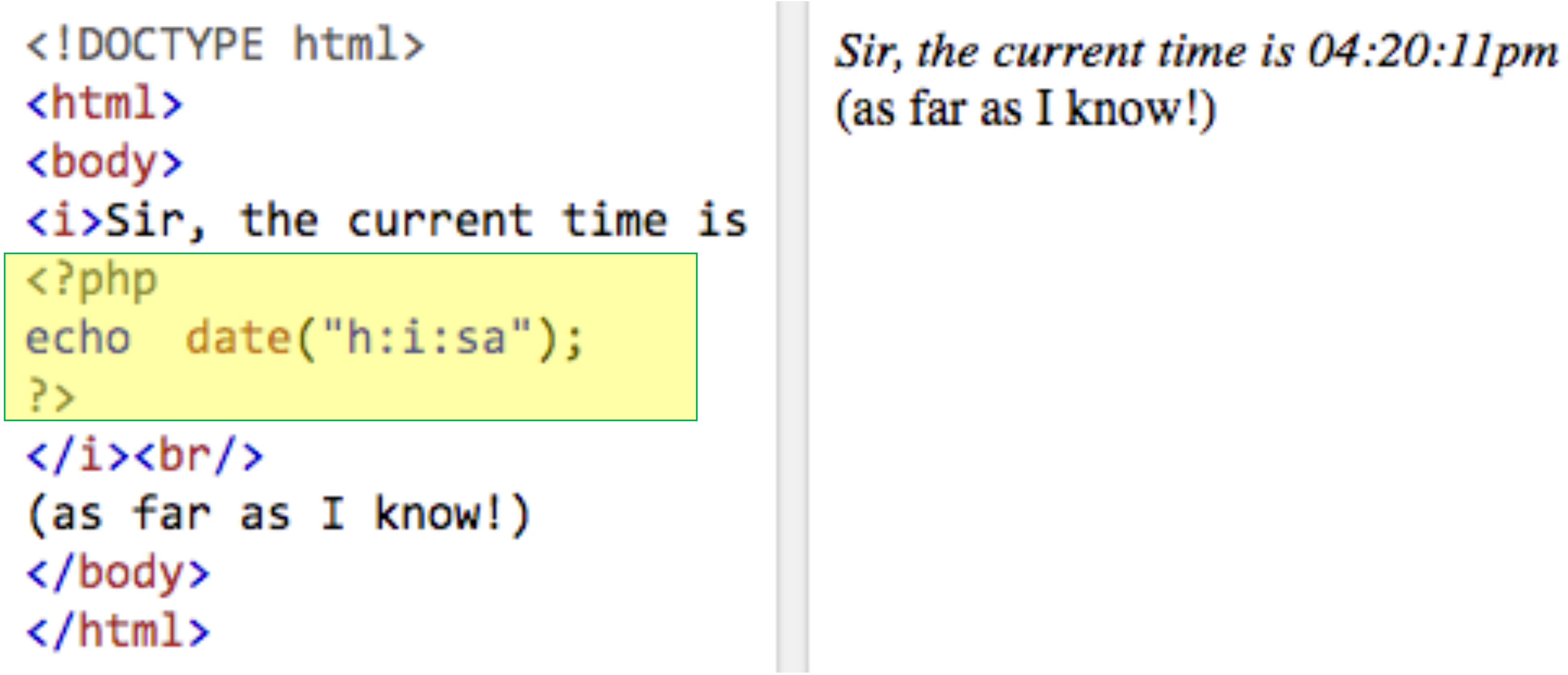
<HTML>
SOME HTML STUFF
CODE OUTPUT
MORE HTML STUFF
</HTML>

Response

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

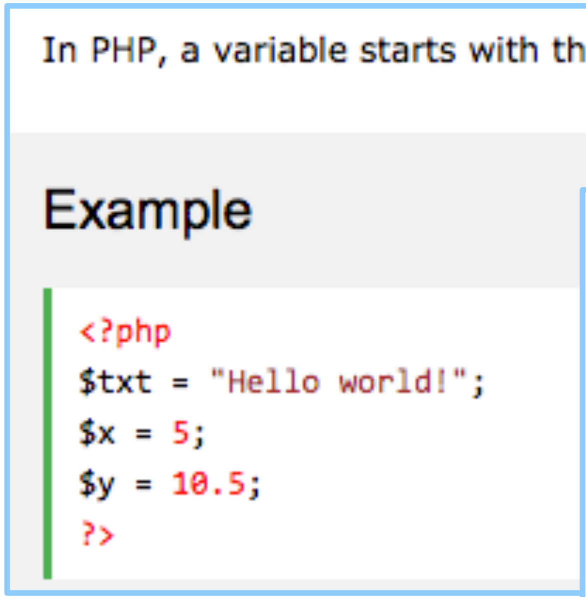
Interpreted

Non-typed language

Case insensitive

PHP Elements

- Variables

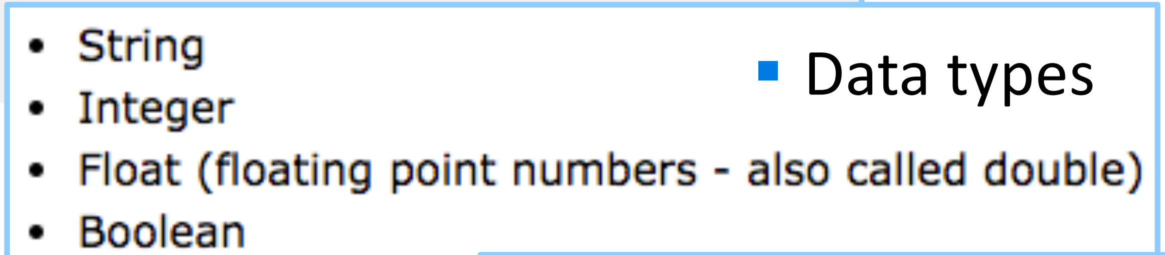


In PHP, a variable starts with the `$` sign, followed by the name of the variable:

