Learning PHP

A GENTLE INTRODUCTION TO THE WEB'S MOST POPULAR LANGUAGE

Intro to PHP

- **Static Websites:** the content does not change and is fixed. The content is the same for all visitors. **E.g. personal websites**
- **Dynamic Websites:** pictures and contents are different for different visitors.**E.g. Amazon.com**

- PHP is a programming language for building **dynamic websites**
- PHP is a **server-side** language
  - **Example**: JavaScript is a **client-side** language
  - **Example**: ASP.NET is a **server-side** language
- PHP is free
- **OS X** and most **Linux** distributions come with PHP **already installed**.

Static Webpages

- PHP runs on the server not on the client

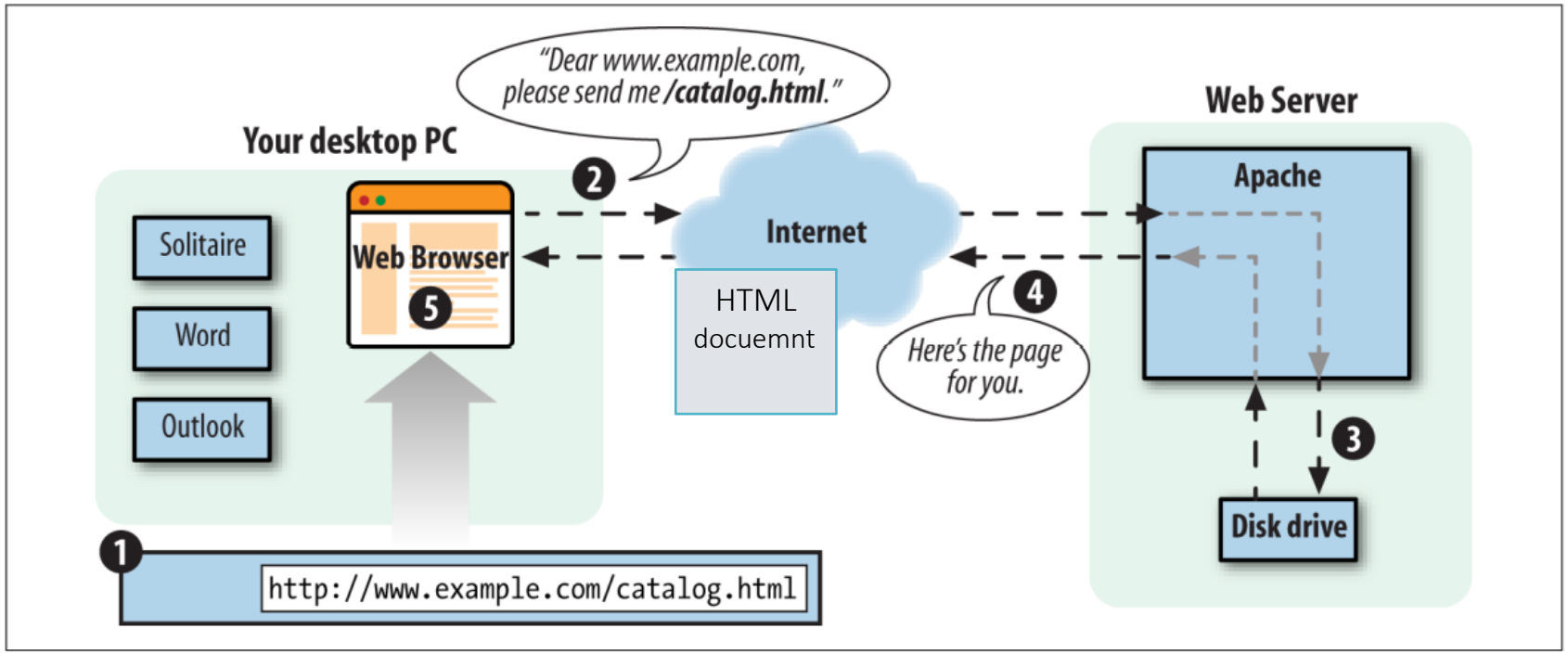

Figure 1-1. Client and server communication without PHP

Dynamic Webpages

