(See php_reference_sheet.png)

PHP – QUICK REFERENCE

Globe

Data Types
integer, float, boolean, string, array, object, resource, NULL

Variable Declarations
$variablename = <value>;
$anothervariable =& $variablename; (Assign by Reference)

Declare Array
$arrayname = array();

Initialize Array
$arrayname = array(<value1>, <value2>, <value3>);
$arrayname = array(<key> => <value>, <key> => <value>); (Define Keys)
$multiarray = array(<key> => array(<value1>,<value2>)); (Multi-dimensional)

Common Array Functions
sort(<array>); (Sort array assigns new keys)
asort(<array>); (Sort array maintain keys)
rsort(<array>); (Sort array in reverse, new keys)
arsort(<array>); (Sort array in reverse, maintain keys)
count(<array>); (Count elements)
count(<array>,COUNT_RECURSIVE); (Count multidimensional array)
array_push(<array>,<value>); (Push item onto end of array)
array_pop(<array>); (Pop item off end of array)

Comments
// Comment text
/* Multi-line comment text */
Comment text

Arithmetic Operators
+ (Addition), - (Subtraction), * (Multiplication), / (Division), % (Modulus)

Relational Operators
== (Equal), === (Equal with type comparison), != (Not equal), <> (Not equal), !== (Not Equal with type comparison), < (Less than), > (Greater than), <= (Less than or equal to), >= (Greater than or equal to)

Logical Operators
! (logical NOT), && (logical AND), || (logical OR), xor (logical XOR)

Assignment Operators
= (Assign), += (Addition), -= (Subtraction), *= (Multiplication), /= (Division), .= (Concatenation), %= (Modulus), &= (And), |= (Or), ^= (Exclusive Or), <<= (Left Shift), >>= (Right Shift)

String Concatenation
. (Period)

String Manipulation
substr(<string>,<start>,[<length>]);
strlen(<string>);
trim(<string>);
ltrim(<string>); // Trim left
rtrim(<string>); // Trim right
strtolower(<string>);
strtoupper(<string>);
str_replace(<search>,<replace>,<string>,[<count>]);
strpos(<string>, <search>);
strcmp(<string1>,<string2>); (Binary safe string comparison)
strcasecmp(<string1>,<string2>); (Binary safe case-insensitive comparison)
explode(<delim>,<string>,[<limit>]); (Break string into array)
implode(<delim>,<array>); (Join array into string separated by delim)

Cookies
setcookie (<cookiename>, [<value>],[<expire_time_in_secs_since_epoch>]);
$_COOKIE['cookiename']; (Returns value of cookie)

Sessions
session_start(); (Create session)
$_SESSION['key_name'] = value; (Set session variable)
$variablename = $_SESSION['key_name']; (Retrieve value from session variable)
session_destroy(); (Destroy session)

Error Handling
```
try {
    <statements that may cause error>;
}
catch (<Exception Class> $exception_name)
{
    <statements to execute when error is caught>;
}
```

Super Globals
$GLOBALS (Access all global variables in script)
$_SERVER (Access web server variables)
$_GET (Values passed to script through URL)
$_POST (Values passed to script through HTTP Post)
$_COOKIE (Values passed by user cookie)
$_FILES (Values passed by HTTP Post File Uploads)
$_ENV (Values passed to script via the environment)
$_REQUEST (Values passed by URL, HTTP Post, or user Cookies)
$_SESSION (Values passed through user's session)

If Else
```
if (<condition 1>)
   { <statement 1>; }
elseif (<condition 2>)
   { <statement 2>; }
else
   { <statement 3>; }
```

Inline If (Ternary)
<condition> ? true : false;

For Loop
```
for (<initialize>;<condition>;<update>)
{
   <statements>;
}
```

For Each Loop
```
foreach (<array> as [<value> |<key> => <value>])
{
   <statements>;
   [break];
   [continue];
}
```

While Loop
```
while (<condition>)
{
   <statements>;
}
```

Do-While Loop
```
do
{
   <statements>;
} while (<condition>);
```

Switch
```
switch (<expression>)
{
   case <literal or type>:
      <statements>;
      [break;]
   case <literal or type>:
      <statements>;
      [break;]
   default:
      <statements>;
}
```

