

Dynamische Websites

Arbeiten mit PHP

blanq

Statische Websites

Vergleich statischer & dynamischer Websites

- ▶ Statische Inhalte (HTML, CSS, JS)
- ▶ Einzelne Dokumente
- ▶ Zugriff eins-zu-eins auf das Dokument

Vorteile

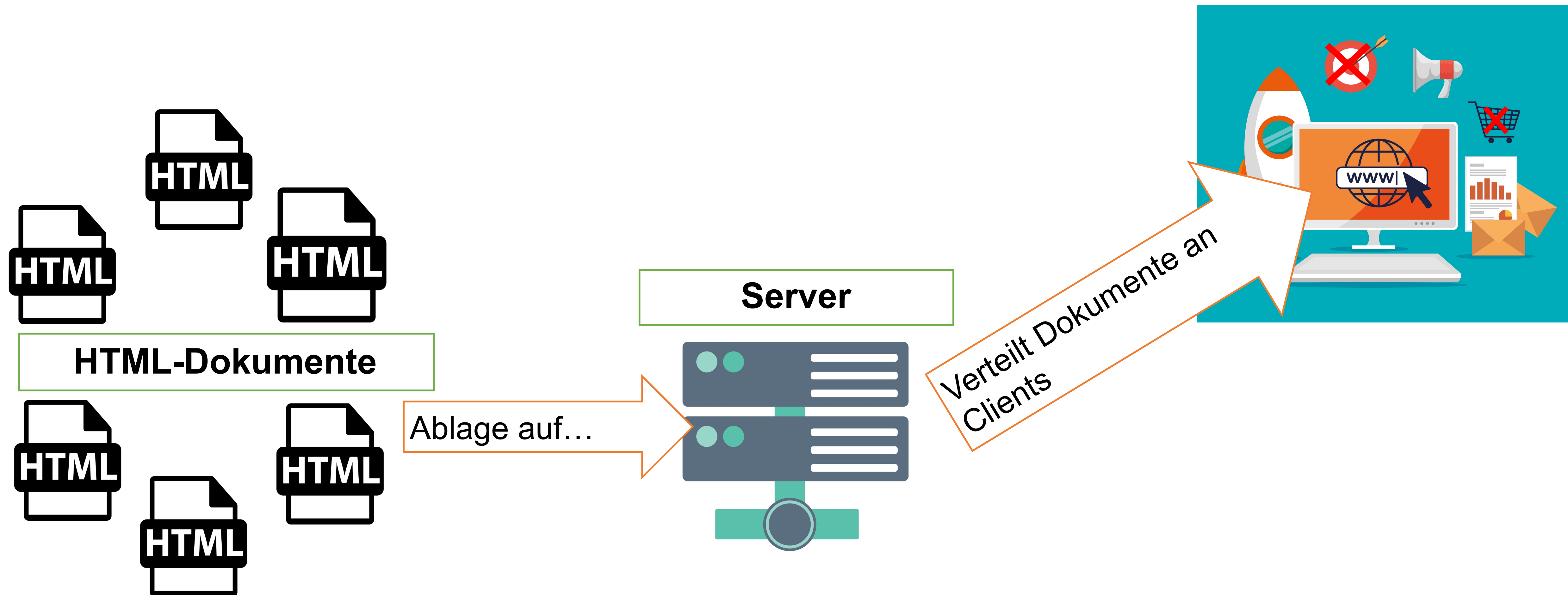
- ▶ Gute Performance
- ▶ Niedrige Anforderungen (Server u. ä.)

Nachteile

- ▶ Bearbeitung nur über Code
- ▶ Keine Datenbank-Verbindung möglich

Ablauf bei statischen Websites

Vergleich statischer & dynamischer Websites



Dynamische Websites

Vergleich statischer & dynamischer Websites

- Kommunikation mit Server
- CMS/Shop-Systeme
- Einfache Features, z. B. Formulare
- Vorstufe von HTML

