



**TRY
PHP**

Level 1

Getting Started

The Basics of PHP



What Will This Course Cover?

Here's what we'll go over in this course.

Level 1

Syntax Basics & Variables

Level 2

Simple Arrays & Associative Arrays

Level 3

Operators & Comparison Statements

Level 4

Looping Constructs



What Do You Need to Know?

Suggested prerequisites:



Basic HTML & CSS

Front-end Foundations & Front-end Formations



Why PHP, Why Now?

PHP is a server-side scripting language that has been around since 1997 and has grown into a modern and performant tool for building websites and applications.

Allows execution of code inline with our HTML markup

Simple reading and processing of files and images

Request and response processing with forms

High performance, scales easily

Let's get started!



Starting From Scratch

Example of a simple HTML file:

index.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

Output

Hello World

Renaming Our File

Let's change the file from .html to .php so it can be processed by the server.

index.php

PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

PHP files can render HTML as well as PHP!

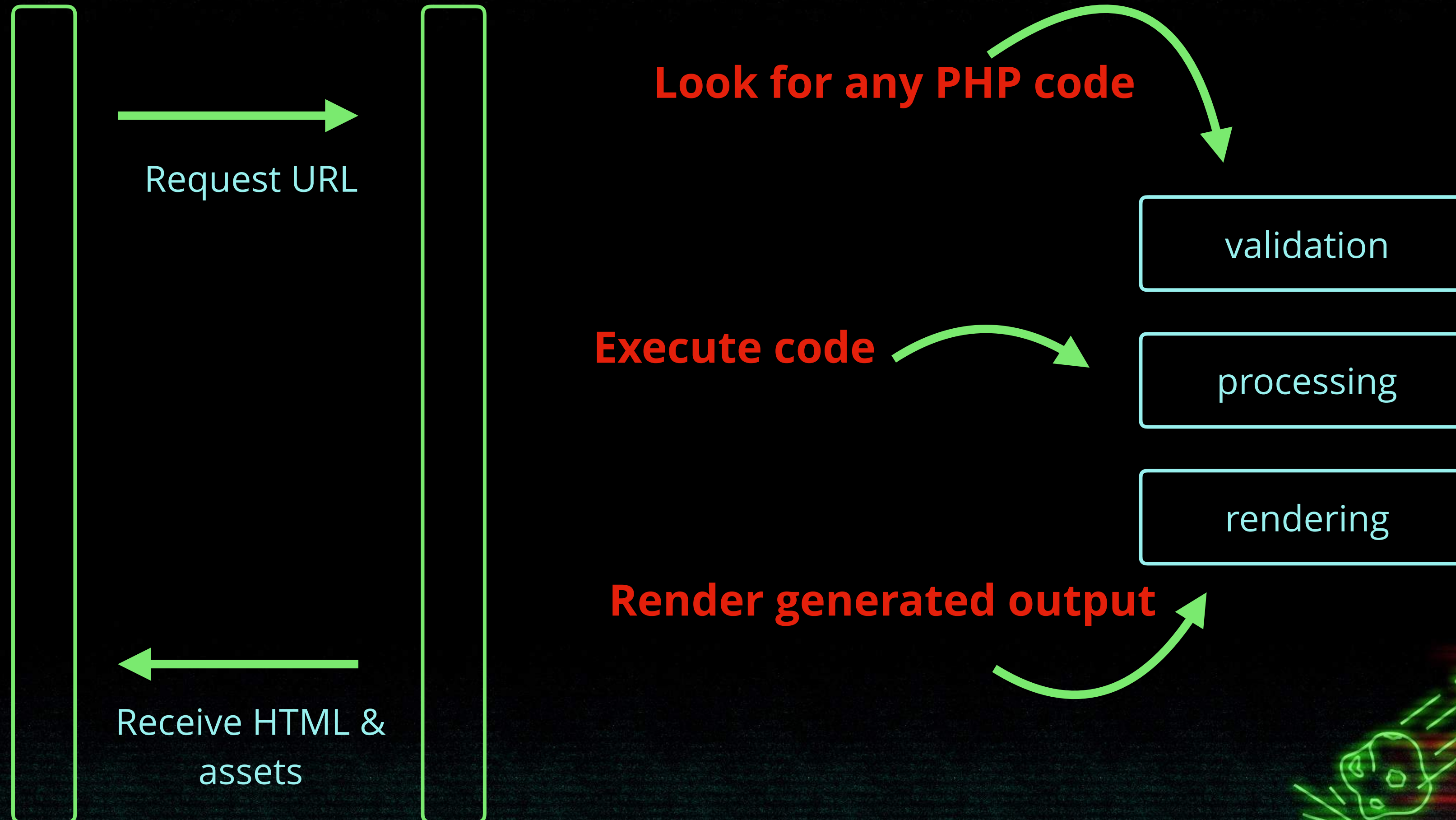
Output

Hello World

What Is Different Now?

Web Browser

Web Server



**TRY
PHP**

Creating a Code Block

index.php

PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <p><?php ?></p>
  </body>
</html>
```

*PHP code is written
between these symbols*

Output

*Let's write some code that will output
something so we can see it here!*

Our First PHP Code

index.php

PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <p><?php echo 'Hello World'; ?></p>
  </body>
</html>
```

*PHP statements end in
semicolons*



Output

Hello World

*echo outputs whatever
