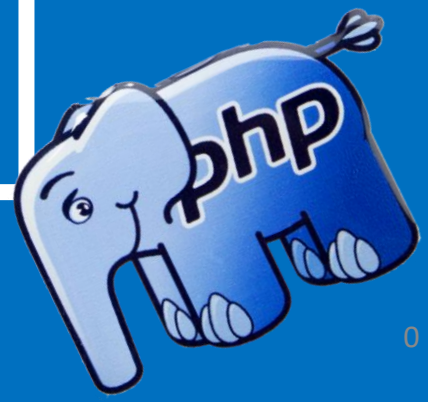
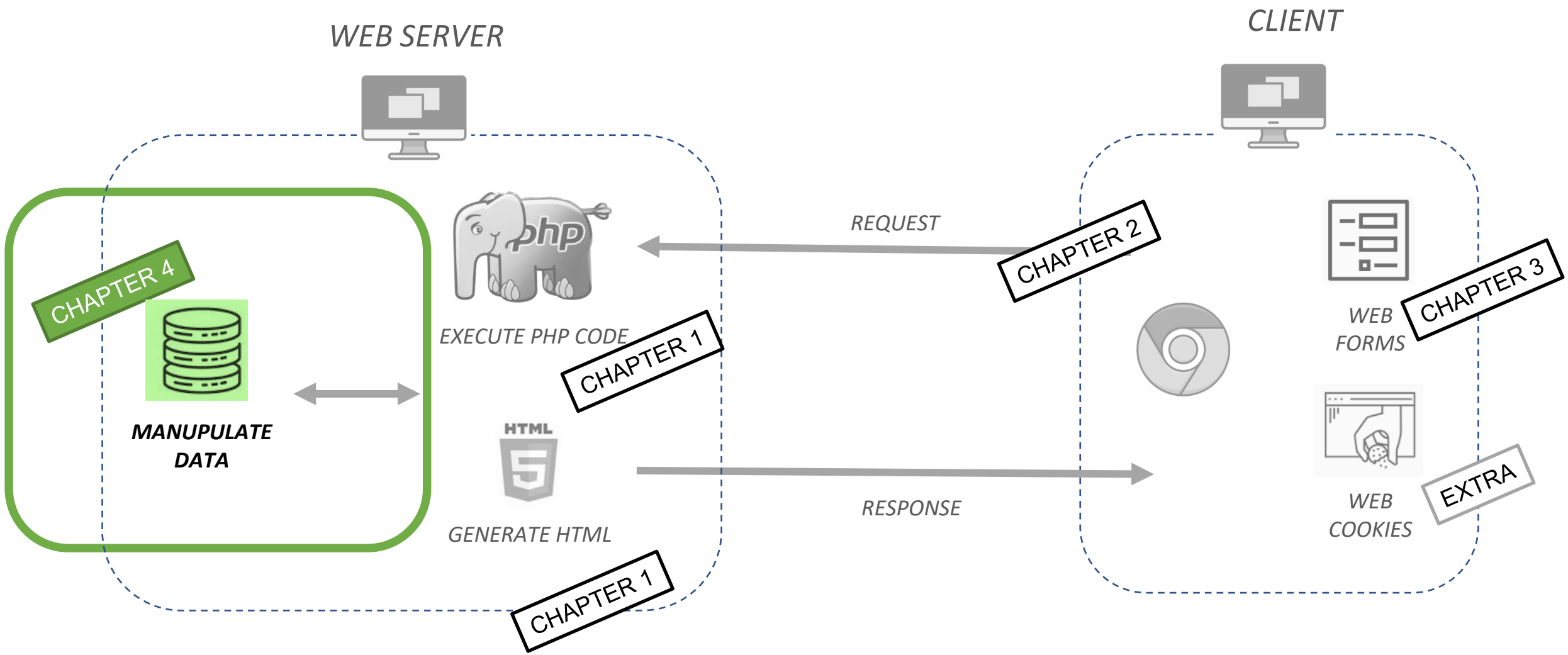
PHP

