PHP

control structures - if statements
indexed arrays

PHP files are processed top to bottom in sequence

```
<html>
<?php ... ?>
<head>
<?php ... ?>
<title>... <?php ... ?> ...</title>
</head>
<body>
<p>
<?php ... ?>
</p>
</body>
</html>
```

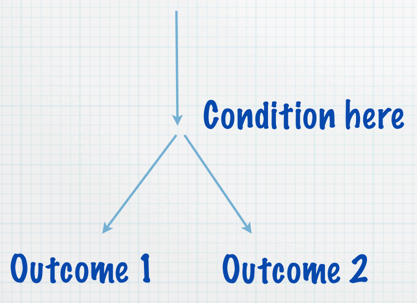
Starting at the top

Working down to the bottom

The control flow

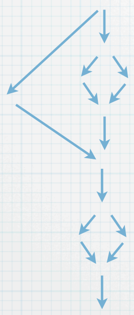
But sometimes we need to have choices / alternatives

Start at the top



Work down to the bottom

Can be complex flows



Done with an if or a switch statement

```
if (expression)
  statement;
```

Perform the statement if the expression is true

```
if (expression)
  statement1
else
  statement2;
```

Perform statement1 if the expression is true otherwise statement2

```
if (expression){
   statement;
   statement;
   }
else {
   statement;
   statement;
   };
```

Perform blocks of statements if the expression is true otherwise ...

```
if (expression){
   statement;
   }
else
  if (expression){
         statement;
         }
     else {
           statement;
           };
```

If there are many choices a nested series of if statement may be required

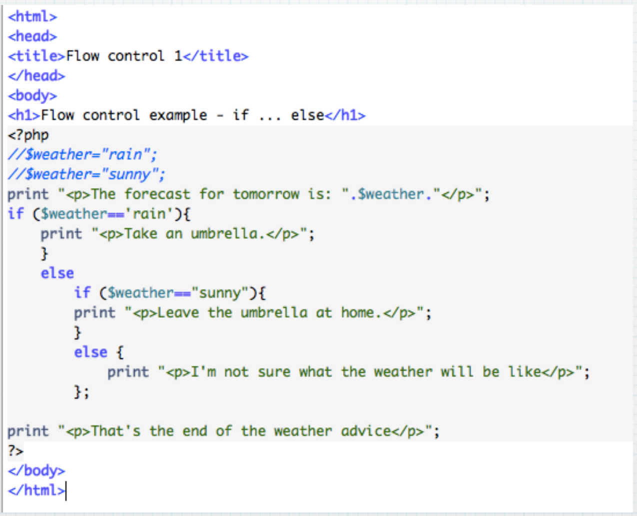
Note how tabs are used to help readability - do the same with your code

```
switch($name) {
      case 'value 1 of name':
        // do something
        break;
      case 'value 2 of name':
        // do something
        break;
      case 'value 3 of name':
        // do something
        break;
      case 'value 4 of name':
        // do something
        break;
    }
```

When the if statement has many sub-if parts, a switch statement may be better

Best to choose this when $name has a limited set of values known in advance

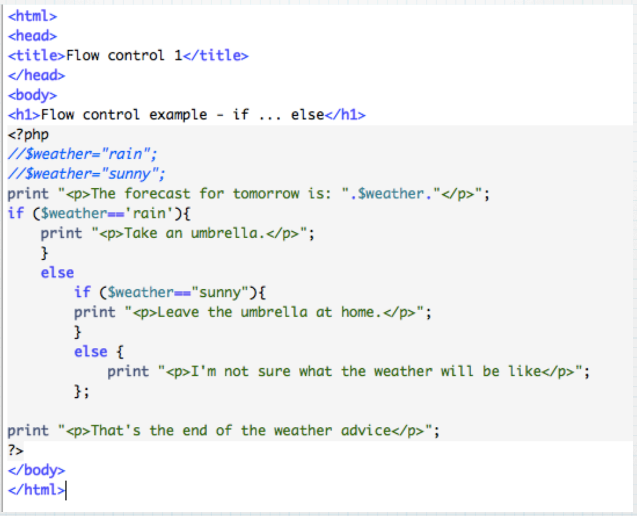
Examples Should I take an umbrella?



```
<html>
<head>
<title>Flow control 1</title>
</head>
<body>
<h1>Flow control example - if ... else</h1>
<?php
//$weather="rain";
//$weather="sunny";
print "<p>The forecast for tomorrow is: ".$weather."</p>";
if ($weather=='rain'){
    print "<p>Take an umbrella.</p>";
    }
    else
        if ($weather=="sunny"){
        print "<p>Leave the umbrella at home.</p>";
        }
        else {
            print "<p>I'm not sure what the weather will be like</p>";
        };

print "<p>That's the end of the weather advice</p>";
?>
</body>
</html>
```