Vorteile

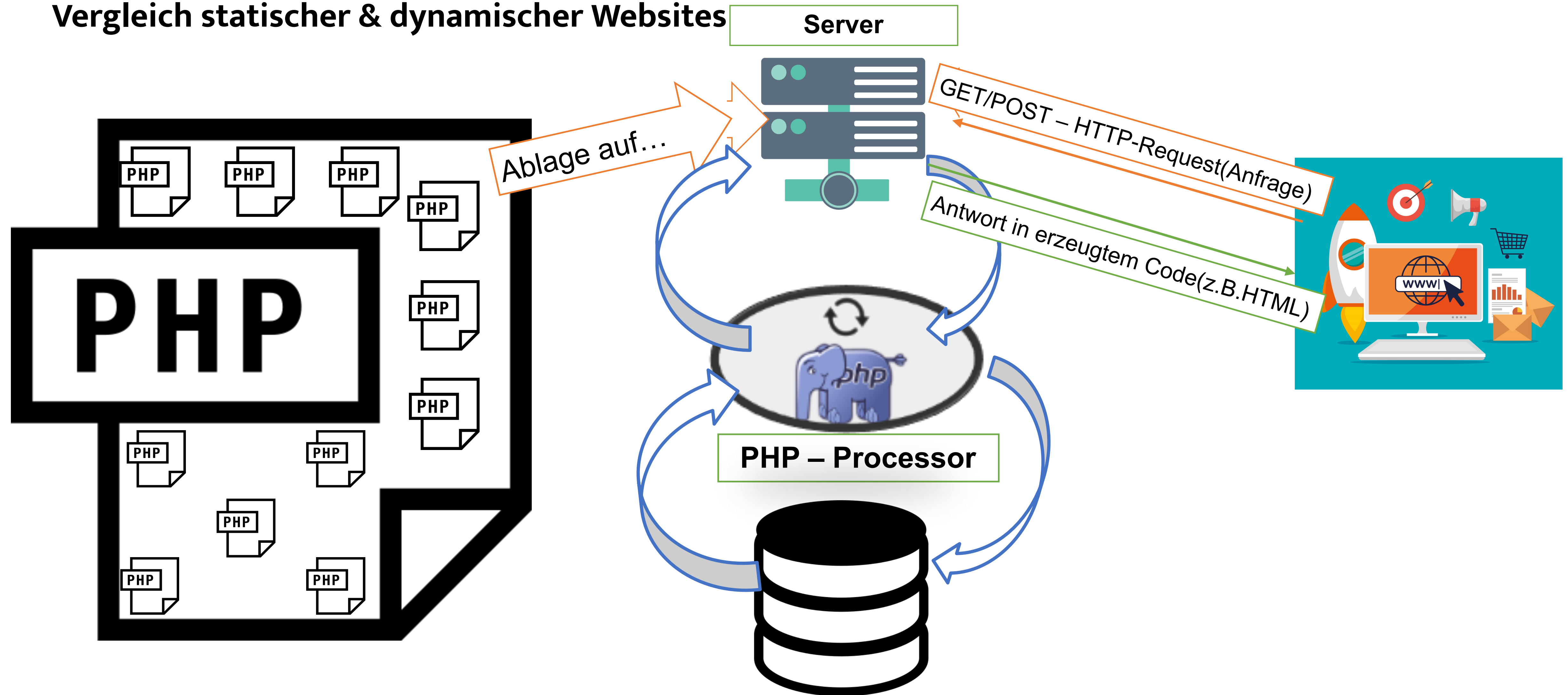
- Anbindung an Datenbank möglich
- Logik lässt sich implementieren

