```
<html>

	<head>
		<title>Information Gathered</title>
	</head>

	<body>

<!--

You embed PHP code between tags
echo puts what ever is between quotes in the browser

php code doesn't show if a user tries to view source

A semicolon has to finish every php statement

Single quotes : Print what is between them and ignore
	escape sequences except for \' and \\

Double quotes : Print many escape sequences, the values
	for variables, and more

-->
<?php

/* Multiline
	comment */

// Single line comment

# Another single line comment

echo "<p>Data Processed at </p>";

// Define the time zone based on the coordinated universal time
date_default_timezone_set('UTC');

/* Echos the date
	h : 12 hr format
	H : 24 hr format
	i : Minutes
	s : Seconds
	u : Microseconds
	a : Lowercase am or pm
	l : Full text for the day
	F : Full text for the month
	j : Day of the month
	S : Suffix for the day st, nd, rd, etc.
	Y : 4 digit year
*/
echo date('h:i:s:u a, l F jS Y e');
echo "</p>";

/*
	You store values in variables that have a name
	that starts with a $

	Variables can be of any length and contain letters,
	numbers, or underscores

	They can't begin with a digit adnd are case sensitive.
	numOfCats is not equal to numofcats

	A variable is created and given a data type when it
	receives a value. That data type can change based on
	if the data is changed.

	a. Integer : Whole Numbers
	b. Float : Decimal Numbers
	c. String : Strings or characters
	d. Boolean : true or false
	e. Array : Multiple Items
```

COMMENTS IN HTML

COMMENTS IN PHP

questo è importante!

a questo punto puoi tranquillamente usare questi

```
        f. Object : A Object defined by a class

        A variable by default gets the value NULL

        You can access values from the html that called
        this php script to execute by putting the name
        assigned in the html in single quotes

        The data is stored in an array which is named
        $_POST
*/

$usersName = $_POST['username'];
$streetAddress = $_POST['streetaddress'];
$cityAddress = $_POST['cityaddress'];

echo '<p>Your Information</p>';

// You can combine variables with text using a .

echo $usersName. ' lives at </br>';
echo $streetAddress. ' in </br>';
echo $cityAddress. '</br></br>';

/*
        You can define text using heredoc syntax in the
        same way you use double quotes.
        Starts with <<< and an identifier that can't be
        used any place else in the text.
        It ends with the identifier and a semicolon
        without any white space or anything else.
*/
$str = <<<EOD
The customers name is
$usersName and they
live at $streetAddress
in $cityAddress</br></br>
EOD;

echo $str;

/*
        You can define constants thats value can't change
        When we call for a constant we don't use a $ and
        they are normally uppercase
*/
define('PI', 3.1415926);

echo "The value of PI is " . PI;

// Arithmetic operators
echo "</br></br>5 + 2 = " . (5 + 2);
echo "</br>5 - 2 = " . (5 - 2);
echo "</br>5 * 2 = " . (5 * 2);

// You can cast from 1 type to another like this
echo "</br>5 / 2 = " . (integer) (5 / 2);
echo "</br>5 % 2 = " . (5 % 2) . "</br></br>";

// Use this shortcut when performing an operation using
// the same variable +=, -=, *=, /=, %=, .=

$randNum = 5;
echo $randNum += 5;

echo "</br></br>";

// You can increment and decrement with this shortcut

echo "++randNum = " . ++$randNum . "</br>";
echo "randNum++ = " . $randNum++;
```

I nomi delle variabili cominciano con $ seguito da una lettera

accedo alla componente 'username' dell' array $_POST

strings concatenation

break (per andare a capo)

permette di definire multiple lines strings

NO white space here!