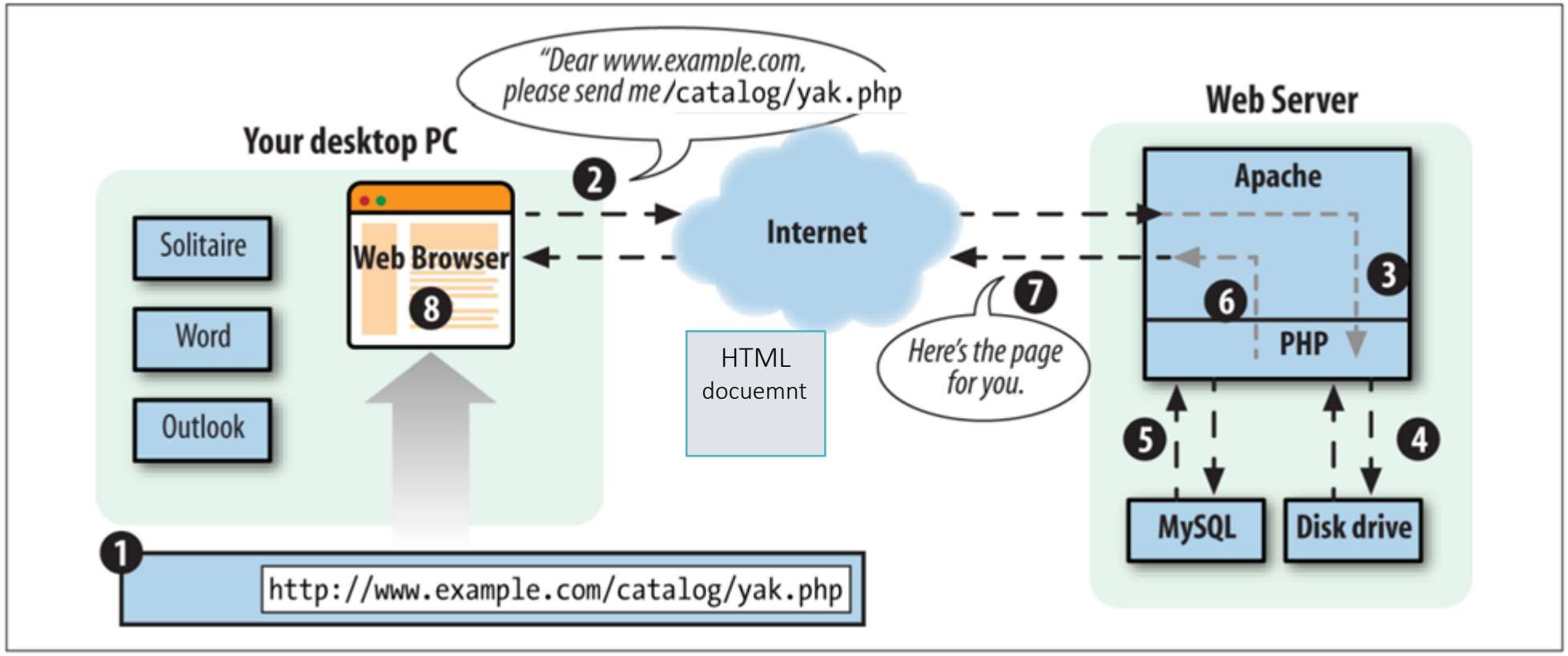

Figure 1-2. Client and server communication with PHP

PHP and PHP Engine

- PHP is a language
- PHP Engine is the software
  - Running on a Web Server
  - Understanding PHP language and executes the commands
  - For example, talking to DBMS, retrieving data and generating pages
- **PHP Engine** is written in the **C programming** language
- PHP works with a web server running on Windows, Mac OS X, Linux, and many other versions of Unix.
- PHP works on **Web Servers** such as Apache, nginx, MS IIS, or any web server that supports CGI standard.
- PHP works on many **DBMSs**: MySQL, PostgreSQL, Orcle, MS SQL Server, SQLite, Redis, and MongoDB
- PHP is used on more than 200 million different websites, including giants like **Facebook**, **Wikipedia**, and **Yahoo**

