

PHP Functions / Modularity

M. Bilal Naeem

www.invictussolutions.co

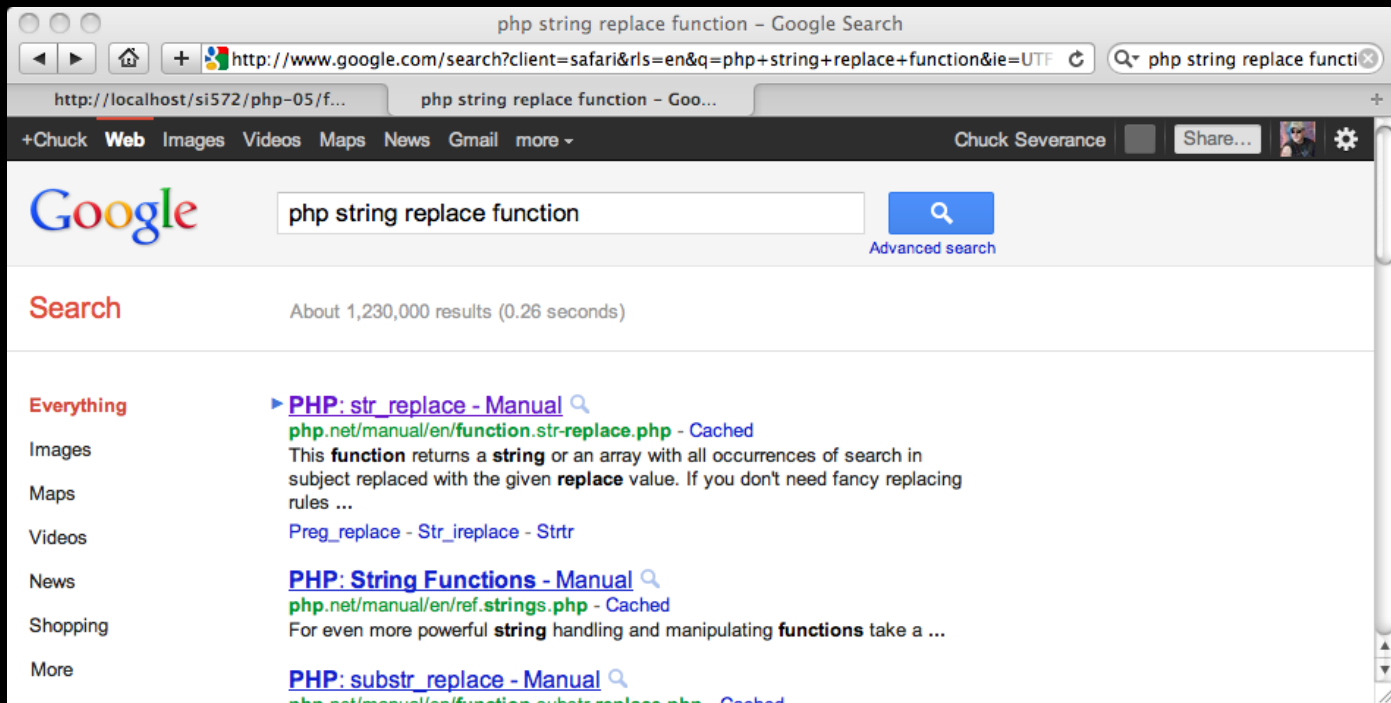
Why Functions?

- PHP has lots of built-in functions that we use all the time.
- We write our own functions when our code reaches a certain level of complexity.