comes after it*



Variables and Data Subject to Change

index.php

PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <p><?php $name = 'Hoba'; ?></p>
  </body>
</html>
```

*Variables in PHP always
start with a \$*

Output



*Notice that the data in the variable
isn't printed out automatically*

Outputting Data That's Stored in Variables

index.php

PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <p>
      <?php
        $name = 'Hoba';
        echo $name;
      ?>
    </p>
  </body>
</html>
```

*Echoing the variable outputs
the data inside of it*

Output

Hoba

Variable Naming Conventions

Variables must always begin with a \$ followed by a letter.



`$name`



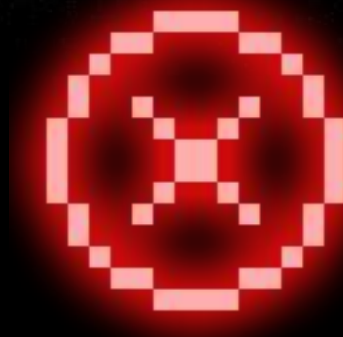
`$_age`



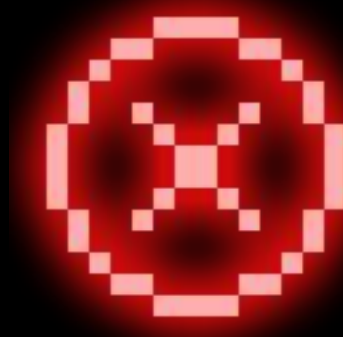
`$full_name`



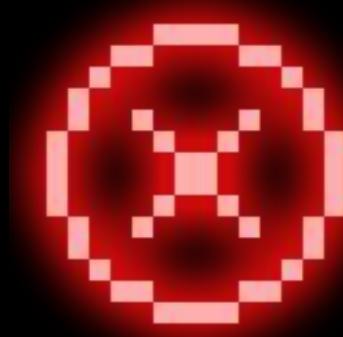
`$war_1984`



`$10_best_targets`



`$so-very-invalid`



`stalemate`



PHP Code Can Go Anywhere

index.php

PHP

```
<?php $name = 'Hoba'; ?>
```

← *Assign the variable*

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title></title>
```

```
</head>
```

```
<body>
```

```
<p>
```

```
<?php echo $name; ?>
```

← *Use the variable*

```
</p>
```

```
</body>
```

```
</html>
```

Output

Hoba

Variables Can Be Used in Multiple Locations

index.php

PHP

```
<?php $name = 'Hoba'; ?>
<!DOCTYPE html>
<html>
  <head>
    <title><?php echo $name; ?></title>
  </head>
  <body>
    <p>
      <?php echo $name; ?>
    </p>
  </body>
</html>
```

Same variable used twice

Output

Hoba

What Have We Learned?

Let's have a quick review.

- Syntax basics
- Code blocks
- PHP request and response server workflow
- Variables and naming rules
- The `echo` statement

Now we can move on to some challenges!





**TRY
PHP**