* Variable
  + **NULL** is\_null()
  + **Scalar** is\_ Scalar()
    - Boolean $x = true; is\_bool()
    - String $x = "RTC"; is\_string()
    - **Number** is\_numeric()
      * Integer $x = 5; is\_int()=is\_Integer()=is\_long()
      * Float=double $x = 5.5; is\_float()=is\_double()=is\_real()

* + **Composed**
    - Array array("Volvo","BMW","Toyota"); is\_array()
    - Object is\_object()
* Constants define(*name*, *value);* define("*name*");
* session

session\_start();  
$\_SESSION["favcolor"] = "green";

Array

Array Type

Indexed Array

Associative Array

Array dimensional

One dimensional Array

Multidimensional Array

Array Methods

- Add Items

- Remove Items

-Edit Items

-Print Items

-Print Array

# Path Control

## Switching Flow

### if

### Switch

### ternary operator

## Loops

### For

### Foreach

### While

### Do

# Array

=========

## Indexed Array

============

$manths=array("jan","feb","mar","apr","may","jun","jul","aug","sep","oct""nov","dec");

$manths[]="jan";

$manths[]="feb";

$manths[]="mar";

$manths[]="apr";

$manths[]="may";

$manths[]="jun";

$manths[]="jul";

Print Array

Print\_r($manths);

Print Array in orderList

echo "<ol>";

foreach($Manths as $Manth){

echo "<li>$Manth </li>";

}

echo "</ol>";

Get Array count

Echo count($manths);

# Switching Flow

=============

## if Statement

=========

if (expression)

one statement

if (expression) {

// first block (one or more statements)

}

if (expression) {

// first block (one or more statements)

}

else{

// second block (one or more statements)

}

if (first expression) {

// first block (one or more statements)

}

elseif(second expression){

// second block (one or more statements)

}

else{

// third block (one or more statements)

}

## switch Statement

=================

switch (expression) {

case result1:

// execute this if expression results in result1

break;

case result2:

// execute this if expression results in result2

break;

default:

// execute this if no break statement

// has been encountered hitherto

}

## ternary operator

===============

(expression) ? returned\_if\_expression\_is\_true : returned\_if\_expression\_is\_false;

# Loops

======

## Loops (for)

=============

for (initialization expression; test expression; modification expression) {

// code to be executed

}

foreach($Manths as $Manth){

// code to be executed

}

foreach($Manths as $Manth=>$val){

// code to be executed

}

## Loops (while)

=============

while (expression) {

// do something

};

## Loops (do)

=============

do {

// code to be executed

} while (expression);

for (initialization expression; test expression;

modification expression) {

// code to be executed

}