FRESNO STATE
Discovery. Diversity. Distinction.

Web Programming (CSci 130)

Department of Computer Science
College of Science and Mathematics
California State University Fresno
H. Cecotti

Learning outcomes

- Introduction to PHP
  - Programming on the server side
  - PHP syntax and structure
- Last main brick in this course
  - Plateau after PHP syntax/structure
    - → Combination of HTML5+CSS3+JS+PHP
    - Possibility to do many types of dynamic websites
- Good practice:
  - Take notes as comments in the code
  - The documents in Canvas are yours, save them, keep them, make them yours for future reference (job interview, work, projects, …)

Introduction

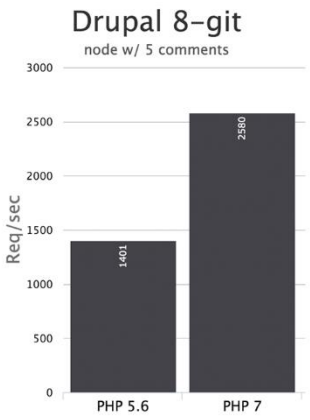
- PHP: **Hypertext Preprocessor**
  - Highly used, free, runs on Windows, Linux, Mac OS X,…
    - **Most popular** server-side language
  - Foundation of many websites
    - Facebook, Wikipedia, Tumblr,…
    - 2016: PHP > 80% of the websites on the internet
- Main competitors
  - ASP.NET
    - In the .NET world of MS
    - 2016: ASP = 15% of the websites on the internet
  - Ruby, Python,…
  - Evolution of languages
    - 90s → Perl for CGI

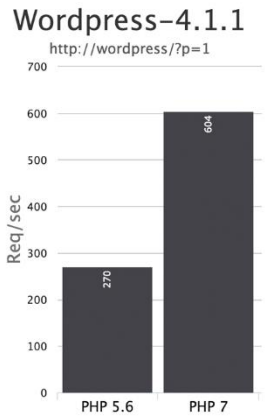
Introduction

- **PHP code**
  - Executed on the server side
    - Apache
  - The browser cannot understand the code in PHP
    - /!\ PHP is installed on the server, not on the client!
- Javascript
  - Executed on the client side
- What we need?
  - A web server (XAMPP)
    - You should have it already – tested with AJAX
  - Text editor
    - Notepad++ or other programming environment for PHP

Different versions

- Like many languages, they evolve over time…
- PHP5, PHP7…
  - ➢ PHP7 has new features
    - Type declarations
    - Error handling
- Performance





HTML → PHP

- Insert a piece of PHP in an HTML page
  - Index.html → Index.php
  - php extension
- Inside HTML code
  - Example
    - echo: output the content of the string on a webpage

```
<html>
<head>
 <title>Test</title>
</head>

<body>
 <p>un bout de code en HTML</p>
<?php
 echo 'My first script in PHP';
?>
</body>
</html>
```

Syntax

- **Code**
  - <?php PHP Code ?>
  - File extension .php
  - Easier to debug ! PHP will tell you where it is wrong with more precise error messages
- Simple to use
- Comments (// , /* */ , #)

```php
<?php
// comment on one line

/* comment
on several
lines */
?>
```

- Variables
  - $ + string
    - Example: $name
  - Assign a value:
    - $name="Jim";
    - $value=54;

```php
<?php
$name = "the mountain";
// $name contains the string mountain

$mmynumber = 12;
// $mynumber contains the value 12

$5toto = "test";
// Not valid
?>
```

Syntax

- **Variable**
  - Declaration
    - No command
    - No type
  - It starts with the $ sign + the name of the variable
  - The name
    - must start with a letter or the underscore character
    - cannot start with a number
    - can only contain alphanumeric chars and _ ([A-z],[0-9], _)
    - **case-sensitive**
      - $name and $Name are 2 different variables
  - Scope
    - Local, global, static

Predefined variables

- Main types
  - **$GLOBALS**
    - To access global variables anywhere in a PHP script
    - $GLOBALS[x] where x is the name of the variable
  - **$_SERVER**
    - Information about headers, paths, script locations…
    - $_SERVER[x] where x is the name of the parameter
      - x= 'PHP_SELF', 'SERVER_ADDR', 'SERVER_NAME', 'SERVER_SOFTWARE',…
  - **$_REQUEST**
    - To collect data after submission of a form
  - **$_POST**
    - To collect data after submission of a form with **POST**
  - **$_GET**
    - To collect data after submission of a form with **GET**
  - **$_FILES**
  - **$_ENV**
  - **$_COOKIE**
  - **$_SESSION**

Predefined variables

- **Remarks**
  - Predefined variables in PHP are **superglobals** (automatic global variables)
    - → They are available in **all** scopes throughout a script.
    - → no need to do global $variable; to access them within functions or methods.
- Official link with examples
  - http://www.php.net/manual/en/reserved.variables.php

Data types

1. String
2. Integer (number in Javascript)
3. Float (number in Javascript)
4. Boolean (true/false)
5. Array (object in Javascript)
6. Object
   - As an instance of a class ! (not like in old Javascript)
7. NULL (and a single value)
8. Resource

Operators

- Arithmetic
  - + Addition $x + $y
  - - Subtraction $x - $y
  - * Multiplication $x * $y
  - / Division $x / $y
  - % Modulus $x % $y
  - ** Exponentiation $x ** $y
- Assignment
  - Like in Javascript
- Increment/decrement
  - Like in Javascript
    - ++$v
    - $v++

Operators

- Comparisons
  - == Equal $x == $y
    - Returns true if $x is equal to $y
  - === Identical $x === $y
    - Returns true if $x is equal to $y, and they are of the **same type**
  - != Not equal $x != $y
    - Returns true if $x is not equal to $y
  - <> Not equal $x <> $y
    - Returns true if $x is not equal to $y
  - !== Not identical $x !== $y
    - Returns true if $x is not equal to $y, **or** they are not of the same type
  - > Greater than $x > $y
    - Returns true if $x is greater than $y
  - < Less than $x < $y
    - Returns true if $x is less than $y
  - >= Greater than or equal to $x >= $y
    - Returns true if $x is greater than or equal to $y
  - <= Less than or equal to $x <= $y
    - Returns true if $x is less than or equal to $y
  - <=> : comparison between $x and $y
    - Returns -1 if x<y, 0 if x==y and 1 if x>y

Syntax

- See example on Canvas
  - **php_syntax.php**
    - To use the file like:
      - http://localhost/mysite1/php_syntax.php
      - If you double click
    - Add your own comments and examples to the file
      - It is **YOUR** file → keep it as a reference and enrich it with your own examples
- Comments about the syntax
- Variables
- Define (constants)
- Conditional structures
  - If, Switch, …
- Loops
  - For, For each, While, …

Conclusion

- **PHP**
  - Robust and flexible language
    - "too flexible?"
  - Your code **must be structured** (like in Javascript)
  - Many elements available online
    - Temptation of copy/paste → code is hard to read, hard to maintain
- With PHP, you can
  - Generate dynamic page content
  - Manage files on the server
    - Create/Open/Close/Read/Write/Delete
  - Obtain data from forms
  - Send/receive cookies
  - Edit databases
    - Add, delete, modify elements