

WEBPROGRAMMIERUNG

WWI218

7. Programmiersprachen für die Webentwicklung

AGENDA

1. Grundlagen technische Kommunikation über Netzwerke
2. Client/Server-Prinzip – Webbrowser
3. Grundlagen Webentwicklung
4. Serverumgebungen für Webseiten
5. Entwicklungsumgebungen für Webprogrammierung
6. Formale Darstellungssprachen zur Entwicklung von Webseiten/Anwendung
- 7. Programmiersprachen für die Webentwicklung**
8. Datenhaltung und Datenbanken
9. Debugging und Testing
10. Responsive Webseiten und mobile Anwendungen
11. Hybride Entwicklung: Webapps
12. Open Source Internet-Anwendungen und Anwendungsgebiete

WIEDERHOLUNG

- Warum verwendet man Markup Languages?
- Welche ML sind am verbreitetsten?

PROGRAMMIERSPRACHEN

Ziele Webprogrammierung

- Erfüllung von nicht-statischen Aufgaben
- Berechnungen durchführen
- Abbildung Geschäftslogik
- Bereitstellung Funktionen
- Umsetzung Kundenwünsche/Anforderungen
- Schaffen von Schnittstellen/Integration

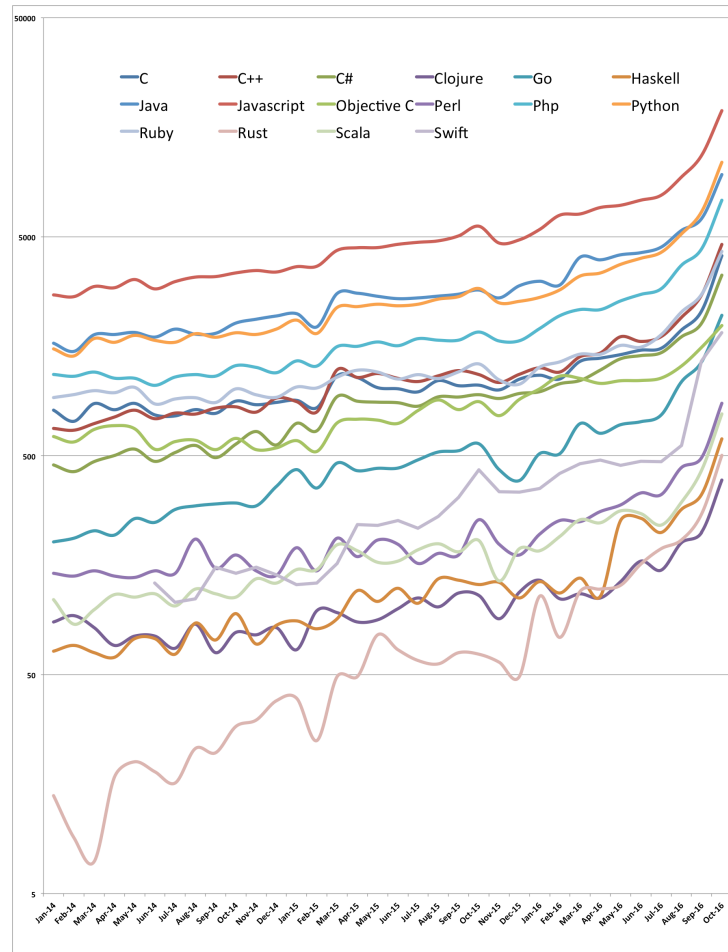
PROGRAMMIERSPRACHEN

Übersicht server- und clientseitige Programmierung

Client	Server
JavaScript	PHP
Java-Applets	ASP
Flash	Perl
ActiveX	Python
	C/C++

PROGRAMMIERSPRACHEN

Verbreitung Programmiersprachen



<https://jaxenter.de/programmiersprachen-rankings-49399>

PROGRAMMIERSPRACHEN

Clientseitige Entwicklung - JavaScript