$randNum=5;
echo ++$rand; → 6
echo $rand++; → 6
echo $rand; → 7

modulo

prima incrementa e poi prosegue

prima stampa $randNum e poi lo incrementa

```php
echo "</br></br>";

/*
	The reference operator (ampersan / &) can create a
	reference to a variable so if one changes so does the
	other
*/
$refToNum = &$randNum;
$randNum = 100;
echo '$refToNum = ' . $refToNum;

echo "</br></br>";

// Comparison Operators : ==, !=, <, >, <=, >=
// === (Equal & Same Type), !== (Not Equal or Same Type)
// An if block will perform one action or another depending
// on conditions
if(5 == 10){

	echo '5 = 10';

} else {

	echo '5 != 10';

}

echo "</br></br>";

/*
	elseif is used when you have more conditions to check
*/
$numOfOranges = 4;
$numOfBananas = 36;

if(($numOfOranges > 25) && ($numOfBananas > 30)){

	echo '25% Discount';

} elseif (($numOfOranges > 30) || ($numOfBananas > 35)){

	echo '15% Discount';

} elseif (!(($numOfOranges < 5)) || (!($numOfBananas < 5))){

	echo '5% Discount';

} else {

	echo 'No Discount For You';

}

echo "</br></br>";

// The ternary operator assigns one or another value
// depending on the condition
// condition ? value if true : value if false

$biggestNum = (15 > 10) ? 15 : 10;

echo 'Biggest Number is ' . $biggestNum;

echo "</br></br>";

// Switch provides different actions depending upon values
switch($usersName) {

	case "Derek" :
		echo "Hello Derek";
		break;
```

Nota le single quotes adesso

curly bracket

usa switch se esiste un # limitato di possibilità

se $usersName è Derek stampa "Hello Derek" e esci dagli switch

```
        case "Sally" :
                echo "Hello Sally";
                break;

        default :
                echo "Hello Valued Customer";
                break;
}

echo "</br></br>";

// The while loop performs actions until a condition is met

$num = 0;

while($num < 20){

        echo ++$num . ', ';

}

echo "</br></br>";

// The for loop performs actions until a condition is met
// like the while, but it a compact way

for($num = 1; $num <= 20; $num++){

        echo $num;

        if($num != 20){
                echo ', ';
        } else {
                break; // or exit() to leave the whole script
        }

}

echo "</br></br>";

// An array can store multiple values

$bestFriends = array('Joy', 'Willow', 'Ivy');

// You can access an item by index starting with 0

echo 'My wife ' . $bestFriends[0];

echo "</br></br>";

// You can add an item by storing in a unused index

$bestFriends[4] = 'Steve';

echo 'My friend ' . $bestFriends[4];

echo "</br></br>";

// You could cycle through the array with for or foreach

foreach($bestFriends as $friend){

        echo $friend . ', ';

}

echo "</br></br>";

// You can create key value pairs in arrays

$customer = array('Name'=>$usersName, 'Street'=>$streetAddress, 'City'=>
```

Molto simile al C come sintassi

→ ottimo: NOIV va in seg fault

questo è l'equivalente del python for friend in bestFriends

↳ questa è una mappa chiave-valore

```
$cityAddress);

foreach($customer as $key => $value){

        echo $key . ' : ' . $value . '</br>';

}

echo "</br></br>";

// You can combine arrays with +

$bestFriends = $bestFriends + $customer;

foreach($bestFriends as $friend){

        echo $friend . ', ';

}

// Other common array operators
// == : Returns true of false if arrays are equal
// != : Returns if not equal
// === : Returns if the same items, same order and data type

echo "</br></br>";

// Multidimensional arrays are arrays in arrays

$customers = array(array('Derek', '123 Main', '15212'),
                                  array('Sue', '124 Main', '15222'),
                                  array('Bob', '125 Main', '15212'));

for($row = 0; $row < 3; $row++){

        for($col = 0; $col < 3; $col++){

                echo $customers[$row][$col] . ', ';

        }
        echo '</br>';

}

// Common Array Functions
// sort($yourArray) : Sorts in ascending alphabetical order or
// if you add , SORT_NUMERIC or , SORT_STRING
// asort($yourArray) : sorts arrays with keys
// ksort($yourArray) : sorts by the key
// Put a r infront of the above to sort in reverse order

echo "</br></br>";

// Strings store a series of characters

$randString = "         Random String       ";

// You can trim white space with ltrim, rtrim, or trim

echo strlen($randString) . "</br>";
echo strlen(ltrim($randString)) . "</br>";
echo strlen(rtrim($randString)) . "</br>";
echo strlen(trim($randString)) . "</br>";

echo "</br></br>";

// printf allows you to print formatted Strings to the screen

echo "The randomString is $randString </br>";

printf("The randomString is %s </br>", $randString);
```

