



## 0.3 - Introduction to PHP

PHP is a different programming language than JavaScript. However, most of the concepts you learned are also applicable in this language. The syntax will differ between the languages. The following chapters will give you a brief introduction of the syntax of the basic programming concepts.

### Variables

In the third episode Jeffrey shows you how to declare and use variables. You will see that the most important difference between PHP and JavaScript is the use of the mandatory `$` character.

#### Step 3: Variables

Now that you're ready to begin coding, let's start at the top.  
What exactly is a variable, and why would you ever use or need

 <https://laracasts.com/series/php-for-beginners/episodes/3>



-  In the video Jeffrey starts a webserver from the terminal with `php -s localhost:8888`. We advise you to use XAMPP instead. After all, the first "P" of XAMPP stands for "php". You can achieve this by changing the documentroot of your Apache webserver to the folder where you store your PHP code (or vice versa: store your code in the documentroot).

### PHP and HTML

In the following episode, Jeffrey shows how PHP pages can be made more dynamically. But, before you can fully understand what's happening in the video, you should know the following:

## Query string

The query string is a part of the URL where you can pass sets of data (key-value pairs) to the server. This is an important fundamental concept, because it is used in almost every web application. This is an example of a URL with a query string:

```
https://example.com/path/to/page?name=ferret&color=purple
```

The `?` is the delimiter which separates the query string from the URL. The left part is the resource and the right part is the query string. The query string in the example consists of two key-value pairs delimited with the `&` character. A key-value pair is sometimes called a *parameter*.

For more details, see:

### Query string

A query string is a part of a uniform resource locator (URL) that assigns values to specified parameters. A query string commonly includes fields added to a base URL by a Web browser or other client application,

 [https://en.wikipedia.org/wiki/Query\\_string](https://en.wikipedia.org/wiki/Query_string)

## Superglobal ( `$_GET` )

A superglobal is a built-in variable in PHP that can be accessed anywhere in the scripts. There are several of these superglobals available (see the [PHP-manual](#)). Each superglobal has a different function. `$_GET` is the superglobal that contains all the key-value pairs of the query string.

## Security

Assume that your users are guilty until proven innocent (Jeffrey Way)

When building websites, you should be defensive on how to use input from your users. The `htmlspecialchars()` function is one of your lines of defense in this. This function returns a string where all dangerous characters are transformed into something less harmful.