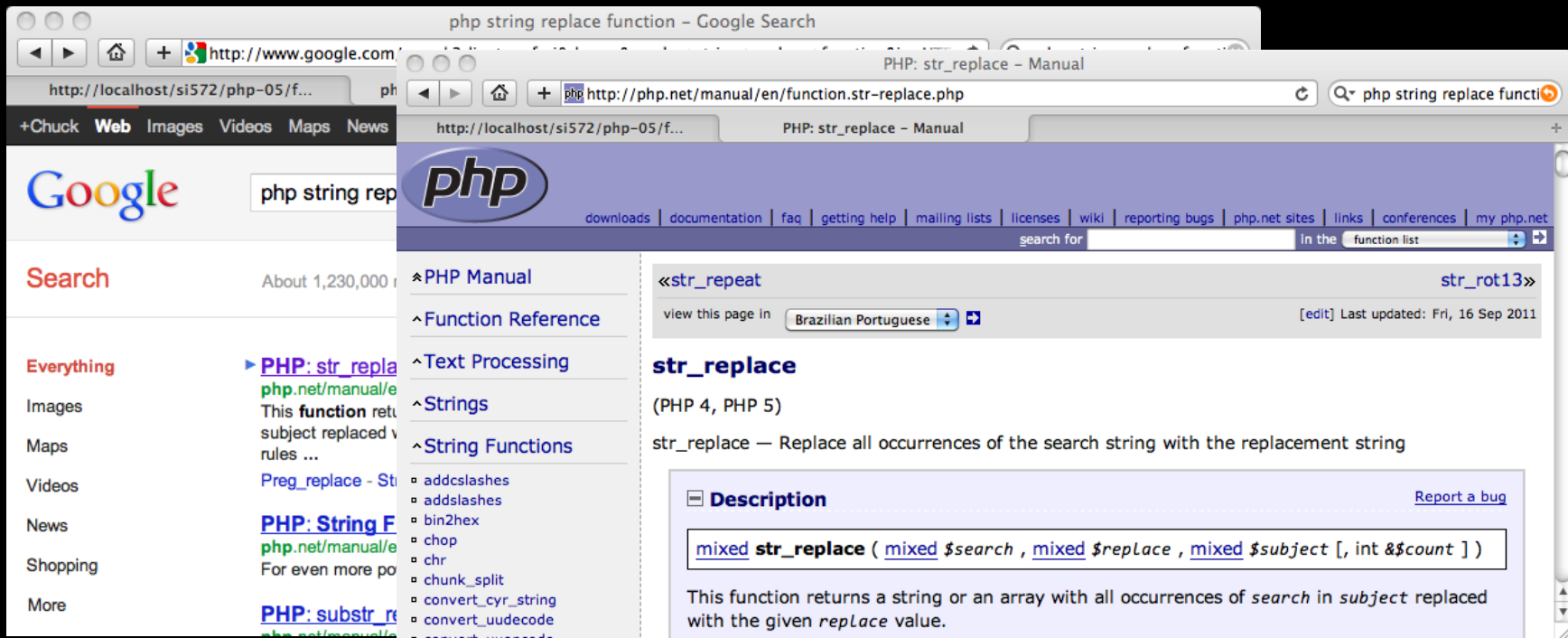
To function or not to function...

- Organize your code into “paragraphs” - capture a complete thought and “name it”.
- Don’ t repeat yourself - make it work once and then reuse it.
- If something gets too long or complex, break up logical chunks and put those chunks in functions.
- Make a library of common stuff that you do over and over - perhaps share this with your friends...

PHP Documentation - Google



PHP Documentation - Google



Built-In Functions...

- Much of the power of PHP comes from its built-in **functions**.
- Many are modeled after C string library functions (i.e. `strlen()`).

```
echo strrev(" .dlrow olleH"); echo str_repeat("Hip ",  
2); echo strtoupper("hooray!");  
echo strlen("intro");  
echo "\n";
```

```
Hello world.  
Hip Hip  
HOORAY!  
5
```

Defining Your Own Functions

We use the `function` keyword to define a function, we `name the function` and take `optional argument variables`. The body of the function is in a block of code `{ }`

```
function greet() {  
    print "Hello\n";  
}
```

Hello
Hello

```
greet();  
greet();
```

Choosing Function Names

- Much like variable names - but do not start with a dollar sign
- Start with a letter or underscore - consist of letters, numbers, and underscores (_)
- Avoid built-in function names

Return Values

Often a function will take its arguments, do some computation, and return a value to be used as the value of the function call in the calling expression. The `return` keyword is used for this.

```
function greeting() {  
    return "Hello";  
}
```

```
print greeting() . " Glenn\n";  
print greeting() . " Sally\n";
```

Hello Glenn
Hello Sally

Arguments

Functions can choose to accept optional **arguments**. Within the function definition the variable names are effectively “aliases” to the **values passed in** when the function is called.

```
function howdy($lang) {  
    if ( $lang == 'es' ) return "Hola";  
    if ( $lang == 'fr' ) return "Bonjour";  
    return "Hello";  
}
```

```
print howdy('es') . " Glenn\n";  
print howdy('fr') . " Sally\n";
```

Hola Glenn
Bonjour Sally

Optional Arguments

Arguments can have defaults, and so can be omitted.

```
function howdy($lang='es') {  
    if ( $lang == 'es' ) return "Hola";  
    if ( $lang == 'fr' ) return "Bonjour";  
    return "Hello";  
}
```

```
print howdy() . " Glenn\n";  
print howdy('fr') . " Sally\n";
```