Example

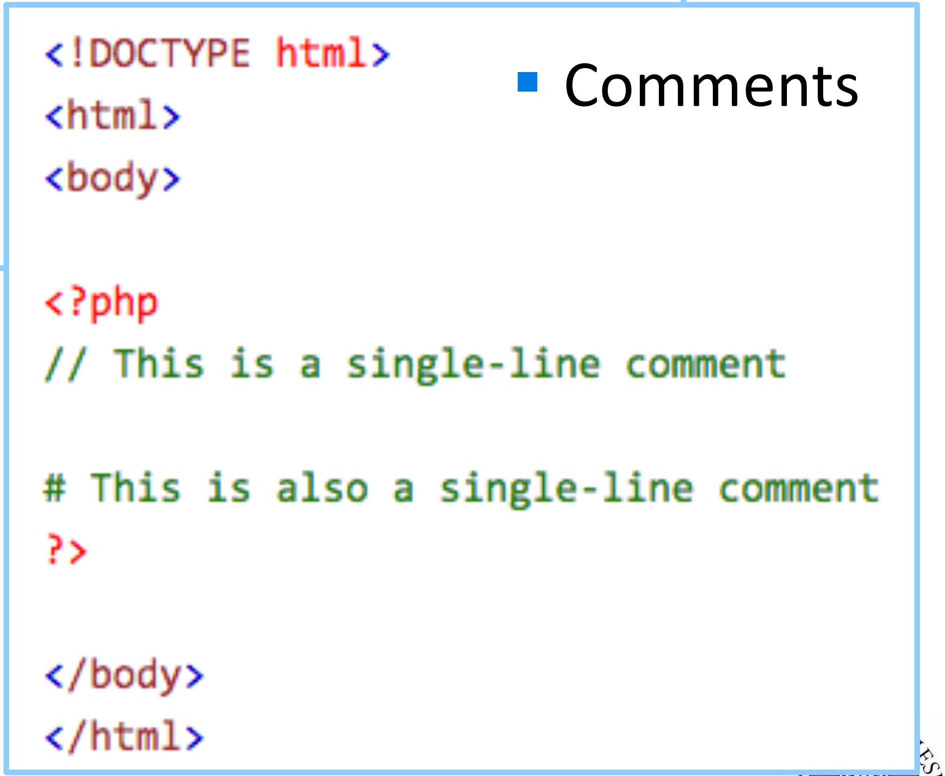
```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

- Data types



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

- Comments

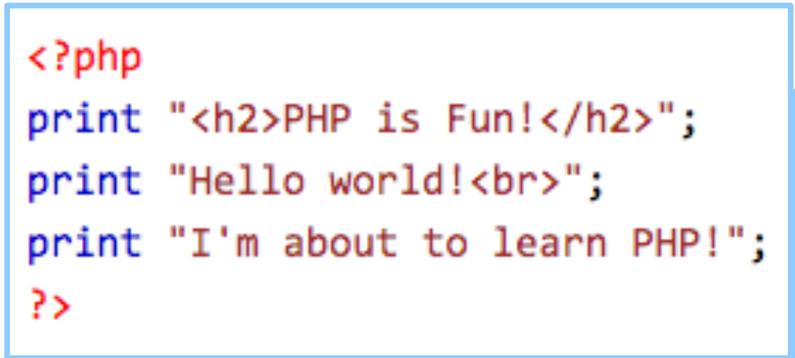


```
<!DOCTYPE html>
<html>
<body>

<?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

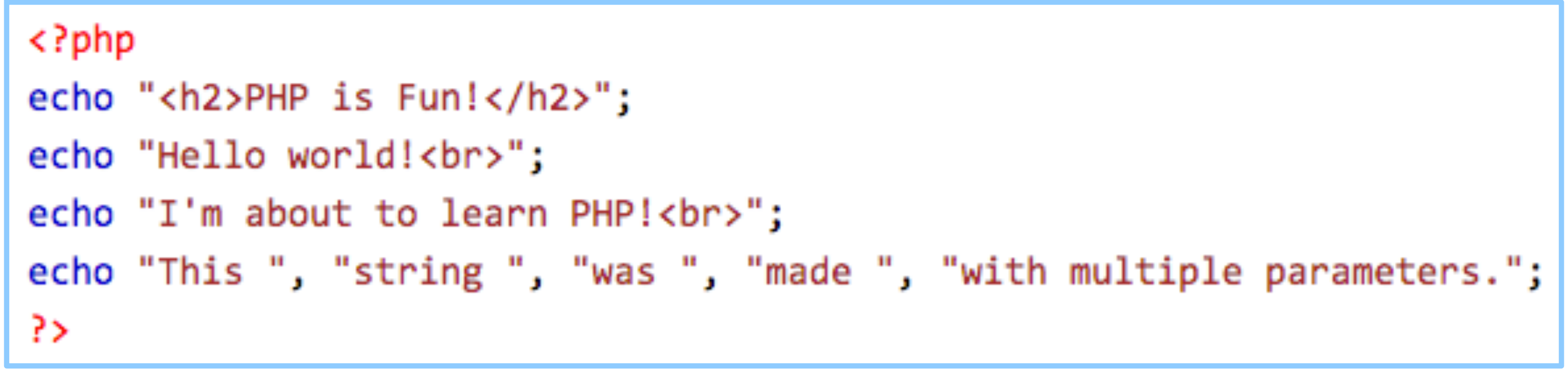
PHP Elements

- output



```
<?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.

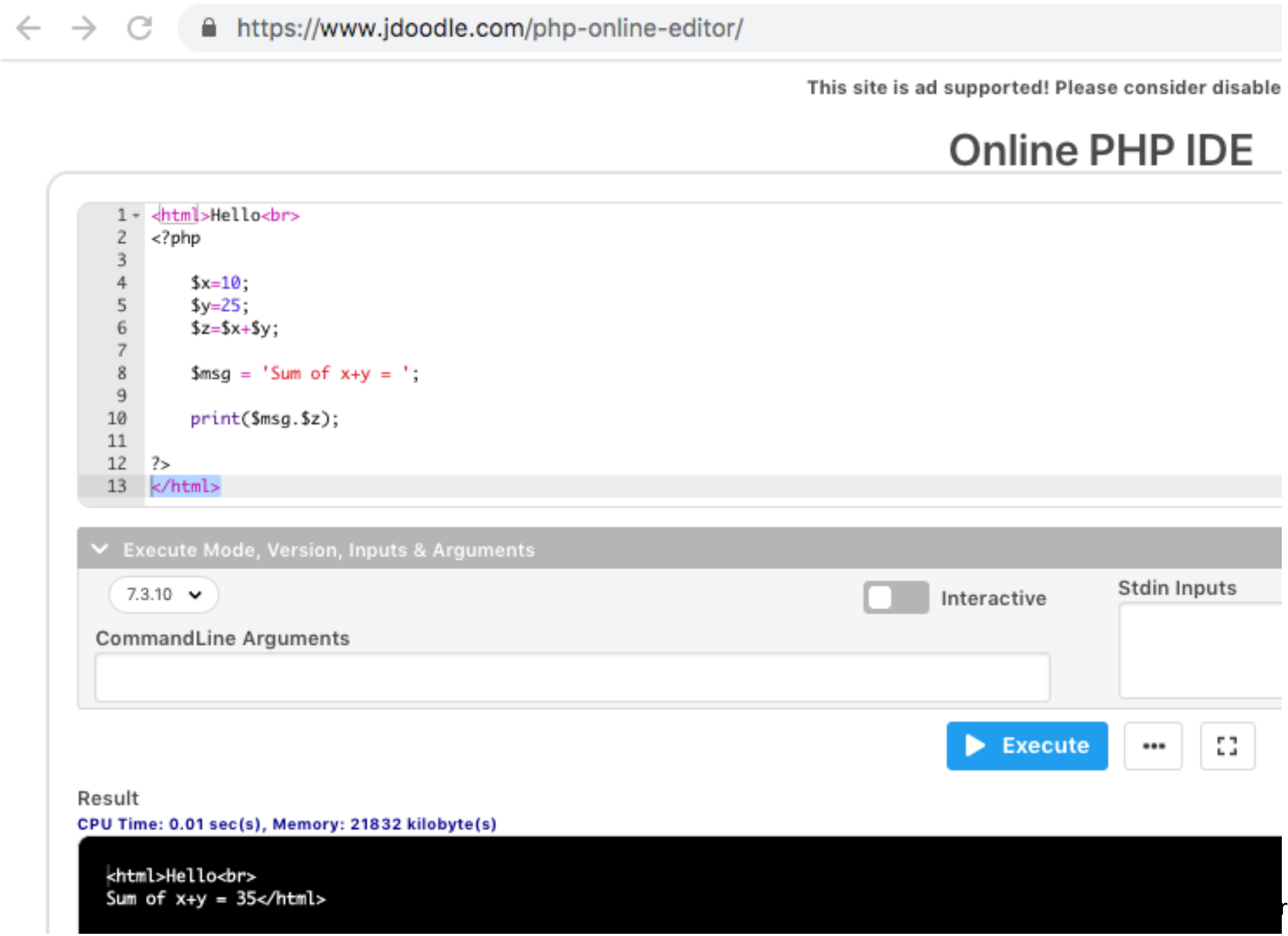
Gym, testing and deployment

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