Now, watch the video

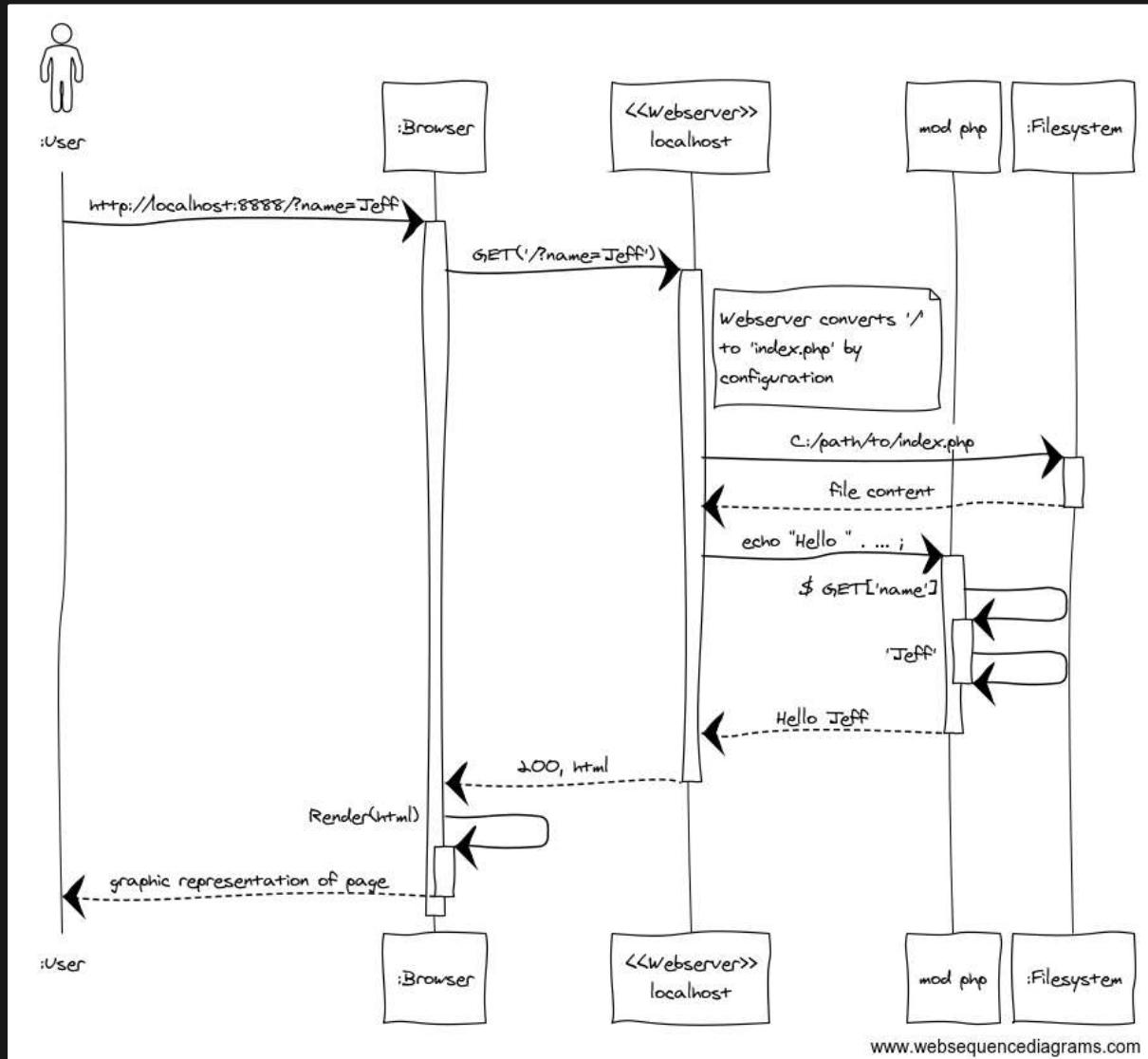
### Step 4: PHP and HTML

PHP is a language built for the web. As such, it mixes with regular HTML beautifully. Let's learn how that all works in this

 <https://laracasts.com/series/php-for-beginners/episodes/4>



In the video Jeffrey creates a small example using the query string en `$_GET` superglobal. The following sequence diagram shows how this example functions:



**Exercise 0.4** Do the *homework* assignment at the end of the video ("Try to recreate a sentence dynamically in very much the same form as we did earlier"). Add a second key-value pair of your choice (i.e. your age: `?name=Jeff&age=42`) to the query string and build another sentence with this. Do not forget to secure your user input with `htmlspecialchars`. If you like, you can challenge yourself by adding code that gives an appropriate response when one of the parameters is not present (or both).

## Separate PHP Logic From Presentation

In the next episode Jeffrey refactors his code to apply *Separation of Concerns*. Before you watch this episode it is good to understand this concept.

Wikipedia defines this concept as follows:

In computer science, separation of concerns (SoC) is a design principle for separating a computer program into distinct sections such that each section addresses a separate concern. ([wikipedia](#))

The most important reason to apply this is to make complex software more readable and maintainable. A well-known example of SoC is HTML/CSS and Javascript. Each of the three languages address a separate concern regarding a website.

- HTML is mainly used for organization of webpage content
- CSS is used for definition of content presentation style
- Javascript defines how the content interacts and behaves with the user

When applied correctly, a front-end developer whose sole concern is the layout of the content of a website should mainly need to work with CSS and does not need to have a deep understanding of Javascript.

SoC can be applied in other ways as well. It is the basic principle when dividing Object Oriented code into different classes. Encapsulation and Abstraction are principles that are related to SoC.

Now, watch the next episode.