Function Structure
```
function <function_name>([<parameters>])
{
   <statements>;
   [return <value>;]
}
```

Class Structure
```
class <class_name> [<extends base_class>]
{
   [var | <modifiers*>] [<class member variables>];
   [<modifiers*>] function <function_name>([<parameters>])
   {
      <statements>;
   }
}
```

* Modifiers <public | pr ivate | static> are implemented in PHP5

Declare and Use Class
$variable = new class_name();
$variable->function_name();
class_name::function_name(); (Static call)

GLOBE LABS' APIS

Globe

OVERVIEW AND REGISTRATION

Getting Started

- Working knowledge of HTTP, XML REST and/or SOAP in the language of your choice (PHP)
- A publicly accessible web server for processing incoming HTTP requests (for Mobile Originating messages & API messages)
- A provisioned account to access to the API

Globe

Hi globedemo1

You are receiving this email because you requested access to the Globe Labs APIs. If you did not make this request, contact us at globelabs@globetel.com.ph with "API Request Error" as the subject line.

Please take note of the following information:

WSDL: http://iplaypen.globelabs.com.ph:1881/axis2/services/Platform?wsdl
URL endpoint: http://iplaypen.globelabs.com.ph:1881/axis2/services/Platform
Access Number or Short-code: 2373
SMS and MMS Suffix (for receiving messages): 0303
Username (uName): [redacted]
Password (uPin): [redacted]
URL CALLBACK: http://www.dummy.com

Before using the API, make sure to define your URL call back and registered Globe numbers. To do this, go to http://202.126.34.119:1888/login.aspx, then login using the username(uName) and password(uPin) contained in this email. You'll be presented with a webtool that you can use to define / edit your callback URL and registered Globe numbers.

Remember, check the Globe Labs website for updates.

Thank you.

Globe Labs

Globe

SMS/MMS API

Please enter the following information below for us to complete your request.

Application URL:

This is the URL our system will access in order to send your application messages.

http://freeformsoftware.org/globe/lbs.php

(If you're not sure yet what your application URL will be, you may fill up this field later).

Allowed Mobile Numbers: These are the mobile numbers your application is allowed to send messages to and receive messages from.

(You may add numbers at a future time but cannot remove or edit previously entered numbers)

Submit

Mobile No(s):

09175889615		
09178306730		

Total: 2

| Page 1 of 1 |

SMS: 3 / 5 available today
MMS: 5 / 5 available today
LBS: 0 / 0 available today

Globe

SENDING MESSAGES

Globe

Sending a message (SMS)

sendSMS	
uName	API username
uPin	API password
MSISDN	Subscriber number
messageString	Message to be sent
Display	Notification / Inbox
udh	User data header
mwi	Message waiting indicator
coding	Bit encoding

Globe

```
POST /axis2/services/Platform/ HTTP/1.0
Host: 202.126.34.122:1881
User-Agent: NuSOAP/0.7.3 (1.114)
Content-Type: text/xml; charset=UTF-8
SOAPAction: ""
Content-Length: 820

<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/
encoding/'
SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
  <SOAP-ENV:Body>
    <ns3676:sendSMS xmlns:ns3676='http://ESCPlatform/xsd'>
      <uName xsi:type='xsd:string'>username</uName>
      <uPin xsi:type='xsd:string'>password</uPin>
      <MSISDN xsi:type='xsd:string'>09175889615</MSISDN>
      <messageString xsi:type='xsd:string'>testing sendSMS function</
messageString>
      <Display xsi:type='xsd:string'>1</Display>
      <udh xsi:type='xsd:string'></udh>
      <mwi xsi:type='xsd:string'></mwi>
      <coding xsi:type='xsd:string'>0</coding>
    </ns3676:sendSMS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<?xml version='1.0' encoding='UTF-8'?>

<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'>
  <soapenv:Body>
    <ns:sendSMSResponse xmlns:ns='http://ESCPlatform/xsd'>
      <ns:return>201</ns:return>
    </ns:sendSMSResponse>
  </soapenv:Body>
</soapenv:Envelope>
```


Globe

Sending a message (MMS)

sendMMS	
uName	API username
uPin	API password
MSISDN	Subscriber number
subject	The subject line for the message
smil	SMIL formatted message