Examples TV form

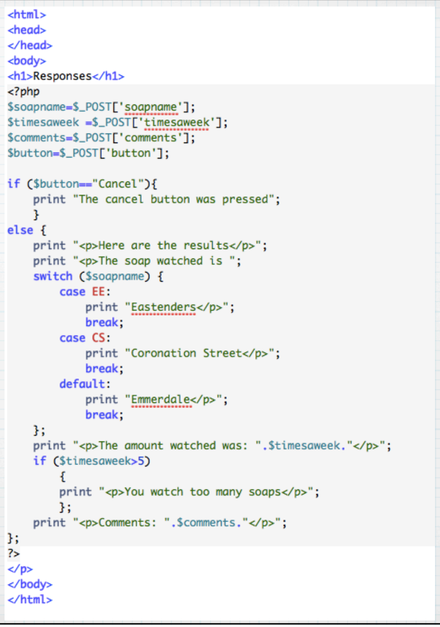


```
<html>
<head>
<title>Flow control 1</title>
</head>
<body>
<h1>Flow control example - if ... else</h1>
<?php
//$weather="rain";
//$weather="sunny";
print "<p>The forecast for tomorrow is: ".$weather."</p>";
if ($weather=='rain'){
    print "<p>Take an umbrella.</p>";
    }
    else
        if ($weather=="sunny"){
        print "<p>Leave the umbrella at home.</p>";
        }
        else {
            print "<p>I'm not sure what the weather will be like</p>";
        };

print "<p>That's the end of the weather advice</p>";
?>
</body>
</html>
```

Examples

TV form response



```
<html>
<head>
</head>
<body>
<h1>Responses</h1>
<?php
$soapname=$_POST['soapname'];
$timesaweek =$_POST['timesaweek'];
$comments=$_POST['comments'];
$button=$_POST['button'];

if ($button=="Cancel"){
    print "The cancel button was pressed";
    }
else {
    print "<p>Here are the results</p>";
    print "<p>The soap watched is ";
    switch ($soapname) {
        case EE:
            print "Eastenders</p>";
            break;
        case CS:
            print "Coronation Street</p>";
            break;
        default:
            print "Emmerdale</p>";
            break;
    };
    print "<p>The amount watched was: ".$timesaweek."</p>";
    if ($timesaweek>5)
        {
        print "<p>You watch too many soaps</p>";
        };
    print "<p>Comments: ".$comments."</p>";
};
?>
</p>
</body>
</html>
```

Arrays

- indexed arrays
- associative arrays

Much of this material is explained in PHP programming 2nd Ed. Chap 5

Arrays

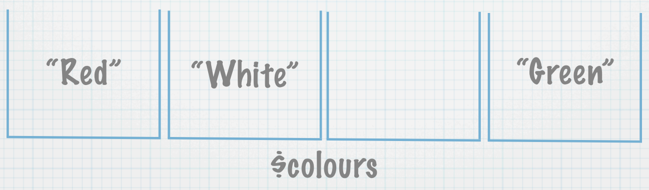
- indexed arrays - today
- associative arrays

Much of this material is explained in PHP programming 2nd Ed. Chap 5

Arrays

Sometimes we have a set of values that are connected

Can use a structure called an array to store these



A series of boxes with the same name

Arrays

So how do we get at the individual values inside the array?