```
<html>Hello<br>
<?php

    $x=10;
    $y=25;
    $z=$x+$y;

    $msg = 'Sum of x+y = ';

    print($msg.$z);

?>
</html>
```

Execute Mode, Version, Inputs & Arguments

7.3.10

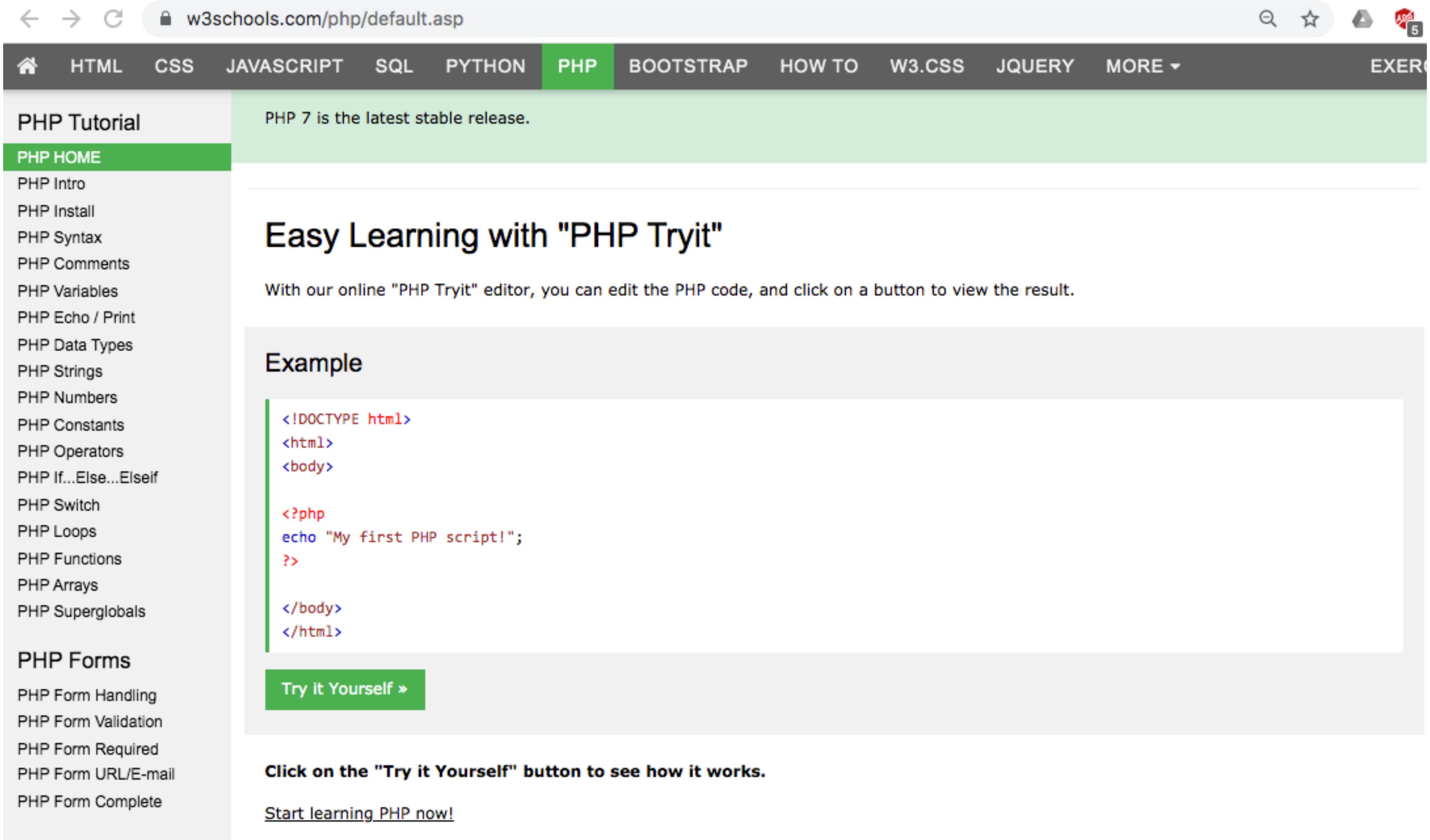
Interactive
Stdin Inputs
CommandLine Arguments
Execute

Result

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

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

w3schools



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

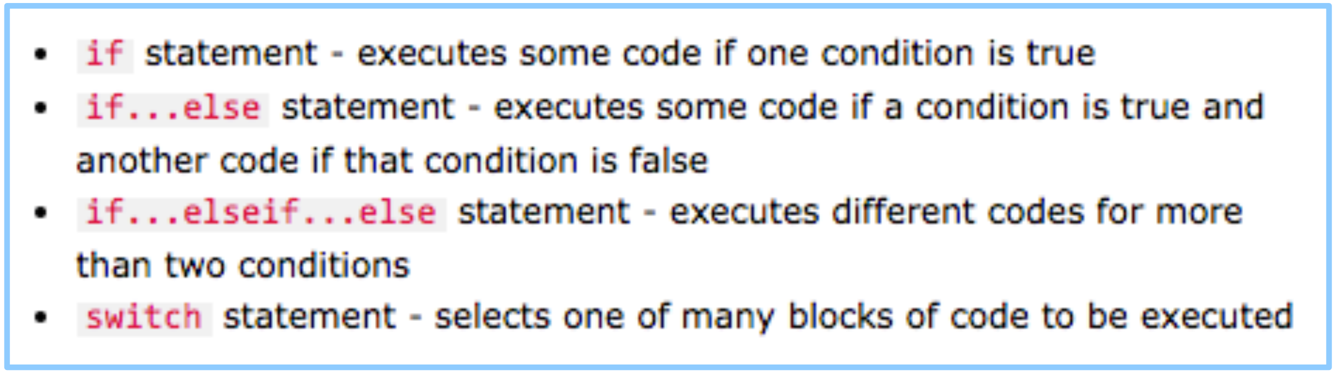
```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

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

Start learning PHP now!