Globe

A note on SMIL

- Synchronized Multimedia Integration Language – similar to HTML
- MMS messages have their display area divided into different sections
- http://open.movilforum.com/files/documentacion/documentacion31_2.pdf

```
<smil>
<head>
 <meta name="title" content="My Message" />
 <meta name="author" content="Greg Igaya" />
 <layout>
  <root-layout width="160" height="120"/>
  <region id="Image" width="100%"
    height="80" left="0" top="0" />
  <region id="Text" width="100%"
    height="40" left="0" top="80" />
 </layout>
</head>
<body>
<par dur="8s">
 <img src="FirstImage.jpg" region="Image" />
 <text src="FirstText.txt" region="Text" />
 <audio src="FirstSound.amr"/>
</par>
<par dur="7s">
 <img src="SecondImage.jpg" region="Image" />
 <text src="SecondText.txt" region="Text" />
 <audio src="SecondSound.amr" />
</par>
</body>
</smil>
```


Globe

Code Exercise

① Review send-sms-simple.phpx and get it to run using your environment and account (15 mins)

② Alter send-sms-simple.phpx to set the message and recipient via a HTML form (20 mins)

③ Review send-mms-simple.phpx and get it to run using your environment and account (15 mins)

④ Alter send-mms-simple.phpx to obtain an image or file from the internet depending on the input from a web-form and send it via MMS (30 mins)

Globe

RECEIVING MESSAGES

Globe

Receiving a message (SMS)

2373 +
4-digit
code

XML to
Callback URL

```
<?xml version="1.0" encoding="utf-8"?>
<message>
  <param>
    <name>messageType</name>
    <value>SMS</value>
  </param>
  <param>
    <name>id</name>
    <value>xxxxxxxxxxxxx</value>
  </param>
  <param>
    <name>source</name>
    <value>xxxxxxxxxx</value>
  </param>
  <param>
    <name>target</name>
    <value>xxxxxxxxxxxx</value>
  </param>
  <param>
    <name>msg</name>
    <value>xxxxxxxxxxxx</value>
  </param>
  <param>
    <name>udh</name>
    <value></value>
  </param>
</message>
```

parameters	
messageType	Type of message (SMS)
source	Subscriber's number
target	2373 + 4 digit access code
msg	Message sent
id	Message identifier
udh	User data header

Receiving a message (MMS)

2373
Subject line
4-digit
code

XML to
Callback URL

Globe

```
<?xml version="1.0" encoding="utf-8"?>
<message>
  <param>
    <name>messageType</name>
    <value>MMS</value>
  </param>
  <param>
    <name>subject</name>
    <value>subject123</value>
  </param>
  <param>
    <name>source</name>
    <value>123</value>
  </param>
  <param>
    <name>target</name>
    <value>123</value>
  </param>
  <param>
    <name>file</name>
    <value>
      <file>http://localhost:1234/testing.jpg</file>
      <file>http://localhost:1234/testing.txt</file>
    </value>
  </param>
</message>
```

parameters	
messageType	Type of message (SMS)
source	Subscriber's number
target	2373 + 4 digit access code
file	File list (more file nodes)
subject	Subject used by subscriber

Globe

Using CURL

- cURL groks URLs
- curl –d @sms.xml –H "Content–type: text/xml" –v http://localhost:8888/catch.php