questo è l' array sul quale vuoi lavorare

memo: questo concatena

N.B: gli array sono concatenati, ma hanno chiavi diverse

questo taglia gli spazi bianchi a sinistra

questo leva lo spazio bianco sia a sinistra sia a destra

stringa

essi, vedi, proprio come il C

```php
// Coversion codes are useful with decimals

$decimalNum = 2.3456;

printf ("decimal num = %.2f </br>", $decimalNum);

// Other conversion codes
// b : integer to binary
// c : integer to character
// d : integer to decimal
// f : double to float
// o : integer to octal
// s : string to string
// x : integer to hexadecimal

printf ("10 to binary %b </br>", 10);

echo "</br></br>";

// String case functions

echo strtoupper($randString) . "</br>";
echo strtolower($randString) . "</br>";
echo ucfirst($randString) . "</br>";

echo "</br></br>";

// Turning strings into arrays and vice versa

$arrayForString = explode(' ', $randString, 2);

$stringToArray = implode(' ', $arrayForString);

echo "</br></br>";

// Get part of a string

$partOfString = substr("Random String", 0, 6);

echo "Part of String $partOfString </br>";

echo "</br></br>";

// Comparing Strings

$man = "Man";
$manhole = "Manhole";

// Returns 0 is equal
// Returns positive if str1 is greater then str2
// Returns negative if str1 is less than then str2
// strcasecmp() isn't case sensitive

echo strcmp($man, $manhole) . "</br>";

echo "</br></br>";

// strstr() returns every character after the sting to look for
// stristr() isn't case sensitive

echo "The String " . strstr($randString, "String") . "</br>";

echo "</br></br>";

// strpos() returns the location for the match

echo "Loc of String " . strpos($randString, "String") . "</br>";

echo "</br></br>";

// str_replace() replaces a string with another
```

voglio 2 cifre decimali

tutto andrà in upper case

solo la prima lettera andrà in upper case

cosa dividerà la stringa

quanto sarà lungo l'array

dal carattere 0 avanti di 6

questo dà -4

$randString = "Random String"

stampa "7"

```
$newString = str_replace("String", "Stuff", $randString)  . "</br>";

echo "New string " . $newString . "</br>";

echo "</br></br>";

// Escaping characters

$dbString = '"Random quotes"';

echo addslashes($dbString) . "</br>";
echo stripslashes($dbString) . "</br>";

echo "</br></br>";

// Get the data type for a variable

echo 'Data Type for $biggestNum is ' . gettype($biggestNum);

echo "</br></br>";

/* You can check for other types of data with
        is_array : is_bool : is_double : is_int : is_null :
        is_numeric : is_string
*/

// empty() returns true or false if a var has a non-zero value

echo 'Does $biggestNum exist ';
echo empty($biggestNum) ? 'false' : 'true';

echo "</br></br>";

// isset() returns true or false if a variable exists

echo 'Does $biggestNum exist ';
echo isset($biggestNum) ? 'true' : 'false';

echo "</br></br>";

// You can execute unix commands by surrounding with `s
echo `ls -la`; // Unix or OSX

// echo `dir /w`; WINDOWS

echo "</br></br>";

/*
        Functions allow you to reuse code
        A function must begin with a letter, but can contain
        numbers and underscores
*/

function addNumbers($num1, $num2){

        return $num1 + $num2;

}

echo "3 + 4 = " . addNumbers(3, 4);
```

?> chiude il PHP

</body>

</html>

sostituisce "String" in "Stuff" in $randString

PHP supporta anche
→ espressioni regolari
→ object oriented programming