PHP Language constructs

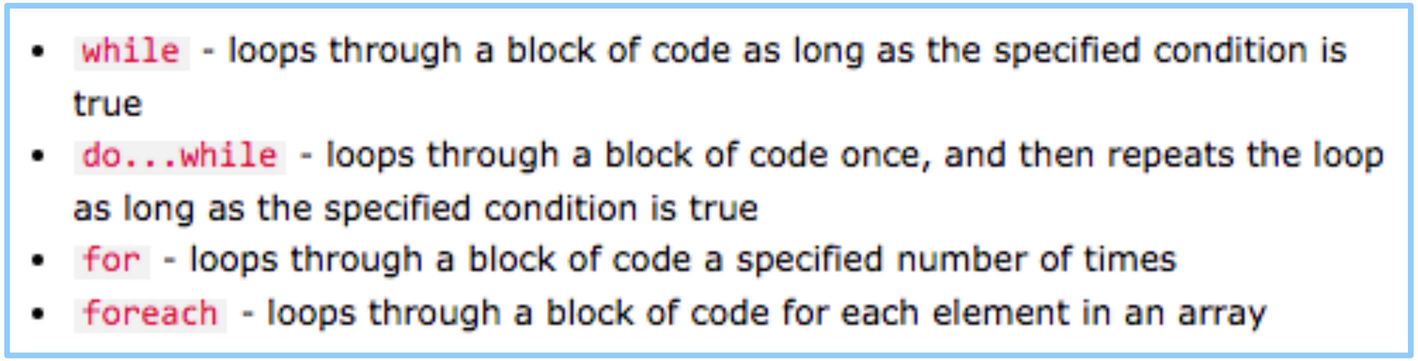
- Flow control



- `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

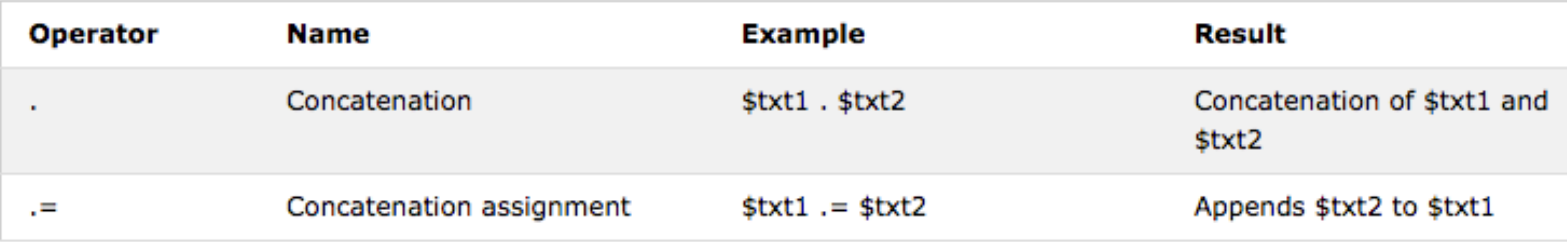
To specify strict we need to set declare(strict_types=1);. This must be on the very first line of the PHP file.

PHP Operators

Same as in Java:

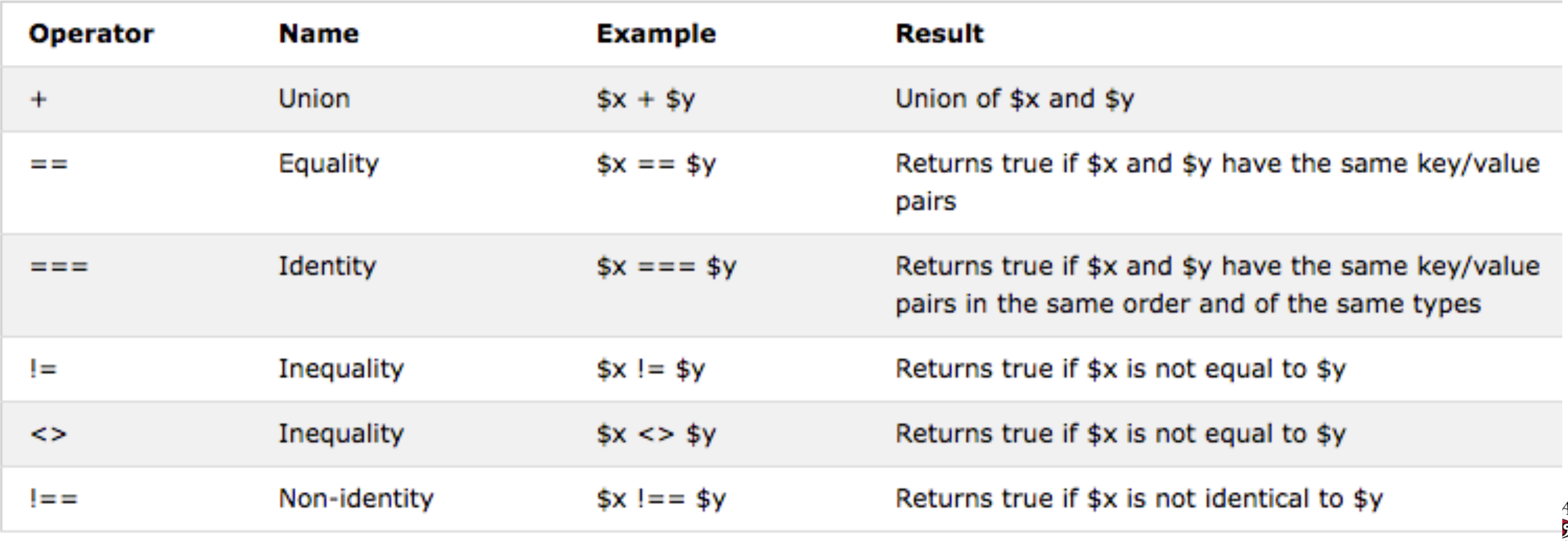
- Arithmetic operators +,/,%,*,**
- Increment, decrement operators ++, --
- Assignment operators +=, -= …
- Logical operators &&, ||, ! but also and, or, not

- String operators



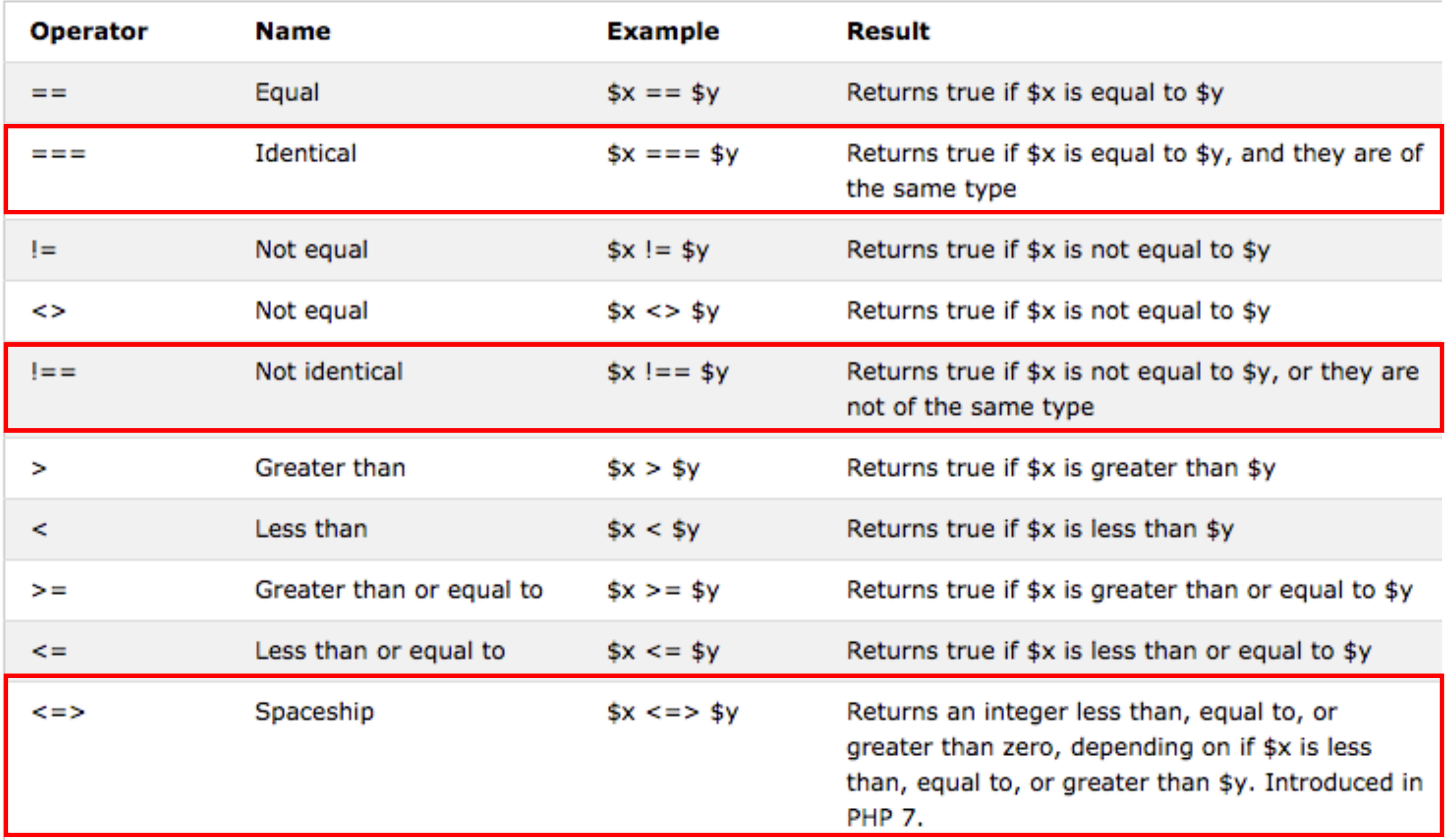
Operator	Name	Example	Result
.	Concatenation	$txt1 . $txt2	Concatenation of $txt1 and $txt2
.=	Concatenation assignment	$txt1 .= $txt2	Appends $txt2 to $txt1

