PHP

built-in functions
control structures - loops cont.
associative arrays

functions

Built-in 'pieces' of code that we can use

Large number available - in the manual

http://www.php.net/manual/en/funcref.php

functions

Each function has a unique name

To use the function we say 'you "call" the function'

We pass values into the function

Sometimes called parameters or arguments/args

We get a result back

functions

Some examples

strlen

(PHP 4, PHP 5)

strlen — Get string length

Description Report a bug

int **strlen** (string *$string*)

Returns the length of the given ***string***.

Parameters Report a bug

string

The string being measured for length.

functions

Some examples

strtoupper

(PHP 4, PHP 5)

strtoupper — Make a string uppercase

Description Report a bug

string **strtoupper** (string *$string*)

Returns ***string*** with all alphabetic characters converted to uppercase.

Note that 'alphabetic' is determined by the current locale. For instance, in the default "C" locale characters such as umlaut-a (ä) will not be converted.

Parameters Report a bug

string

The input string.

functions

Some examples

strpos

(PHP 4, PHP 5)

strpos — Find the position of the first occurrence of a substring in a string

Description Report a bug

mixed **strpos** (string *$haystack* , mixed *$needle* [, int *$offset* = *0*])

Find the numeric position of the first occurrence of ***needle*** in the ***haystack*** string.

Parameters Report a bug

haystack

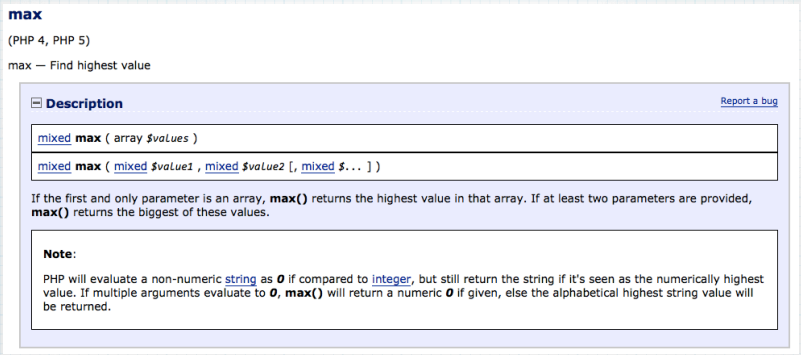
The string to search in.

needle

If ***needle*** is not a string, it is converted to an integer and applied as the ordinal value of a character.

functions

Some examples



max

(PHP 4, PHP 5)

max — Find highest value

Description Report a bug

mixed **max** (array *$values*)

mixed **max** (mixed *$value1* , mixed *$value2* [, mixed *$...*])

If the first and only parameter is an array, **max()** returns the highest value in that array. If at least two parameters are provided, **max()** returns the biggest of these values.

Note:

PHP will evaluate a non-numeric string as ***0*** if compared to integer, but still return the string if it's seen as the numerically highest value. If multiple arguments evaluate to ***0***, **max()** will return a numeric ***0*** if given, else the alphabetical highest string value will be returned.

functions

Exercises - find functions and write 3 small programs to

- Count the number of cells in an array
- Sort an array of numbers and print the results
- Print a random number between 1 and 10

Control structures / flow control

- if ... else
- Switch
- for loops
- while ... do ..
- do ... while ...

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

Control structures / flow control

- if ... else
- Switch
- for loops
- while ... do ..
- do ... while ...

} Done these

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

Loops / Iteration / doing things over and over and over and over ....

Three standard loop types

for
while ...
do ... while

Loops / Iteration / doing things over and over and over and over ....

Three standard loop types

for
while ...
do ... while

Choose a for-loop if the number of times the loop will run is known 'in advance'

The loop will run 4 times
The loop will run 1000 times
The loop will run 'n' times

Loops / Iteration / doing things over and over and over and over ....

Three standard loop types

- for — ...or you are processing an array
- while ...
- do ... while

Loops / Iteration / doing things over and over and over and over ....

Three standard loop types

- for
- while ... — Choose a while-loop if the loop will run 0 or more times till some condition becomes false
- do ... while

Loops / Iteration / doing things over and over and over and over ....