Hola Glenn
Bonjour Sally

Call By Value

- The argument variable within the function is an “alias” to the actual variable.
- But even further, the alias is to a *copy* of the actual variable in the function call.

```
function double($alias) {  
    $alias = $alias * 2;  
    return $alias;  
}
```

```
$val = 10;  
$dval = double($val);  
echo "Value = $val Doubled = $dval\n";
```

Value = 10 Doubled = 20

Call By Reference

Sometimes we want a function to change one of its arguments, so we indicate that an argument is “by reference” using (&).

```
function triple(&$realthing) {  
    $realthing = $realthing * 3;  
}
```

```
$val = 10;  
triple($val);  
echo "Triple = $val\n";
```

Triple = 30

sort

Change language: English

[Edit](#) [Report a Bug](#)

(PHP 4, PHP 5, PHP 7)

sort — Sort an array

Description



```
bool sort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

This function sorts an array. Elements will be arranged from lowest to highest when this function has completed.

Parameters

<http://php.net/manual/en/function.sort.php>

Scope and Modularity

Variable Scope

- In general, variable names used inside of function code do not mix with the variables outside of the function to avoid unexpected side effects if two programmers use the same variable name in different parts of the code.
- We call this "name spacing" the variables. The function variables are in one "name space" whilst the main variables are in another "name space".

<http://php.net/manual/en/language.variables.scope.php>

Normal Scope (isolated)

```
function tryzap() {  
    $val = 100;  
}
```

TryZap = 10

```
$val = 10;  
tryzap();  
echo "TryZap = $val\n";
```

Except for \$_GET

<http://php.net/manual/en/language.variables.superglobals.php>

Global Scope (shared)

```
function dozap() {  
    global $val;  
    $val = 100;  
}
```

```
$val = 10;  
dozap();  
echo "DoZap = $val\n";
```

DoZap = 100

Use this wisely, young
Jedi...

Global Variables – Use Rarely

- Passing variable in as parameter
- Passing value back as return value
- Passing variable in by reference
- If you use global variables, use long names with nice unique prefixes

```
global $LastOAuthBodyBaseString;  
global $LAST_OAUTH_BODY_BASE_STRING;
```

Coping with Missing Bits

Sometimes, depending on the version or configuration of a particular PHP instance, some functions may be missing. We can check that...

```
if (function_exists("array_combine")){  
    echo "Function exists";  
} else {  
    echo "Function does not exist";  
}
```

This allows for evolution.

array_key_exists

Change language: English

[Edit](#) [Report a Bug](#)

(PHP 4 >= 4.0.7, PHP 5, PHP 7)

array_key_exists — Checks if the given key or index exists in the array

Description

```
bool array_key_exists ( mixed $key , array $array )
```

`array_key_exists()` returns **TRUE** if the given **key** is set in the array. **key** can be any value possible for an array index.

Parameters

<http://php.net/manual/en/function.array-key-exists.php>

One Heck of a Function...

- PHP is a very configurable system and has lots of capabilities that can be plugged in.
- The `phpinfo()` function prints out the internal configuration capabilities of your particular PHP installation,