Nachteile

- PHP-Kenntnisse
- Höhere Anforderungen

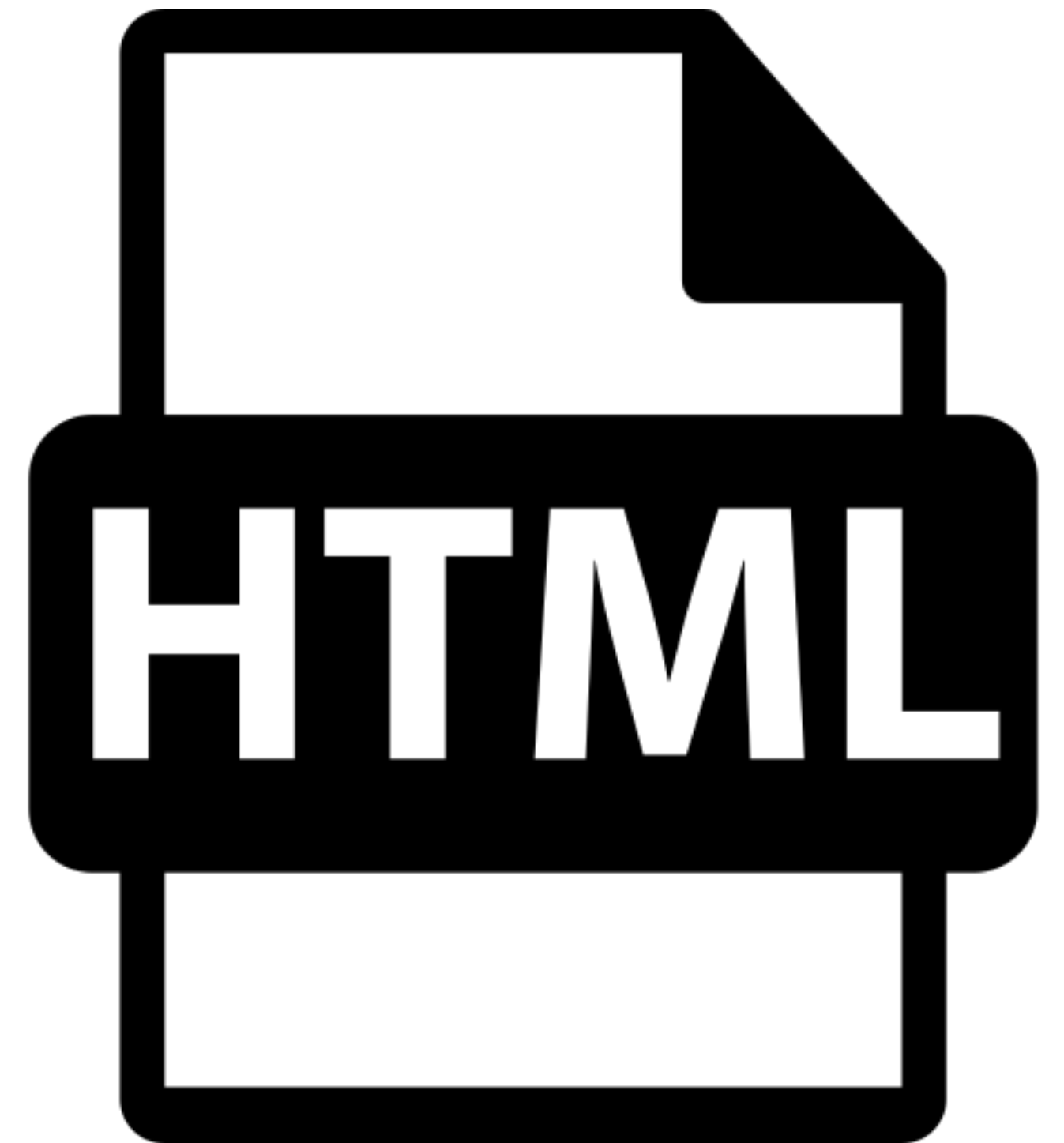
Ablauf bei dynamischen Seiten

Vergleich statischer & dynamischer Websites



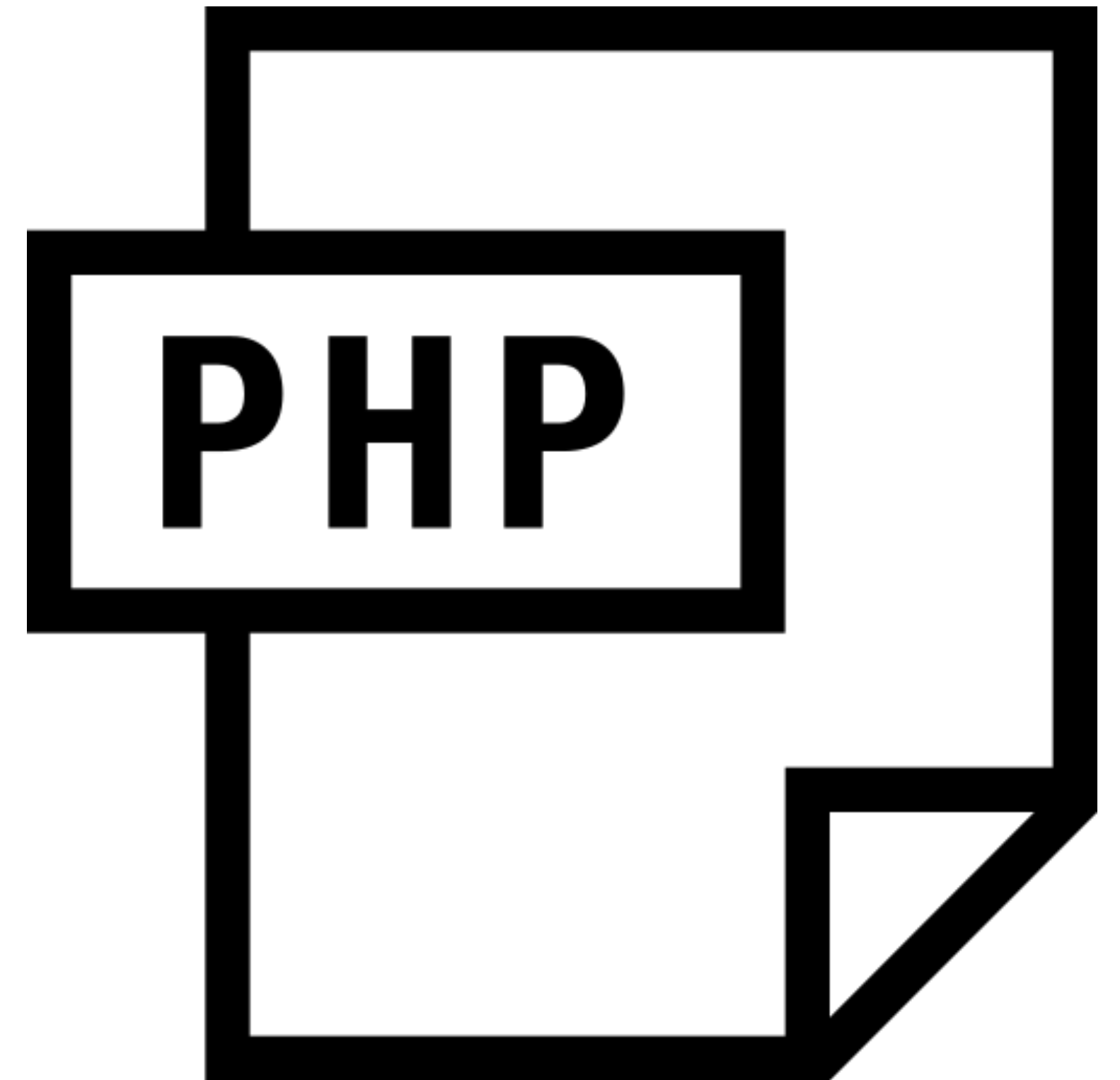
Beispiel HTML-Seite

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <header>
    Der Header der Seite
  </header>
  <main>
    <div class="container">
      Hier ist der Hauptcontent der Seite.
    </div>
  </main>
  <footer>
    ...
  </footer>
</body>
</html>
```



Beispiel PHP-Seite

```
<?php
$title = 'Startseite';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title><?php echo $title; ?></title>
</head>
<body>
<?php
    if ($title === 'Startseite') {
        echo '<header>
            Der Header der Startseite
        </header>';
    } else {
        echo '<header>
            Der Header einer Unterseite
        </header>';
    }
?>
```



Aufgabe 1

1. Installiere MAMP/XAMPP
2. Erstelle eine leere Datei mit dem Namen index.php
3. Fülle die Datei mit dem Standard-HTML5-Gerüst
4. Lege die Datei in den Web-Ordner von MAMP/XAMPP (htdocs o. ä.)
5. Starte MAMP/XAMPP
6. Rufe die Seite über einen Browser auf

PHP-Kennzeichnung