Three standard loop types

- for
- while ...
- do ... while — Choose a do-while-loop if the loop will run 1 or more times till some condition becomes false

Loops / Iteration / doing things over and over and over and over ....

Three standard loop types

for
while ...
do ... while

With these we typically don't know in advance how many times the loop will execute...

Loops / Iteration / doing things over and over and over and over ....

Three standard loop types

for
while ...
do ... while

..but we must know the minimum number of times the loop will work to choose between while or do ... while

i.e. 0 or 1

while loop

The structure of a while statement is:

```
while (condition)
   statement
```

Loop continues whilst condition is true

or with many statements -

```
while (condition){
   statement;
   statement;
   statement;
   statement;
};
```

Do something in here to change the condition (unless you want it to continue ∞)

while loop

```
$today="Monday";

while($today<>"Friday"){
print "<p>Today is ".$today."</p>";
if (rand(1,10)>7){
    $today="Friday";
    };
};

print "Final value of today is ".$today;
```

If the condition is initially false, the loop doesn't execute at all (i.e. 0 times)

while loop

```
$today="Friday";

while($today<>"Friday"){
print "<p>Today is ".$today."</p>";
if (rand(1,10)>7){
    $today="Friday";
    };
};

print "Final value of today is ".$today;
```

If the condition is initially false, the loop doesn't execute at all (i.e. 0 times)

do .. while loop

The structure of a do .. while statement is:

```
do
    statement;
while (condition)
```

Will loop at least once

or with many statements -

```
do {
    statement;
    statement;
    statement;
    statement;
 }
 while (condition);
```

Do something in here to change the condition (unless you want it to continue ∞)

do .. while loop

```
do {
$i=rand(1,10);
print "<p>This time i had the value $i</p>";
} while ($i<8);
```

As condition is at the bottom of the structure the loop must execute at least once

arrays

- indexed arrays
- associative arrays
- useful functions
- iterating over an array

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

arrays

- indexed arrays
- associative arrays
- useful functions
- iterating over an array

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

indexed arrays

Uses consecutive integers to index the cells

$colours

0	1	2	3
Red	Green	Blue	Yellow

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

Green

indexed arrays

Uses consecutive integers to index the cells

$colours

0	1	2	3
Red	Green	Blue	Yellow

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

indexed arrays

Uses consecutive integers to index the cells

$colours

0	1	2	3
Red	Green	Purple	Yellow

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

indexed arrays

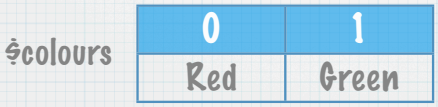
Use simple assignment to create the array

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

indexed arrays

Use simple assignment to create the array

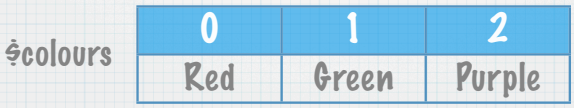
```
$colours[0]="Red";
$colours[1]="Green";
```



indexed arrays

Use simple assignment to create the array

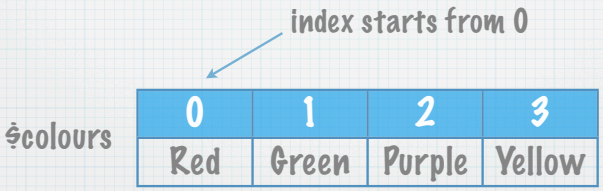
```
$colours[0]="Red";
$colours[1]="Green";
$colours[2]="Purple";
```



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";
```

indexed arrays

To add an element to the end, use []

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

associative arrays

Uses labels to index the cells

$favTV

0	1	2	3
West Wing	Mad men	Horizon	Desperate Housewives

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV['Barry'];
```

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV['Barry'];
```

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV['Barry'];
```

Mad men

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV["Dan"];
```

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV[“Dan”];
```

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV[“Dan”];
```

West Wing

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
$favTV[‘Walter’]=”ER”
```

associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	ER

```
$favTV['Walter']="ER"
```

associative arrays

Use simple assignment to create the array

```
$favTV["Dan"]="West Wing";
```

$favTV

Dan
West Wing

associative arrays

Use simple assignment to create the array

```
$favTV["Dan"]="West Wing";
$favTV["Barry"]="Mad men";
```