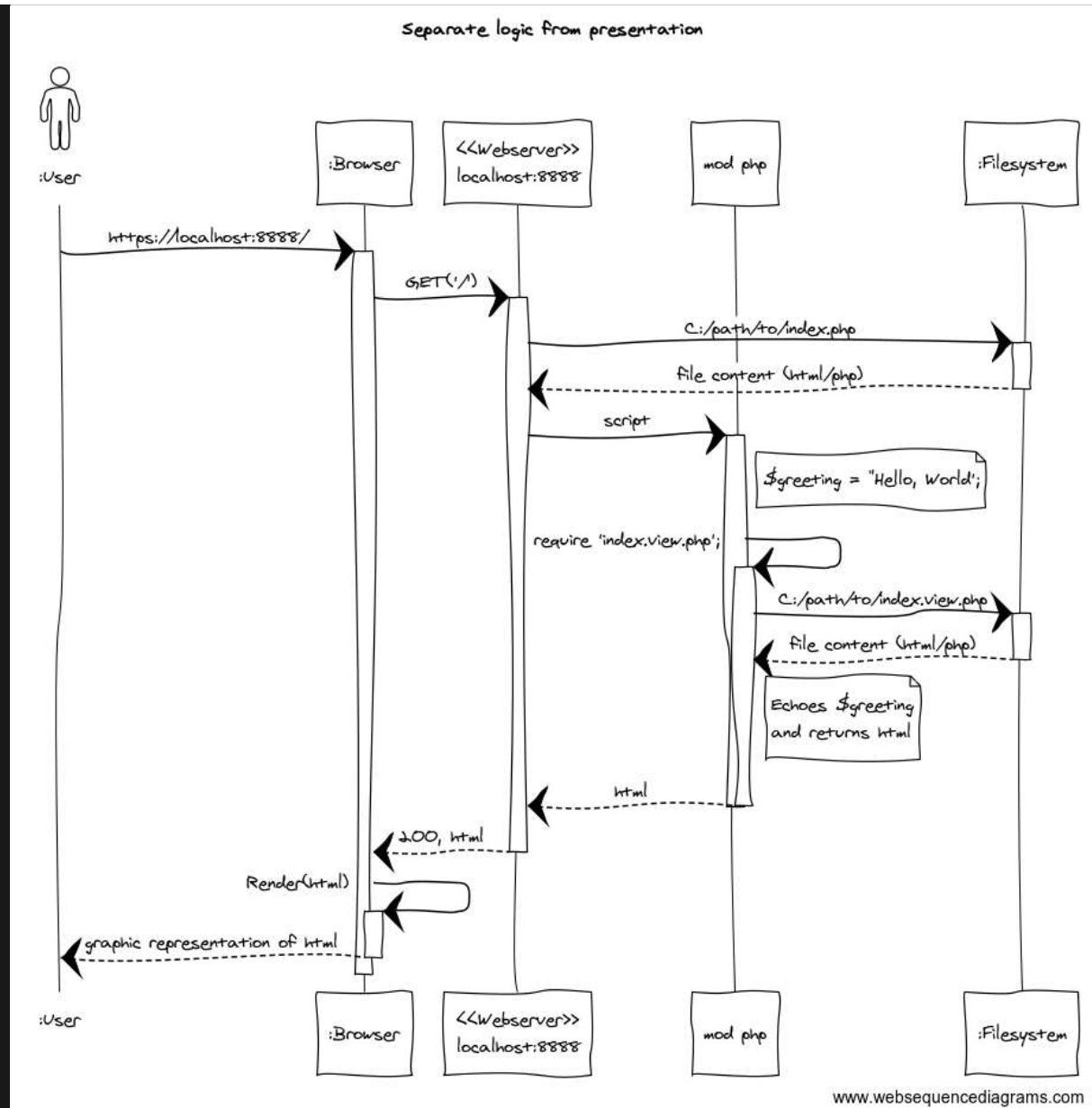
#### Step 5: Separate PHP Logic From Presentation

We'll touch on a slightly higher level topic in this episode, but only lightly. If you stick with it long enough, you'll begin to hear

 <https://laracasts.com/series/php-for-beginners/episodes/5>



Jeffrey divided PHP logic and the presentation of the data into two different files. If modeled into a sequence diagram, the interactions look like:



This model starts to get complicated and therefore hard to read. It is possible to simplify this. There are two ways of simplifying a sequence diagram model.