```
<?php  
    phpinfo();  
?>
```

MAMP

localhost:8888/MAMP/index.php?language=English&page=phpinfo

PHP Version 5.5.10

System	Darwin 0587369276.wireless.umich.net 13.4.0 Darwin Kernel Version 13.4.0: Sun Aug 17 19:50:11 PDT 2014; root:xnu-2422.115.4~1/RELEASE_ARM64_T8020
Build Date	Mar 7 2014 08:13:38
Configure Command	./configure '--with-mysql=/Applications/MAMP/Library/' '--with-apxs2=/Applications/MAMP/Library/bin/apxs' '--with-gd' '--with-jpeg-dir=/Applications/MAMP/Library/' '--with-png-dir=/Applications/MAMP/Library/' '--with-zlib' '--with-zlib-dir=/Applications/MAMP/Library/' '--with-freetype-dir=/Applications/MAMP/Library/' '--prefix=/Applications/MAMP/bin/php/php5.5.10/' '--exec-prefix=/Applications/MAMP/bin/php/php5.5.10/' '--sysconfdir=/Applications/MAMP/bin/php/php5.5.10/conf' '--with-config-file-path=/Applications/MAMP/bin/php/php5.5.10/conf' '--enable-ftp' '--enable-gd-native-ttf' '--with-bz2=/usr' '--with-ldap' '--with-mysql=/Applications/MAMP/Library/bin/mysql_config' '--with-tlib=/Applications/MAMP/Library/' '--enable-mbstring=all' '--with-curl=/Applications/MAMP/Library/' '--enable-sockets' '--enable-bcmath' '--with-imap-shared=/Applications/MAMP/Library/lib/imap-2007f' '--enable-soap' '--with-kerberos' '--enable-calendar' '--with-pgsql-shared=/Applications/MAMP/Library/pg' '--enable-xml' '--with-libxml-dir=/Applications/MAMP/Library/' '--with-gettext-shared=/Applications/MAMP/Library/' '--with-xsl=/Applications/MAMP/Library/' '--with-pdo-mysql-shared=/Applications/MAMP/Library/' '--with-pdo-pgsql-shared=/Applications/MAMP/Library/pg' '--with-mcrypt-shared=/Applications/MAMP/Library/' '--with-openssl' '--enable-zip' '--with-iconv=/Applications/MAMP/Library/' '--enable-opcache' '--enable-intl' '--with-tdy-shared' '--with-icu-dir=/Applications/MAMP/Library'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/Applications/MAMP/bin/php/php5.5.10/conf
Loaded Configuration File	/Applications/MAMP/bin/php/php5.5.10/conf/php.ini
Scan this dir for additional .ini files	(none)

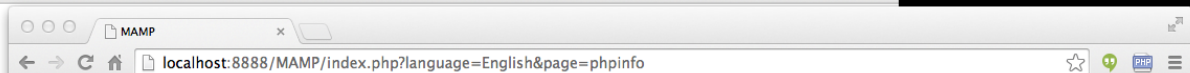
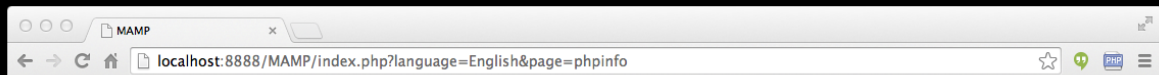
localhost:8888/MAMP/index.php?language=English&page=phpinfo

Configuration

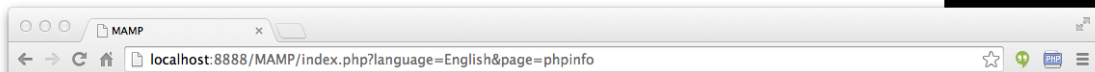
apache2handler

Apache Version	Apache/2.2.25 (Unix) mod_fastcgi/2.4.6 mod_wsgi/3.4 Python/2.7.5 PHP/5.5.10 mod_ssl/2.2.25 OpenSSL/0.9.8za DAV/2 mod_perl/2.0.8 Perl/v5.18.0	
Apache API Version	20051115	
Server Administrator	you@example.com	
Hostname:Port	localhost:8888	
User/Group	csev(505)/-1	
Max Requests	Per Child: 10000 - Keep Alive: on - Max Per Connection: 100	
Timeouts	Connection: 300 - Keep-Alive: 5	
Virtual Server	No	
Server Root	/Applications/MAMP/Library	