sms.xml

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<param>
<name>id</name>
<value>2373017320090210204843</value>
</param>
<param>
<name>messageType</name>
<value>SMS</value>
</param>
<param>
<name>target</name>
<value>23730173</value>
</param>
<param>
<name>source</name>
<value>09175889615</value>
</param>
….
```


Globe

Code Exercise

① Review catch.phpx and get it to run using your own environment and account (15 mins)

② Revise catch.phpx to send an SMS response from an SMS input (30 mins)

③ Create a simple SMS-based service which sends an SMS / MMS depending on keyword input. The service can be either use time or currency conversion. (40 mins)

Globe

LOCATION BASED SERVICES

Globe

Location Based Services API

- Allows an application to determine the approximate location of subscriber through the network
- Uses Enhanced Cell ID to determine position
  - Uses location of serving cell tower as base and uses Timing Advance and Network Management Records to fine-tune the measurement

Globe

Enhanced Cell-ID

Timing Advance

Estimate distance to tower

Cell ID Only

Full coverage of cell site 500m to several km

Known coordinate:
14.343231, 121.343543

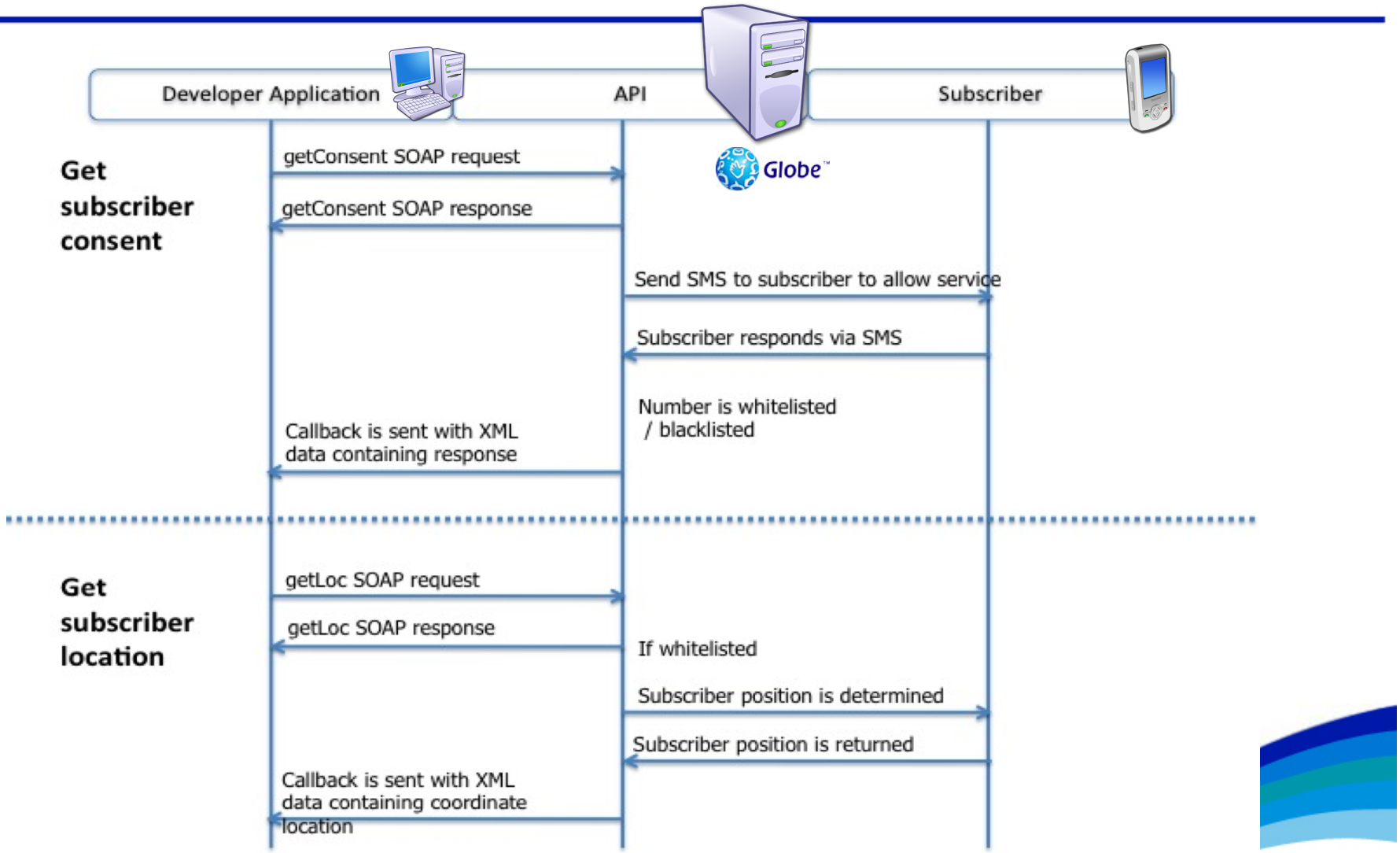
Globe

Enhanced Cell-ID

- Most-common and cost effective but least accurate vis a vis other technologies such as Angle of Arrival, Time Difference of Arrival (triangulation) or GPS
- Accuracy varies on cell size, but could be 300 meters to several kilometers. Accuracy improves on density.
- no speed or direction of travel is available

Globe

LBS API - Overview



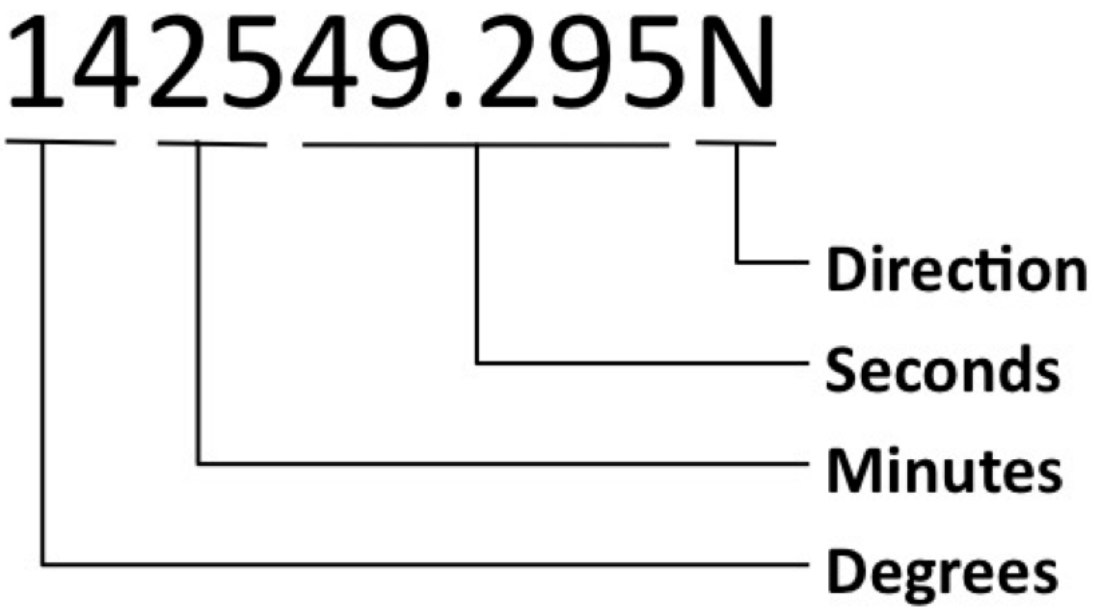
Globe

LBS Notes

- WGS-84 spatial reference system
- Returns coordinate data to application
  - Degrees Minutes Seconds
- Known Limitations