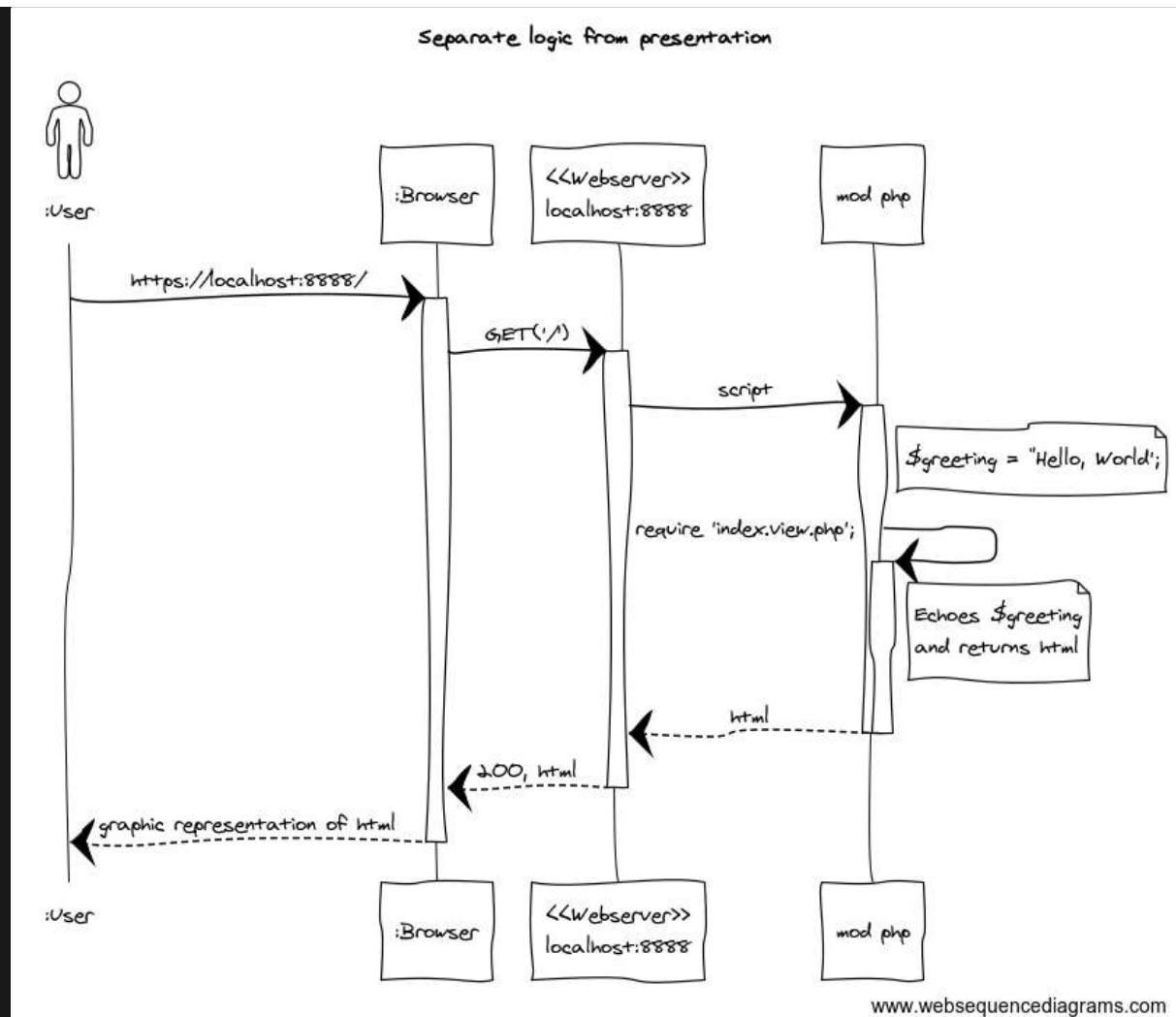
## Simplifying our Sequence Diagram

### Omit obvious sequences

Sequences that everyone understands and are always present might be left out. These sequences do not add anything to the subject the model is trying to explain. In the example, there are two obvious sequences:

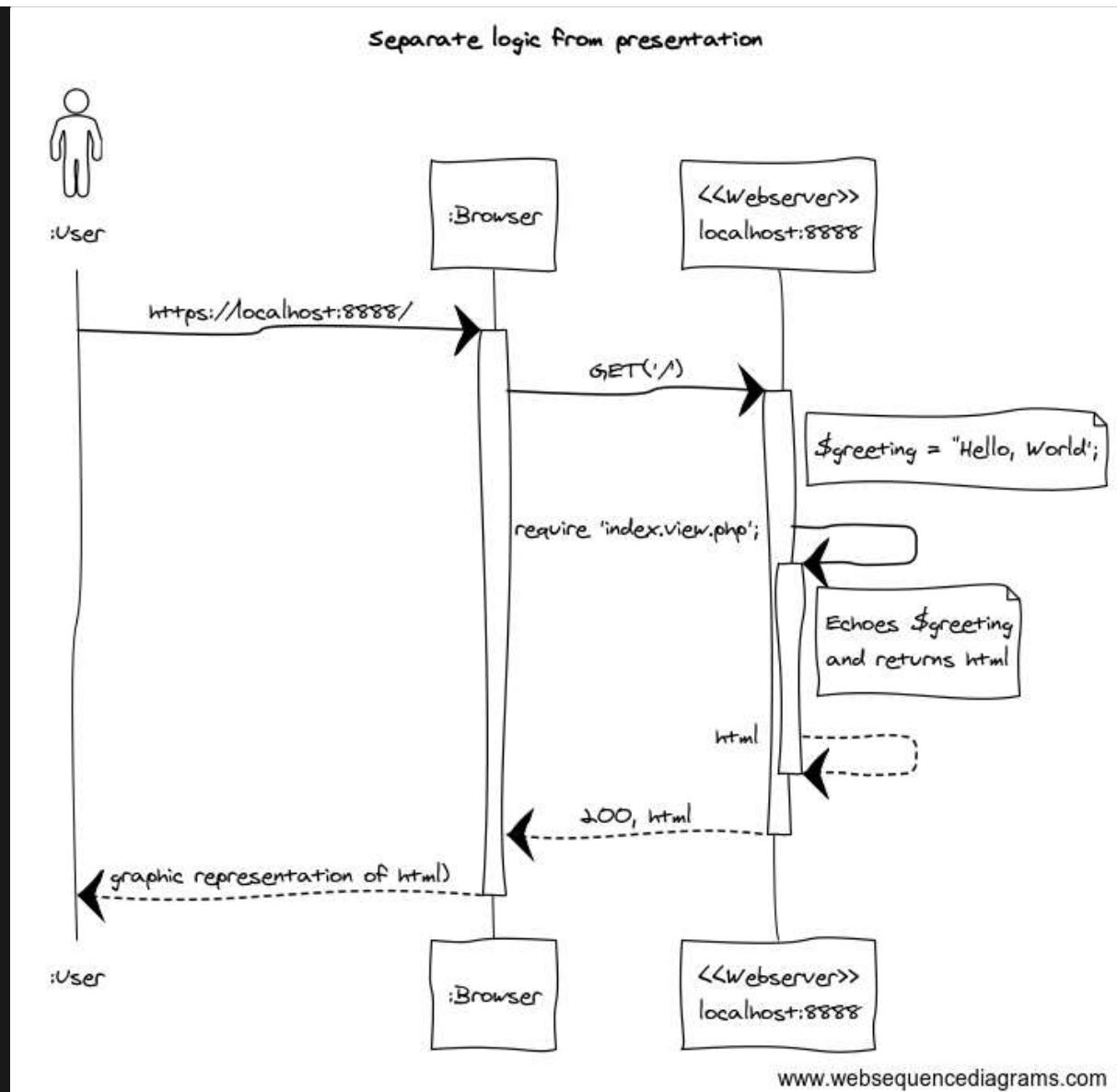
- a webserver always tries to fetch a file from the documentroot.
- a browser always renders the incoming HTML in a response

Omitting these two sequences result in the following model:



## Group objects that are closely related

Objects that are closely related to each other can be combined into one object. Closely related means here that it does not add anything to the subject the model is trying to explain when these objects are modeled separately. In the example, the webserver and `mod_php` objects are closely related. We all know now how they collaborate to create the HTTP response. The next model combines these objects:



## PHP Arrays

Up to now, you already have an understanding of the concept of arrays and you applied them in Javascript and Typescript. You also learned about *loops* to iterate over an array. The next episode explains arrays in PHP and the `foreach` loop.

### Step 6: Understanding Arrays

Let's learn about a new construct: arrays. Think of them as primitive collections of related data. Maybe you have an array of

 <https://laracasts.com/series/php-for-beginners/episodes/6>



Jeffrey introduces *Associative Arrays* in episode 7. This will be a new concept for the most of you. It is important to understand them because they will be used throughout the entire course.

#### Step 7: Associative Arrays

We've learned about basic arrays, but you may also create associative arrays when necessary. The only difference is that

 <https://laracasts.com/series/php-for-beginners/episodes/7>



Do you see the similarities between these associative arrays and Javascript objects? You may also have noticed that the `$_GET` superglobal is an associative array.



**Exercise 0.5** Do the *homework* assignment at the end of the video (the associative array with information about a *Task*). You can experiment with it by trying to display the information in the view.

## More programming concepts

In the next video Jeffrey provides a solution for the homework assignment. He also introduces booleans as a type but this is basically the same as in Javascript.

#### Step 8: Booleans

"Boolean" is a scary, but simple term to understand. It's a data type that represents one of two values: true or false. As you'll

 <https://laracasts.com/series/php-for-beginners/episodes/8>



The next video is about `if` - `else`. This is also pretty much the same as in Javascript

#### Step 9: Conditionals