- 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	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y
<=>	Spaceship	$x <=> $y	Returns an integer less than, equal to, or greater than zero, depending on if $x is less than, equal to, or greater than $y. 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;
?>
```

=> 30

=> *(nothing)*

=> *(nothing)*

=> *(nothing)*

$GLOBALS marks variables as global vars

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

30 <=

(nothing) <=

5 <=

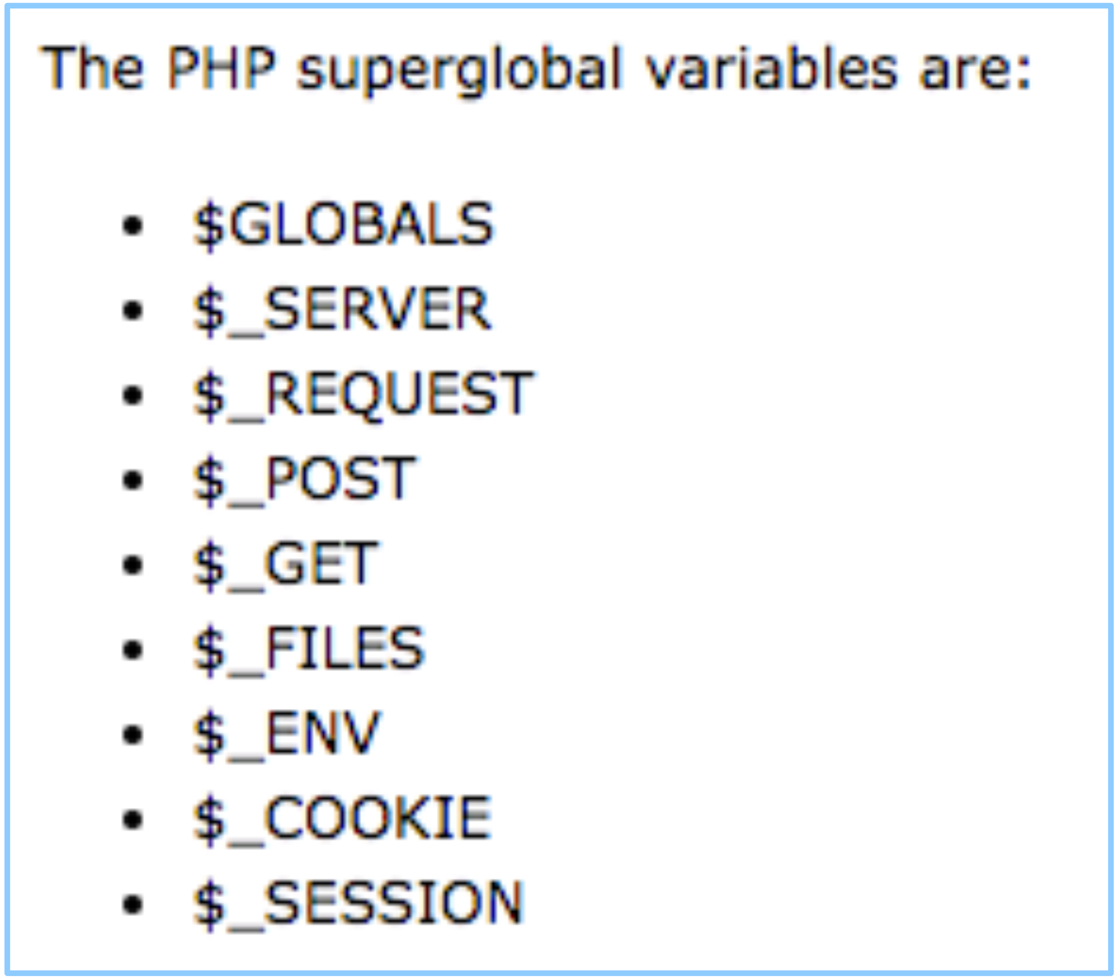
2 <=

OUTPUT

30 3052

PHP Superglobals variables

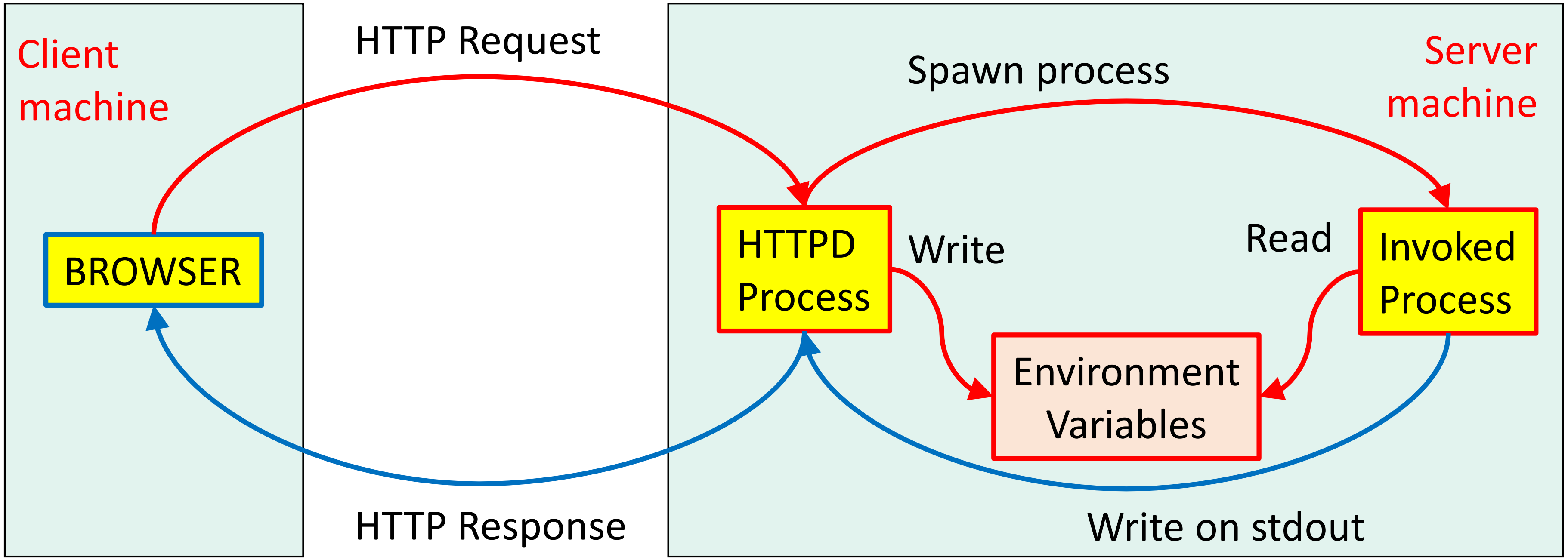
A **superglobal** variable is visible everywhere.



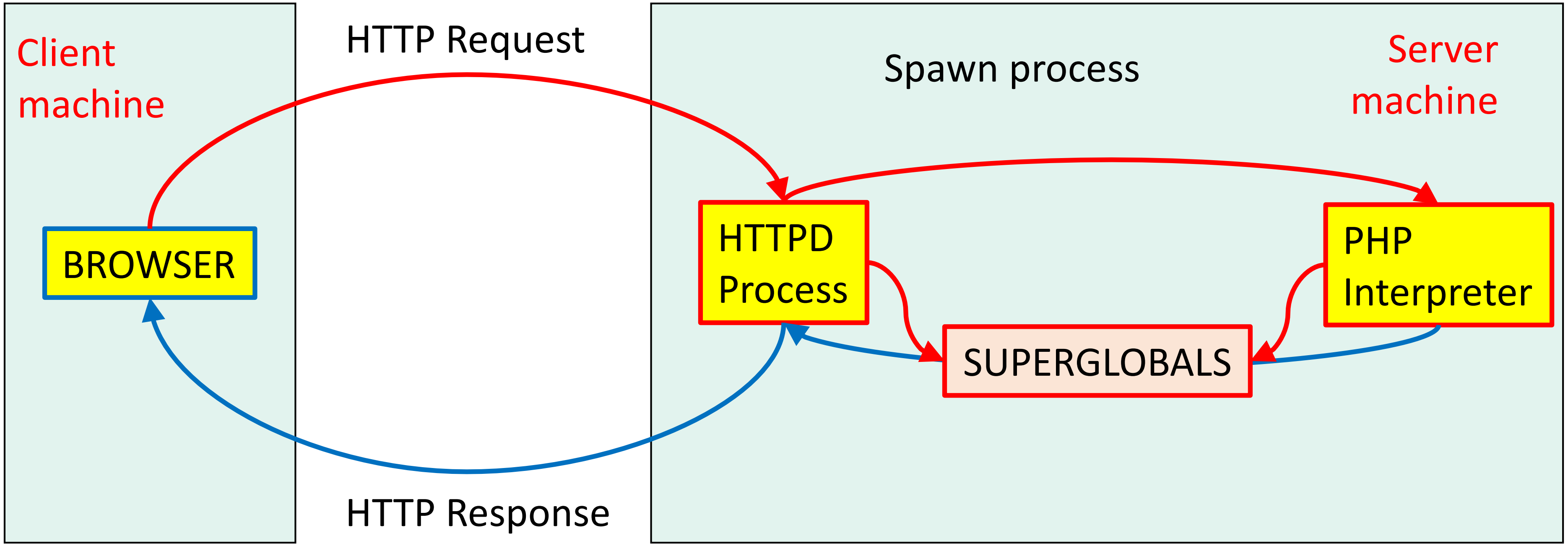
The PHP superglobal variables are:

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

Getting info about the request



Getting info about the request



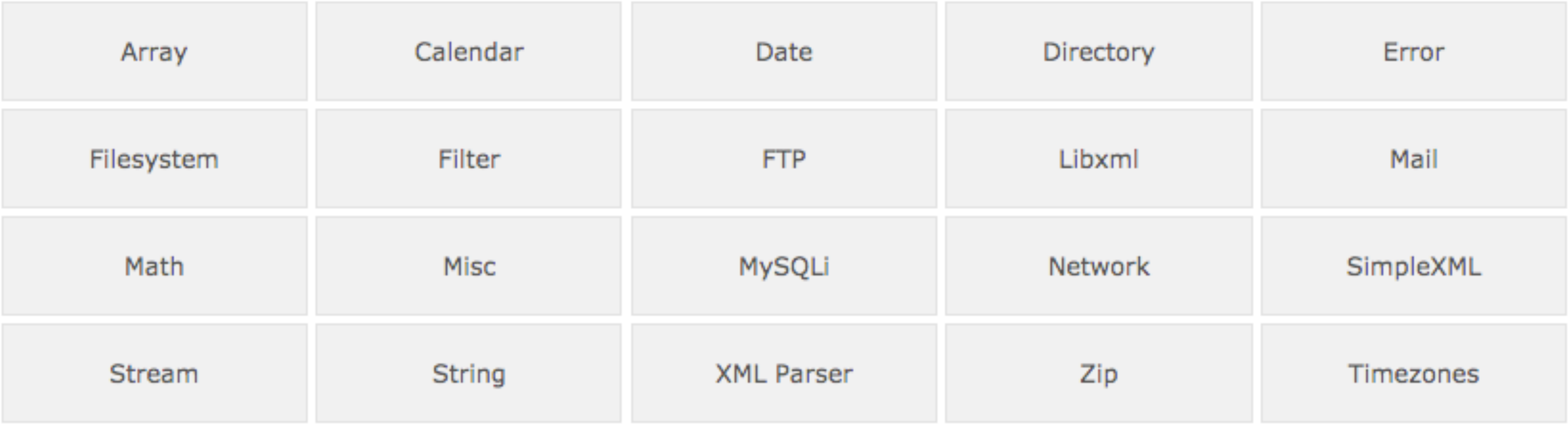
Some superglobals:

- 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

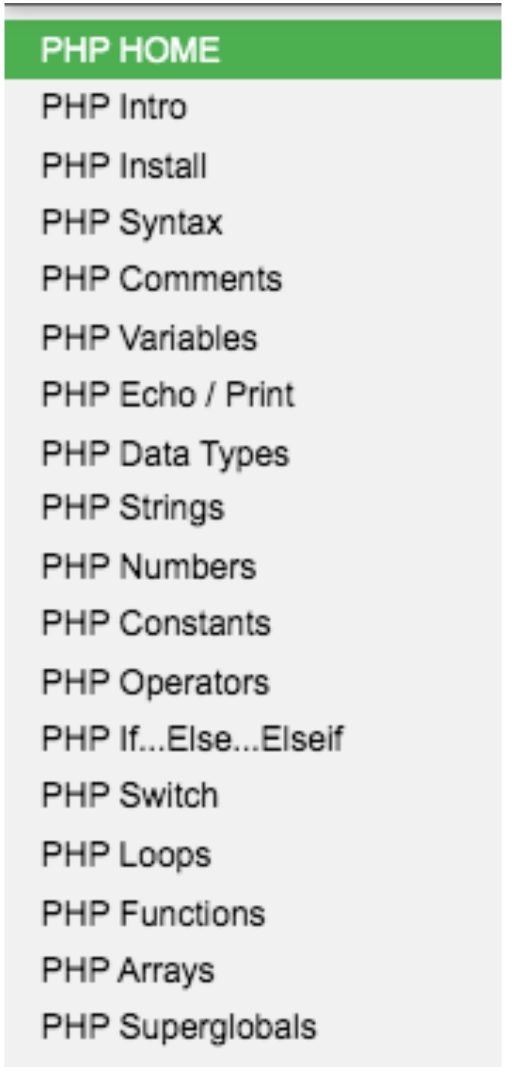
PHP has:

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

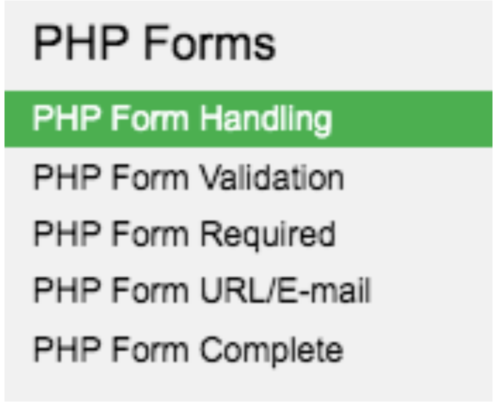
See
https://www.w3schools.com/php/php_oop_classes_objects.asp

Homework

Go though 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:

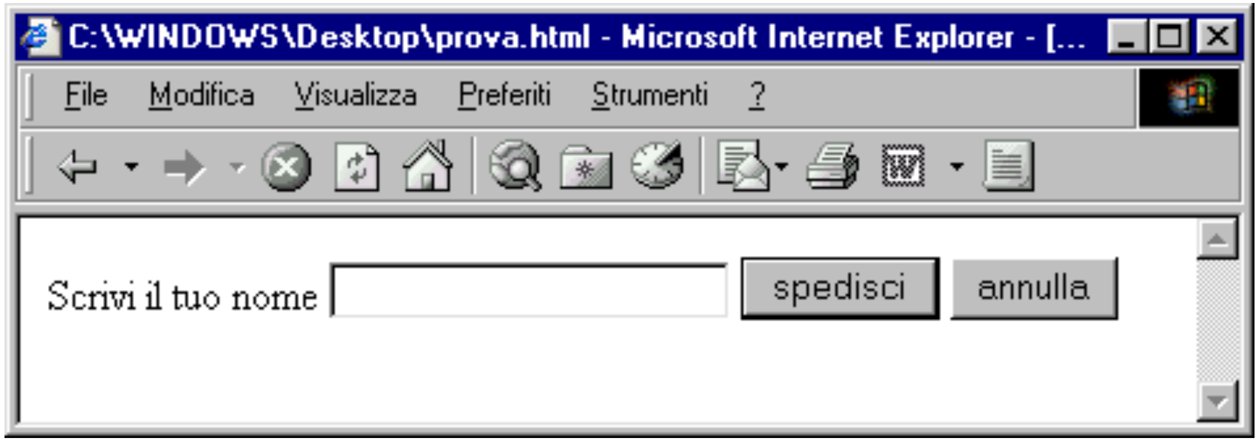
- ACTION identifies the processing engine
- ENCTYPE specificies the MIME type used to pass data to the server (Es. Text/html)

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>
```



