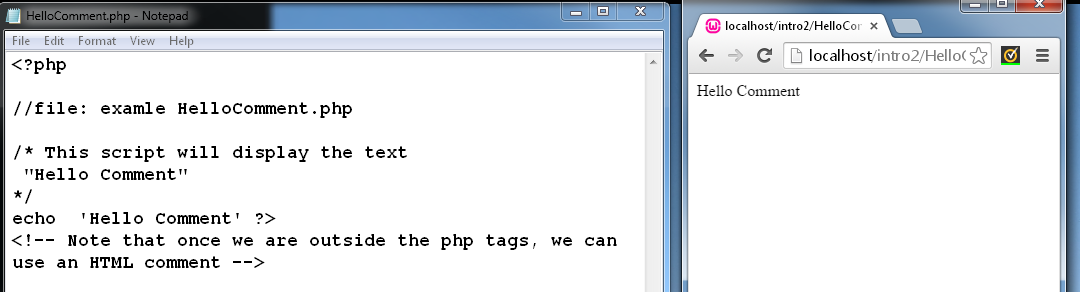
PHP Built-in elements and Structure of the language.

# Commenting your language.

It is good practice to include comments in your code.



# Basic Variables.

There is a set of rules which need to be referred to when you name your variables. A valid variable is preceded by a $ symbol, and it can start with a letter or an underscore character, followed by any number of letters, numbers and underscores. Here are some examples below:

<?php

$var= ‘Elizabeth’;

$\_var= 56;

$Var=’Green’;

?>

The following variables are invalid:

var = ‘Dan’; //invalid – does not start with a $ symbol

$2var = ‘Sam’; //Invalid – name starts with a number.

**Variable Type Naming conventions.**

|  |  |
| --- | --- |
| Type | Variable Name |
| String | **str** |
| Boolean | **Boo** |
| Double | **double** |
| Array | **arr** |
| Object | **obj** |

It is useful to use variable names that indicate the type of data the variable is storing. The table above lists three letter indicators for the start of a variable name, that also indicate the type of data the variable is storing.

**Boolean**: is TRUE or FALSE

$booAnswer =True;

**Integer**: whole numbers from {… -2,-1,0,1,2,3…..}

$intNumber = 4;

$intSecond Number = -6;

**Double**: numbers with a floating decimal point.

$doubleNumber = 1.31

**String**: a series of characters.

<?php

$strName =’Cactus’

Echo $strName;

?>

**Array**: an ordered map or list, which maps values to keys.

**Object**: the object type is an instance of a class

**Resource:** Is a special type of variable that holds a reference(handler) to an external item, such as a database, a text file or an image.

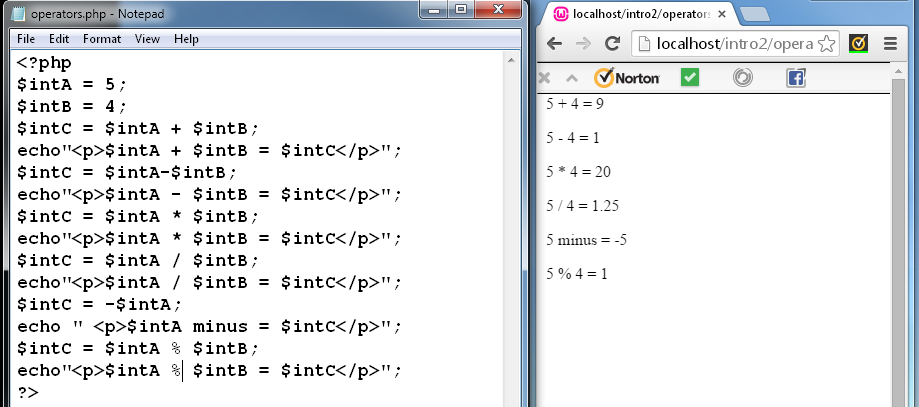
**NULL**: Used to specify a variable that contains no value. It can be considered null if iit has been defined but not assigned any value.