CHAPTER 4 – CONNECT TO SQL SERVER



PHP COURSE ROADMAP



OBJECTIVES FOR TODAY

- ✓ **Connect** PHP to database
- ✓ **Run** SQL **queries** on PHP code
- ✓ Get **results** from querries

EXPLORE

15 MIN

INDIV

Activity 1

- Can you create and display many posts ?
- Can you share posts with your friends if you share the link of your server ?
- What do you think about the variable *$posts* ? How long is its lifetime ?
  How many posts are stored ?
- Does the server store the data from request parameters ($_POST) ?

```php
$posts = [
    [
        'name' => 'Rady',
        'message' => 'Hello !'
    ],
    [
        'name' => 'Hyacinthe',
        'message' => 'Lorem ipsum...'
    ]
];
```

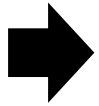
EXPLAIN

15 MIN

CLA
SS

PHP without database

$_GET
$_POST + 




For each PHP run, the variable *$posts* **is reset**

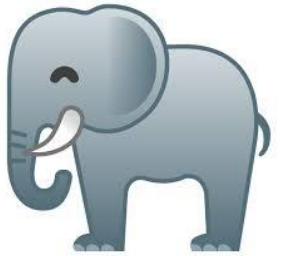
- PHP get the parameters from $GET or $POST
- PHP generates a HTML file from the PHP code

Fish memory !

We cannot keep anything on server side

PHP with database

- PHP get the parameters from $GET or $POST
- **PHP requests queries to the database on server**
- PHP generates a HTML file from the PHP code



Elephant memory !

Now we can keep and remember values on server side !

1 - **Connect** to database

PDO need **parameters** to know how to connect to the good database

```php
$host     = 'localhost'; // MySQL is running on the same computer as the web server
$database = 'mydatabase'; // Name of the database to use (first CREATE DATABASE with same name in MySQL)
$user     = 'root'; // Default username to connect to MySQL is root
$password = ''; // Default password to connect to MySQL is empty

$dsn = "mysql:host=$host;dbname=$database;charset=UTF8";
$db = new PDO($dsn, $user, $password);
```

PDO is an interface for accessing databases in PHP
With the ***new*** keyword, you create a new connection

Store the returned value in a variable **$db**, we will use it to execute SQL queries on the database !

2 – Execute a **query**

The **$db->query()** will execute a SQL query from PHP code

```
$db->query("CREATE TABLE users (id int PRIMARY KEY AUTO_INCREMENT, username VARCHAR(20));");
$db->query("INSERT INTO users (username) VALUES ('rady');");
$db->query("SELECT id, username FROM users WHERE id=1;");
```

2 – Execute a **query**

To insert data, just concatenate the data with the query:

```php
$newUser = 'hyacinthe';
$db->query("INSERT INTO users (username) VALUES ('" . $newUser . "');");
```

Same as:

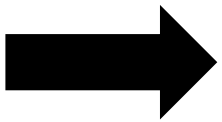
```php
"INSERT INTO users (username) VALUES ('hyacinthe');"
```

3- **Get results** from query

The **fetchAll()** function returns an array of associative arrays of query results

*If you expect only **one record** as result, you can use the **$statement->fetch()** function instead*

```php
$statement = $db->query("SELECT id, username FROM users;");
$users_list = $statement->fetchAll();
```



```
[
    [
        'id' => 1,
        'username' => 'rady'
    ],
    [
        'id' => 2,
        'username' => 'hyacinthe'
    ]
]
```

SQL **table** to PHP **arrays**

id	username
1	rady
2	hyacinthe

Users Table

fetchAll()

```
[
    [
        'id' => 1,
        'username' => 'rady'
    ],
    [
        'id' => 2,
        'username' => 'hyacinthe'
    ]
]
```

$user_list

PRACTICE

15 MIN

INDIV

Activity 2

Add database in the project from activity 1:

1. Create a new database *PHP_connect* in MySQL
2. Create table *posts* (from *posts.sql* file)
3. Complete the PHP code