Basics of PHP

- It can be part of a HTML file
- It starts with **<?php** and ends with **?>**
- **PHP engine** executes only code between **<?php and ?>,** text out of them is ignored
- If there is no code at the end of the file, **?> end tag is optional**
- There can be multiple blocks of PHP code in an HTML file

```
<span>Five plus five is:<span>
<?php print 5 + 5; ?>
<p>
Four plus four is:
<?php
print 4 + 4;
?>
</p>
<img src="vacation.jpg" alt="My Vacation" />
```

Basics of PHP

- PHP is a case-sensitive language
  - **$_POST** and **$_post** are different
  - **But**, Language keywords (such as print) and function names are **not case-sensitive**.

Example 1-13. Keywords and function names are case-insensitive

```php
<?php
// These four lines all do the same thing
print number_format(320853904);
PRINT Number_Format(320853904);
Print number_format(320853904);
pRiNt NUMBER_FORMAT(320853904);
?>
```

Basics of PHP

Comments

1. are an essential part of any program. By **explaining in plain language** how the programs work, comments make programs much more understandable.
2. You can also disable a part of code for testing your program

- inline comments //
- Inline comments #
- multiline commensts /* */

Example 1-15. Multiline comments

```php
<?php
/* We're going to add a few things to the menu:
   - Smoked Fish Soup
   - Duck with Pea Shoots
   - Shark Fin Soup
*/
print 'Smoked Fish Soup, Duck with Pea Shoots, Shark Fin Soup ';
print 'Cost: 3.25 + 9.50 + 25.00';
```

Example 1-14. Single-line comments with // or #

```php
<?php
// This line is a comment
print "Smoked Fish Soup ";
print 'costs $3.25.';

# Add another dish to the menu
print 'Duck with Pea Shoots ';
print 'costs $9.50.';
// You can put // or # inside single-line comments
// Using // or # somewhere else on a line also starts a comment
print 'Shark Fin Soup'; // I hope it's good!
print 'costs $25.00!'; # This is getting expensive!

# Putting // or # inside a string doesn't start a comment
print 'http://www.example.com';
print 'http://www.example.com/menu.php#dinner';
?>
```

First Program: HELLO WORLD!

The .PHP file

```
<!DOCTYPE html>
<html>
<head>
    <title>PHP says hello</title>
</head>
<body>
    <b>
        <?php
        print "Hello, World!";
        ?>
    </b>
</body>
</html>
```

OUTPUT:

```
<!DOCTYPE html>
<html>
<head>
    <title>PHP says hello</title>
</head>
<body>
      <b>
            Hello, World!
      </b>
</body>
</html>
```

Another Example

.html File

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title></title>
</head>
<body>
    <form method="POST" action="sayhello.php">
        Your Name: <input type="text" name="user" />
        <br />
        <button type="submit">Say Hello</button>
    </form>

</body>
</html>
```

.php file

```
<?php
print "Hello, ";
// Print what was submitted in the form parameter called 'user'
print $_POST['user'];
print "!";
?>
```

Basics of PHP

- Every program is composed of **statements**
- Statments end with **semi-colon** (;)
- You can write **multiple PHP statements** on the same line of a program as long as they are separated with a **semicolon**.
- You can put as many **blank lines between statements** as you want. The PHP engine ignores them.
- It is recommended to put **one statement on a line** and blank lines between statements only when it improves the readability.

Example 1-9. This PHP is too cramped

```
<?php print "Hello"; print " World!"; ?>
```

Example 1-10. This PHP is too sprawling

```
<?php

print "Hello";


print " World!";

?>
```

Example 1-11. This PHP is just right

```
<?php
print "Hello";
print " World!";
?>
```