  - Accuracy of 300 meters in dense cell coverage areas up to several kilometers in rural areas
  - 2G / 3G network divergence
  - 1 TPS limit, 6PM – 2AM restriction

Globe

- X – Latitude – parallel to equator
- Y – Longitude – perpendicular to equator
- How to read returned DMS data



Application Possibilities

300-1000 meters	< 50 meters	< 10 meters
✓Location sensitive call routing ✓Fleet Management ✓City Guide / Yellow Pages / Location sensitive Info Directory or Stream ✓Traffic Alerts ✓Find your friend ✓Remote workforce monitoring ✓Points of Interest / Tourist Attraction ✓Location contextual entertainment ✓Location aware advertising – pull (user initiated) ✓Dating / meet-up ✓Geo-tagging	✓Emergency services ✓Asset tracking ✓Finding ✓Location sensitive charging and billing	✓Safetly and medical monitoring ✓Step-by-step navigation ✓Location sensitive charging and billing ✓Stolen vehicle recovery ✓Location aware advertising – push (telco initiated)

To be compelling, applications built on top of the LBS API need to integrate other proprietary data

- Road maps and POI
- Coordinate locations of restaurants / venues
- Existing social networks

Globe

Code Exercise

① Obtain consent from a subscriber and record consent approval in persistent storage. Use a database (mysql) or the filesystem to accomplish this. (30 mins)

② Determine a subscriber's location and map the coordinates on Google Maps using Google's static map API. http://code.google.com/apis/maps/documentation/staticmaps/ (45 mins)

Globe

Common SVN commands

- Obtain / Update your working copy
  - svn checkout /update
- Make changes
  - svn add
  - svn delete
  - svn copy
- Examine your changes
  - svn status
  - svn diff
  - svn revert
- Merge others' changes into your working copy
  - svn update
  - svn resolved
- Commit your changes
  - svn commit

Globe