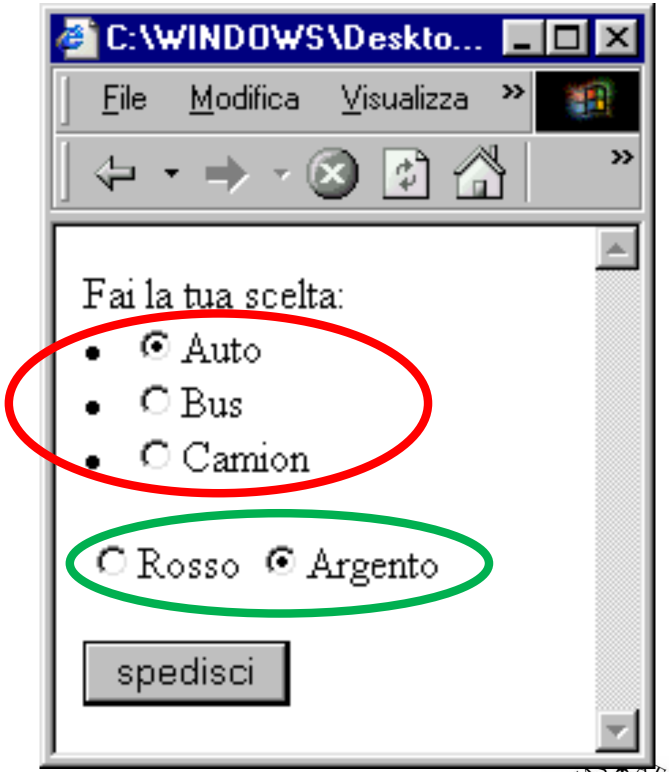
C:\WINDOWS\Desktop\prova.html - Microsoft Internet Explorer - [...