Use a number - the index

Index is indicated in square brackets

indexed arrays

Uses consecutive integers to index the cells

	0	1	2	3
$colours	Red	Green	Blue	Yellow

```
print $colours[1];
```

Green

indexed arrays

Uses consecutive integers to index the cells

	0	1	2	3
$colours	Red	Green	Blue	Yellow

```
$colours[2]="Purple";
```

indexed arrays

Uses consecutive integers to index the cells

	0	1	2	3
$colours	Red	Green	Purple	Yellow

```
$colours[2]="Purple";
```

indexed arrays

Use simple assignment to create the array

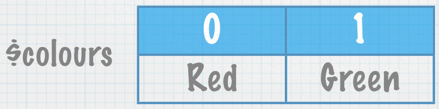
```
$colours[0]="Red";
```

	0
$colours	Red

indexed arrays

Use simple assignment to create the array

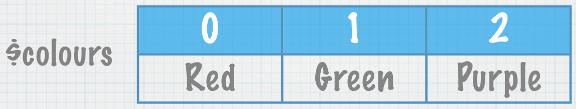
```
$colours[0]="Red";
$colours[1]="Green";
```



indexed arrays

Use simple assignment to create the array

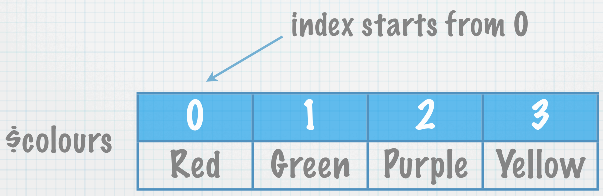
```
$colours[0]="Red";
$colours[1]="Green";
$colours[2]="Purple";
```

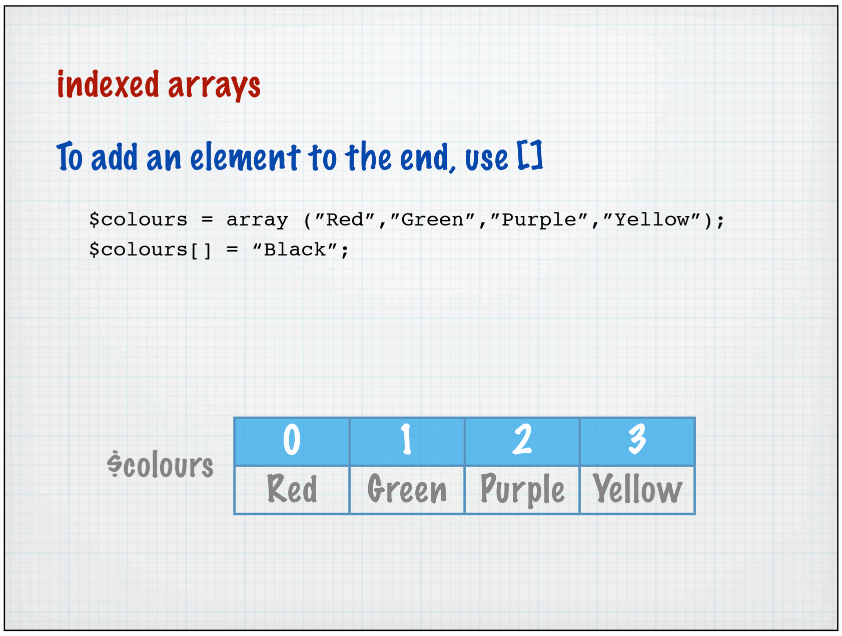


indexed arrays

If all the values are known in advance, use the reserved word array

```
$colours = array ("Red","Green","Purple","Yellow");
```



