



# PHP

**Libraries, HTML, JS, and PHP integration**

COS216

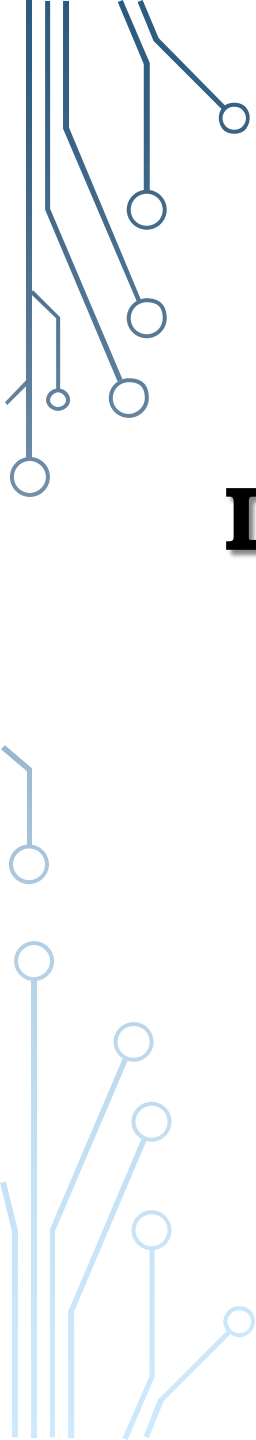
AVINASH SINGH

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF PRETORIA

# PHP – LIBRARIES

- PHP provides a lot of native functionality
  - Some more advanced features require the installation of separate libraries
  - These libraries are called “PHP extensions”
- 
- Extensions have to be installed via the web panel (eg: cPanel) and must be enabled in the PHP configuration
  - You do not have access to this on the Wheatley server, however, if you purchase your own web server, you can typically install extensions
  - Hundreds of extensions exist (official and unofficial)



# PHP – LIBRARIES

Name	Description
MySQL, MySQLi, PDO	Various MySQL database interfaces
PostgreSQL, SQLite, MS-SQL, Oracle, etc	Various other SQL-based databases
cURL	Make HTTP requests from PHP
Bzip2, ZIP, Zlib	Compress and decompress archive files
MCrypt, MHash	Various encryption and hashing algorithms
IMAP, POP3, SMTP	Email services
POSIX, Win32API	Low-level OS specific libraries
PDF	Create, manipulate, and interpret PDF documents
Sockets	Sockets in PHP
Printer	Access to printers connected to the server
GD, Imagick	Image manipulation and conversion
FFMpeg (unofficial)	Video and audio manipulation and conversion



# PHP – INTEGRATION

- PHP code can be integrated/called in the following ways:
  1. Create a PHP script that prints HTML which is rendered by a browser
  2. Make HTML form requests to PHP
  3. Make AJAX calls from JS to PHP
- The first two options are “old” ways of making requests
- Arguably, a well-written/modern site should only employ AJAX calls on APIs
  - Write a static HTML page which uses AJAX/JS to update in the background
  - Even HTML form data can/should be transmitted via AJAX
  - Other sites, mobile and desktop apps can then use the exact same API

# PHP – OLD APPROACH

- Dynamically assemble HTML on the server side based on requests
- Does not require any JS
- Requests are done by HTML forms



# PHP – OLD APPROACH

- Example

# PHP – NEW APPROACH

- Use a static HTML page
- An API is implemented in PHP
  - Only contains API call functionality
  - Does not contain any HTML/CSS/JS code (cleaner)
- All requests, including form requests, are done via AJAX in JS
- JS updates the page once the response comes back from the API
- Security must be taken into consideration



# PHP – NEW APPROACH

- Example



# PHP – EXAMPLES

## Welcome to Black Hole

Username:   
Password:

## Welcome to Black Hole

You are not who you say you are

Username:   
Password:

## Welcome to Black Hole

Hi there satoshi, you are entering the black hole ...

## Welcome to Black Hole

Hi there satoshi, you left the black hole ...

Username:   
Password:

## Welcome to Black Hole

satoshi, you are already inside the black hole, there is no way to leave ...

# PHP – COMPARISON

Description	Old Approach	New Approach
Easy to scale for very large systems (with a lot of code)	No	Yes
Easy to extend features, add new requests, or remove unused ones	No	Yes
Easy to add any app on top of system (website, mobile app, desktop program)	No	Yes
Smooth user interaction (website looks almost like a standalone program due to background updates)	No	Yes
Easy for other sites to integrate your code or API, or use your data	No	Yes
Easy to understand and maintain	Depends	Depends