Loaded Modules	core prefork http_core mod_so mod_authn_file mod_authn_dbm mod_authn_anon mod_authn_dbd mod_authn_default mod_authz_host mod_authz_groupfile mod_authz_user mod_authz_dbm mod_authz_owner mod_authz_default mod_auth_basic mod_auth_digest mod_file_cache mod_cache mod_disk_cache mod_mem_cache mod_dbd mod_bucketeer mod_dumpio mod_echo mod_example mod_case_filter mod_case_filter_in mod_reqtimeout mod_ext_filter mod_include mod_filter mod_substitute mod_charset_lite mod_deflate mod_log_config mod_logio mod_env mod_mime_magic mod_cern_meta mod_expires mod_headers mod_ident mod_usertrack mod_setenvif mod_version mod_proxy mod_proxy_connect mod_proxy_ftp mod_proxy_http mod_proxy_scgi mod_proxy_apc mod_proxy_balancer mod_ssl mod_mime mod_dav mod_status mod_autoindex mod_asis mod_info mod_cgi mod_fastcgi mod_cgid mod_dav_fs mod_vhost_alias mod_negotiation mod_dir mod_imagemap mod_actions mod_speling mod_userdir mod_alias mod_rewrite mod_perl mod_wsgi mod_php5	

Directive	Local Value	Master Value
engine	1	1
last_modified	0	0
xbithack	0	0



Configuration



gettext

GetText Support	enabled
-----------------	---------

hash

hash support	enabled
Hashing Engines	md2 md4 md5 sha1 sha224 sha256 sha384 sha512 ripemd128 ripemd160 ripemd256 ripemd320 whirlpool tiger128,3 tiger160,3 tiger192,3 tiger128,4 tiger160,4 tiger192,4 snfru snfru256 gost adler32 crc32 crc32b fnv132 fnv164 joaat haval128,3 haval160,3 haval192,3 haval224,3 haval256,3 haval128,4 haval160,4 haval192,4 haval224,4 haval256,4 haval128,5 haval160,5 haval192,5 haval224,5 haval256,5

iconv

iconv support	enabled
iconv implementation	libiconv
iconv library version	1.14

Directive	Local Value	Master Value
iconv.input_encoding	ISO-8859-1	ISO-8859-1
iconv.internal_encoding	ISO-8859-1	ISO-8859-1
iconv.output_encoding	ISO-8859-1	ISO-8859-1

imap

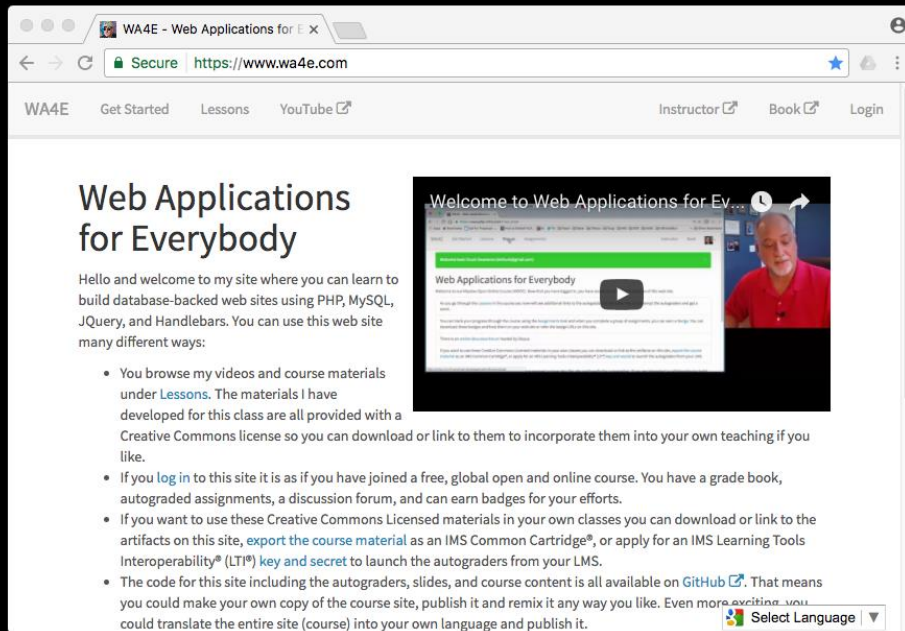
IMAP c-Client Version	2007f
Kerberos Support	enabled

Programming in Multiple Files

Including Files in PHP

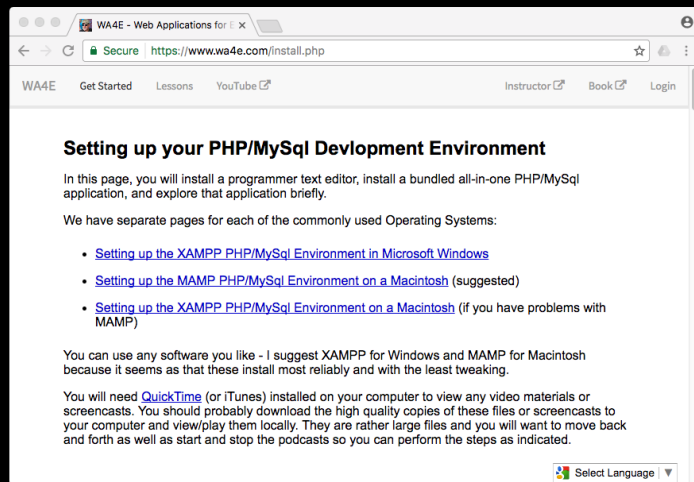
- `include "header.php";` - Pull the file in here
- `include_once "header.php";` - Pull the file in here unless it has already been pulled in before
- `require "header.php";` - Pull in the file here and die if it is missing
- `require_once "header.php";` - You can guess what this means...
- These can look like functions - `require_once("header.php");`

```
<?php
require "top.php";
require "nav.php";
?>
<div id="container">
<h1>Web Applications...</h1>
.
.
.
</div>
<?php
require "foot.php";
```



index.php

```
<?php
require "top.php";
require "nav.php";
?>
<div id="container">
<iframeheight="4600" width="100%" frameborder="0" marginwidth="0"marginheight="0"
scrolling="auto"src="software.php">
</iframe>
</div>
<?php
require "foot.php";
```



install.php

Summary

- Built-in functions
- Making new functions
- Arguments - pass by value and pass by reference
- Including and requiring files
- Checking to see if functions are present