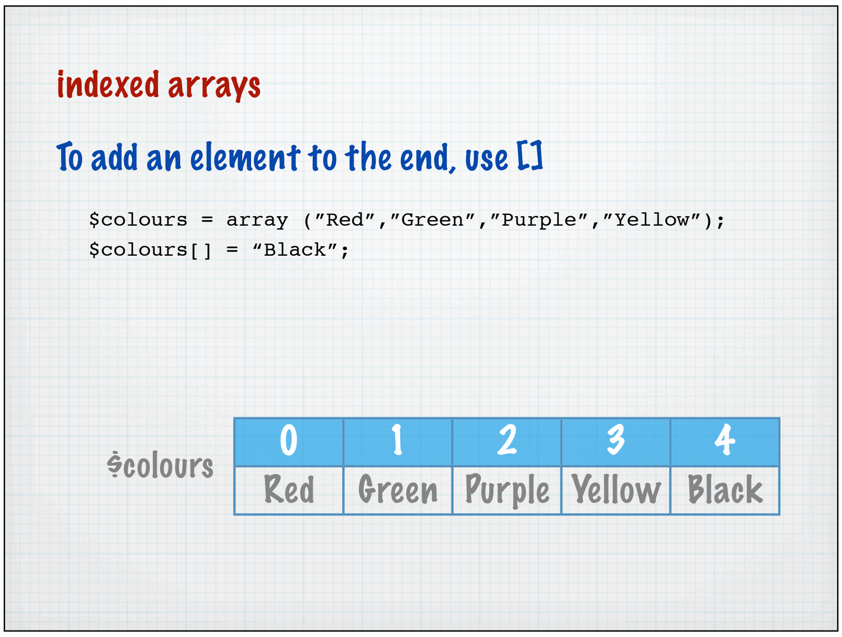


indexed arrays

To add an element to the end, use []

```
$colours = array ("Red","Green","Purple","Yellow");
$colours[] = "Black";
```

$colours	0	1	2	3
	Red	Green	Purple	Yellow

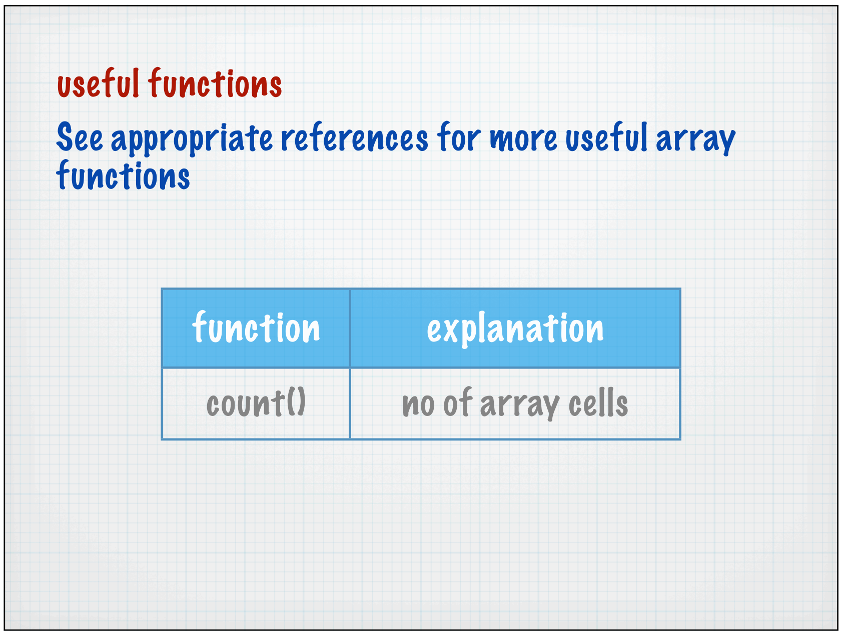


indexed arrays

To add an element to the end, use []

```
$colours = array ("Red","Green","Purple","Yellow");
$colours[] = "Black";
```

$colours	0	1	2	3	4
	Red	Green	Purple	Yellow	Black



useful functions

See appropriate references for more useful array functions

function	explanation
count()	no of array cells