$favTV

Dan	Barry
West Wing	Mad men

associative arrays

Use simple assignment to create the array

```
$favTV["Dan"]="West Wing";
$favTV["Barry"]="Mad men";
$favTV["Phil"]="Horizon";
```

$favTV

Dan	Barry	Phil
West Wing	Mad men	Horizon

associative arrays

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

```
$favTV = array (
        "Dan"=>"West Wing",
        "Barry"=>"Mad men",
        "Phil"=>"Horizon"
        );
```

$favTV

Dan	Barry	Phil
West Wing	Mad men	Horizon

useful functions

See appropriate references for full definitions of these

functions	explanation
count()	no of array cells
array_pad()	create an array with the same value
array_slice()	extract part of an array
array_keys()	returns an array of the keys
array_key_exists()	see if a key exists

Processing an indexed array

Use a for loop to iterate over the cells

$colours

0	1	2	3
Red	Green	Blue	Yellow

```
for ($i=0; $i<count($colours); $i++){
  print $colours[$i];
  );
```

Processing an associative array

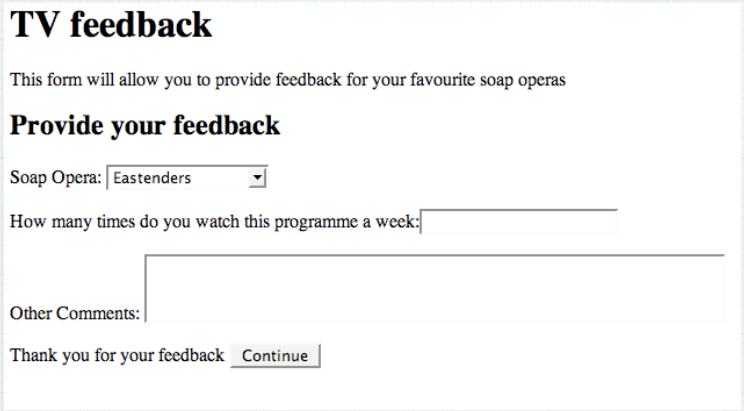
Use a foreach loop to iterate over the cells

$favTV

Dan	Barry	Phil
West Wing	Mad men	Horizon

```
foreach ($favTV as $person=>$programme){
  print $person.” likes “.$programme;
  );
```

We have seen associative arrays before -



Web forms

Two ways of passing information between pages

get	form information is passed through the URL
post	form information is embedded in the HTTP stream

Web forms

```
<form name="form1" action="responseget.php" method="get">
<p>Soap Opera:
    <select name="soapname">
      <option value="EE">Eastenders</option>
      <option value="CS">Coronation Street</option>
      <option value="EMD">Emmerdale</option>
    </select>
</p>
<p>How many times do you watch this programme a week:<input type="text" name="timesaweek"></p>

<p>Other Comments: <textarea name="comments" cols="60" rows="3"></textarea></p>
<p>Thank you for your feedback <input type="submit" name="continue" value="Continue" /></p>
</form>
```

$_GET associative array

PHP automatically creates an associative array with the passed URL values

```
responseget.php?soapname=EE&timesaweek=3&comments=Its
+depressing&continue=Continue
```

$_GET

soapname	timesaweek	comments	continue
EE	3	its depressing	Continue

Web forms

```
<form name="form1" action="responsepost.php" method="post">
<p>Soap Opera:
    <select name="soapname">
      <option value="EE">Eastenders</option>
      <option value="CS">Coronation Street</option>
      <option value="EMD">Emmerdale</option>
    </select>
</p>
<p>How many times do you watch this programme a week:<input type="text" name="timesaweek"></p>

<p>Other Comments: <textarea name="comments" cols="60" rows="3"></textarea></p>
<p>Thank you for your feedback <input type="submit" name="continue" value="Continue" /></p>
</form>
```

Small change here

$_GET associative array

PHP automatically creates an associative array with the passed URL values from the HTTP stream

$_POST

soapname	timesaweek	comments	continue
EE	3	its depressing	Continue