Scrivi il tuo nome

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>
```



Fai la tua scelta:

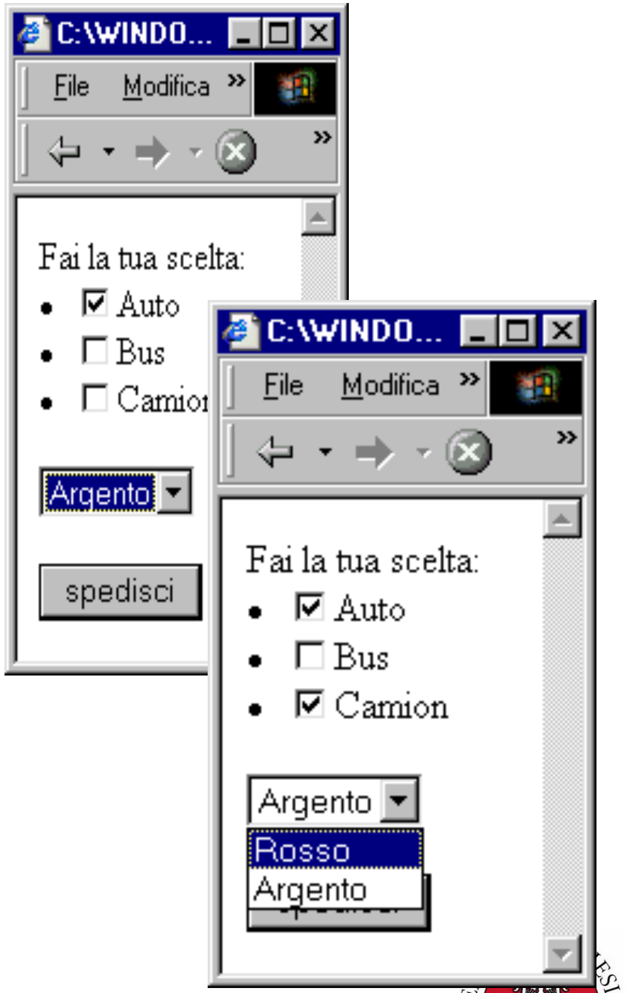
- ● Auto
- ○ Bus
- ○ Camion

○ Rosso ● Argento

spedisci

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>
```