$var = NULL;

**Arithmetic Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Example** | **Result** |
| + | Addition | $a + $b | The sum of $a and $b |
| - | Subtraction | $a - $b | The difference between $a and $b |
| \* | Multiplication | $a \* $b | Product of $a and $b |
| / | Division | $a / $b | $a divided by $b |
| - | Negation | -$a | The negative of $a |
| % | Modulus | $a % $b | Remainder of $a divided by $b |

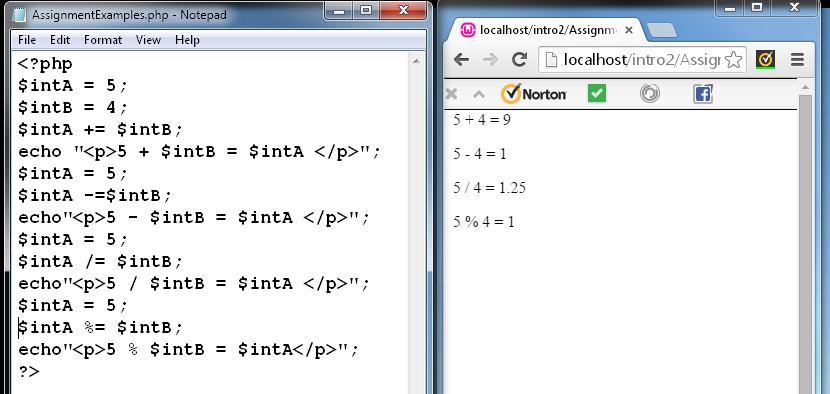
**The following script illustrates the outputs obtained from using the above operators:**



**Assignment Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Name | Example | Result |
| = | Assign | $a = $b | Assign |
| += | Add and Assign | $a += $b | Assign the sum of $a and $b to $a |
| -= | Subtract and Assign | $a -=$b | Assign difference of $a and $b to $a |
| \*= | Multiply and Assign | $a \*= $b | Assign product of $a and $b to $a |
| /= | Divide and Assign | $a /= $b | Assign quotient of $a and $b to $a |
| %= | Get remainder and Assign | Get the remainder and assign. | Assign the remainder of $a divided by $b to $a. |

**Using the Assignment operators above.**



**Comparison Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Example** | **Result** |
| **==** | Equal | $a = =$b | $a is equal to $b |
| **===** | Identical | $a == =$b | $a is equal to $b and they are the same type |
|  |  |  |  |
| != | Not equal | $a != $b | $a is not equal to $b |
| <> | Not equal | $a <> $b | $a is not equal to $b |
| **!==** | Not identical | $a !== $b | $a is not equal to $b |
|  |  |  |  |
| **<** | Less than | $a < $b | $a is less than $b |
| **>** | Greater than | $a > $b | $a is greater than $b |
| **<=** | Less than or equal to | $a <= $b | $a is less than or equal to $b |
| **>=** | Greater than or equal to. | $a >= $b | $a is greater than or equal to $b |
|  |  |  |  |

**Exercises: Complete in a word file and upload to the Assignment button.**

**Where you have picked up syntax errors in questions 5&6, show how you would write the de-bugged code so that it works effectively.**

1. Consider the following variable names. For each one determine if they are valid or invalid.

$var = 5; **Valid**

$\_var = 5; **Valid**

$\_var­ \_= 5; **Invalid, Contains a symbol other than Underscore**

$v3ar = 5; **Valid**

$5var =5; **Invalid, starts with a number**

$\_5var = 5; **Valid**

1. What will be displayed by the following script?

<?php

echo ‘<p>Hello I\’m Samuel </p>’;

?>

**Hello I’m Samuel**

1. What three characters do we suggest you start your variable names with in order to indicate the type of data the variable is storing for each of the following types?

Boolean - **boo**

Integer - **int**

Float - **$var**

String - **str**

Array - **arr**

1. What will be displayed by the following script?

<?php

Define (“NAME”, “Hayley”);

echo “<p>”;

echo ”NAME”;

echo ’NAME’;

echo NAME;

echo ‘</p>’; ?>

**NAMENAMEHayley**

1. What will be displayed by the following script?

<?php

$strName = “Steven”;

echo”<p>”;

echo ‘StrName’;

echo”</p>”;

?>

**StrName**

1. What will be displayed by the following script?

<?php

echo “<p>”

echo 2 + 4 \* 4;

echo “</P>”;

?>

**Nothing at all.**