```
<?php
    if ($title === 'Startseite')
        echo '<header>
            Der Header der
        </header>';
    } else {
        echo '<header>
            Der Header ei
        </header>';
    }
?>
```

} PHP-Code

Aufgabe 2

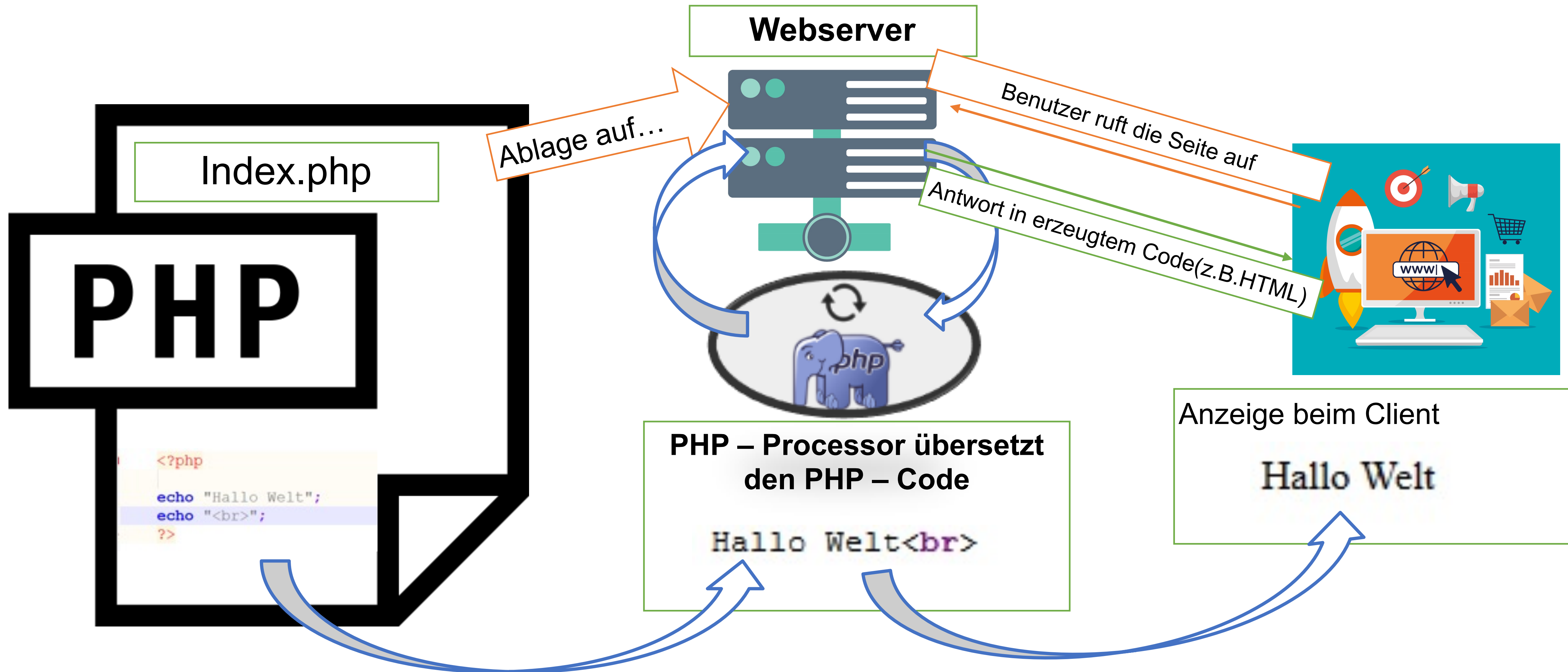
1. Trage `phpinfo();` in die `index.php` ein
2. Aktualisiere die Seite im Browser

Code Syntax

```
<?php  
echo 'Hallo Welt';  
echo '<br>';  
?>
```

- ▶ **echo** => Anweisung/Befehl
- ▶ **'Hallo Welt'** => String
- ▶ **;** => Beendet eine Anweisung/Befehl

Ablauf des Code-Beispiels



Datentypen der einer Variablen

PHP-Variablen

- ▶ Integer
- ▶ Float
- ▶ String
- ▶ Boolean
- ▶ Array
- ▶ **Keine Typsicherheit**, Variablen lassen sich mit anderen Datentypen überschreiben

Aufbau einer Variablen

PHP-Variablen

```
<?php  
$title = 'Startseite';  
?>
```

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

- ▶ \$ => Kennzeichnung einer Variablen
- ▶ title => Bezeichnung der Variablen (Name)
- ▶ = => Zuweisung
- ▶ 'Startseite' => Wert der Variablen

Ein- & Mehrzeiliges Kommentar

PHP-Kommentare

```
/**  
 * Hier kann ein  
 * mehrzeiliges  
 * Kommentar stehen  
 */
```

- ▶ */** => Start des Kommentars
- ▶ Hier kann ... => Text des Kommentars
- ▶ **/* => Zuweisung

```
// Hier steht ein einzeliges Kommentar
```

- ▶ *//* => Start des Kommentars
- ▶ Hier steht ... => Text des Kommentars

Aufgabe 3

1. `phpinfo();` aus der `index.php` entfernen
2. Eine einfache Ausgabe (`echo`) in die `index.php` einfügen
3. Aktualisiere die Seite im Browser
4. Lege eine Variable in der `index.php` an und weise ihr einen String zu
 - Lasse die Variable ausgeben
 - Gebe einen beschreibenden Kommentar an
5. Benutze bei der Ausgabe einen HTML-Tag, bspw. `
`/`` o. ä.
 - Aktualisiere die Seite im Browser

**Vielen Dank für eure
Aufmerksamkeit!**