Fai la tua scelta:
- ☑ Auto
- ☐ Bus
- ☐ Camion

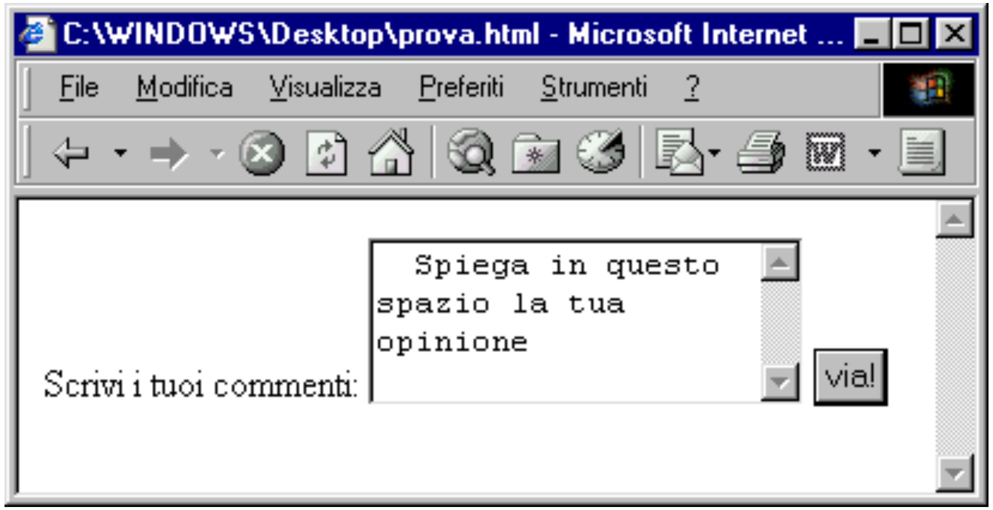
Argento

Fai la tua scelta:
- ☑ Auto
- ☐ Bus
- ☑ Camion

Argento
Rosso
Argento

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>
```



Scrivi i tuoi commenti:

Spiega in questo spazio la tua opinione

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!

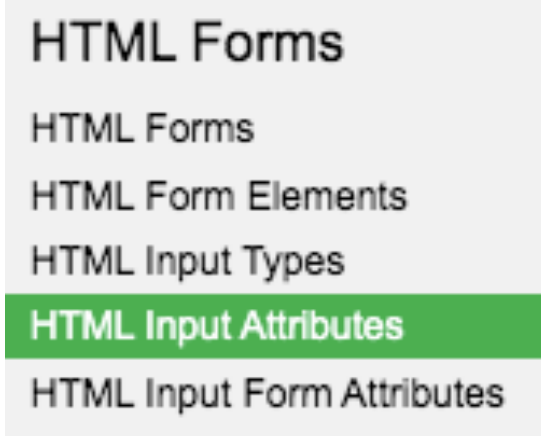
```
<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formaction="/action_page2.php"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

HTML Form: attributes

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

- formaction
- formenctype
- formmethod
- formtarget

W3schools:

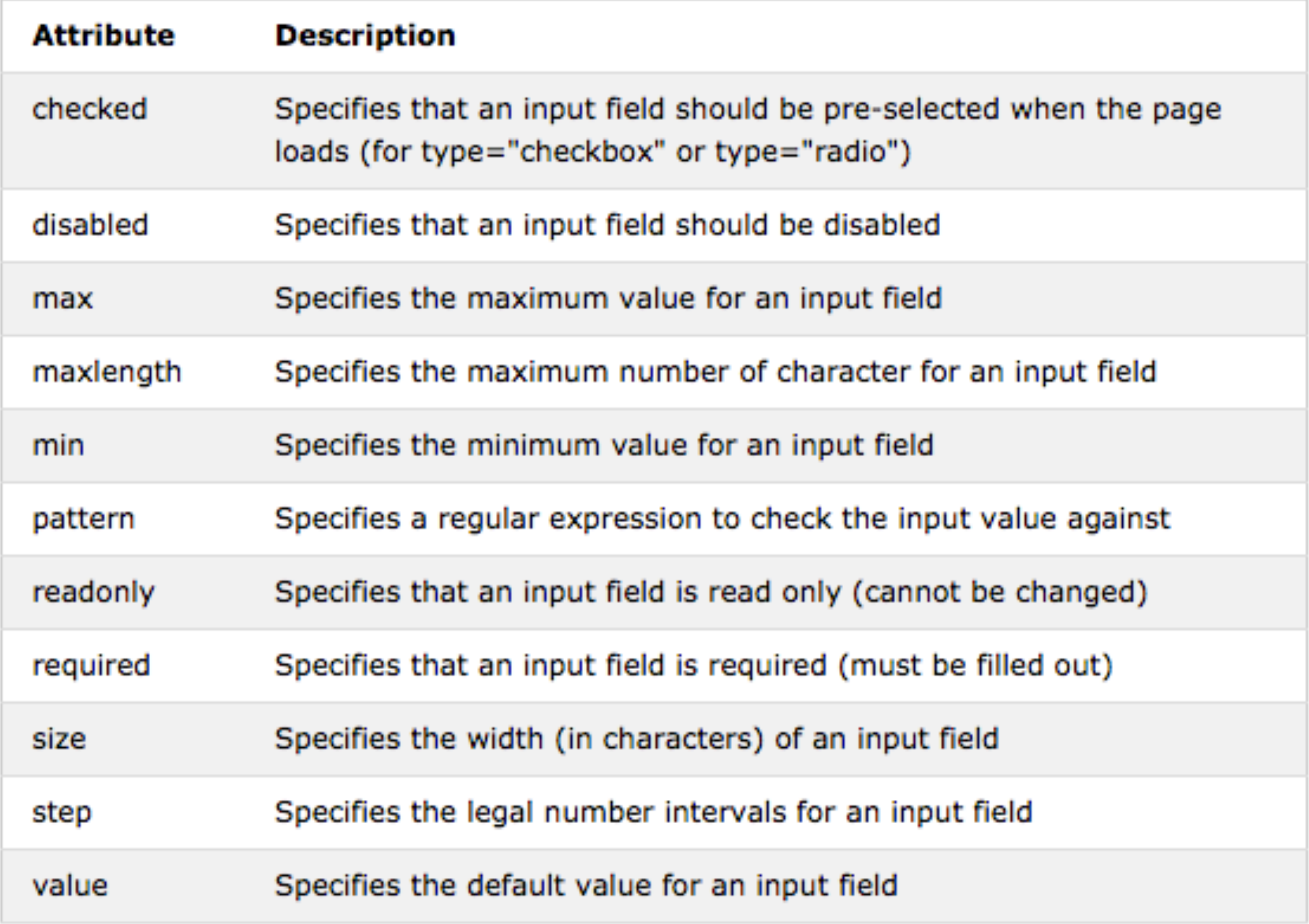


https://www.w3schools.com/html/html_form_attributes_form.asp

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.