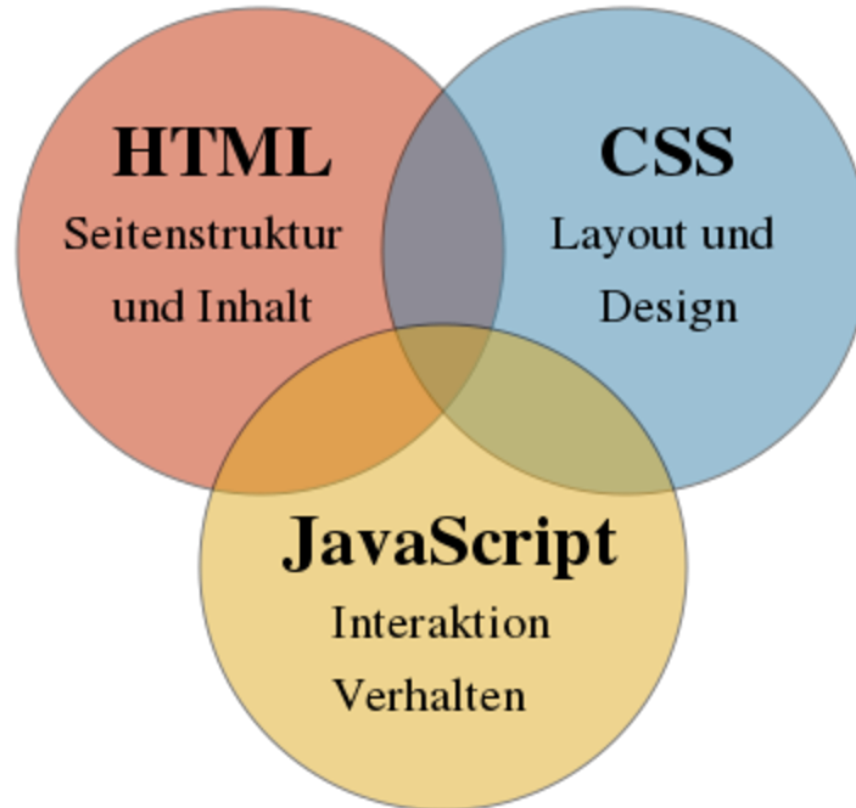
- JavaScript hat sich ursprünglich aus LiveScript weiterentwickelt
- Trotz des Namens hat JavaScript nicht viel mit Java gemeinsam:
 - Java ist eine Programmiersprache
 - Java kann nicht ohne zusätzliche Plugins/Programme ausgeführt werden
- JS ist eine Skriptingsprache
- JS kann direkt im Browser ausgeführt werden



JavaScript™

PROGRAMMIERSPRACHEN

JavaScript Übersicht

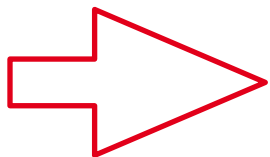


https://wiki.selfhtml.org/wiki/HTML/Tutorials/Trennung_von_Inhalt,_Pr%C3%A4sentation_und_Verhalten

PROGRAMMIERSPRACHEN

Document Object Model

- JS hat Zugriff auf das Browserfenster und das darin angezeigte HTML-Dokument
- Dadurch können dynamisch Inhalte verändert werden
- Das Dokument muss nicht neu vom Server geladen werden



Document Object Model (DOM)

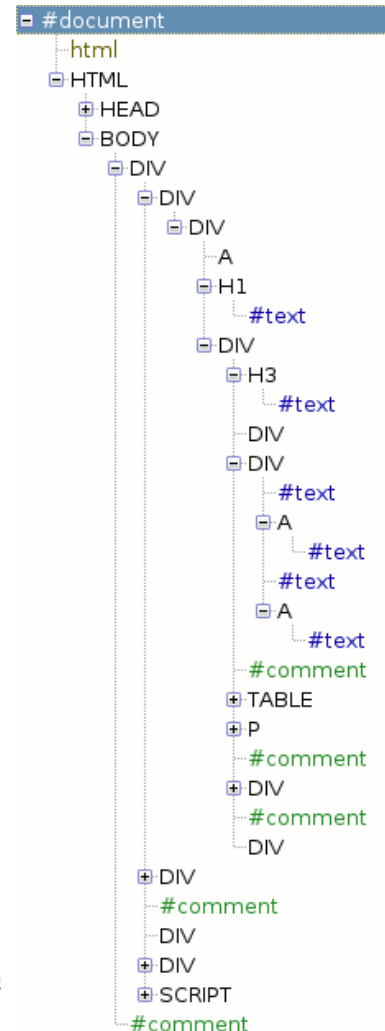
PROGRAMMIERSPRACHEN

Document Object Model

- DOM ist eine Spezifikation einer Programmierschnittstelle, kein Modell
- HTML/XML als Baumstruktur, jeder Knoten repräsentiert ein Objekt (z.B. Überschrift, Text...)
- Das Ziel ist die Manipulation des DOM zur Laufzeit

PROGRAMMIERSPRACHEN

Document Object Model



https://de.wikipedia.org/wiki/Document_Object_Model#/media/File:Dom-inspector-baumstruktur.png

PROGRAMMIERSPRACHEN

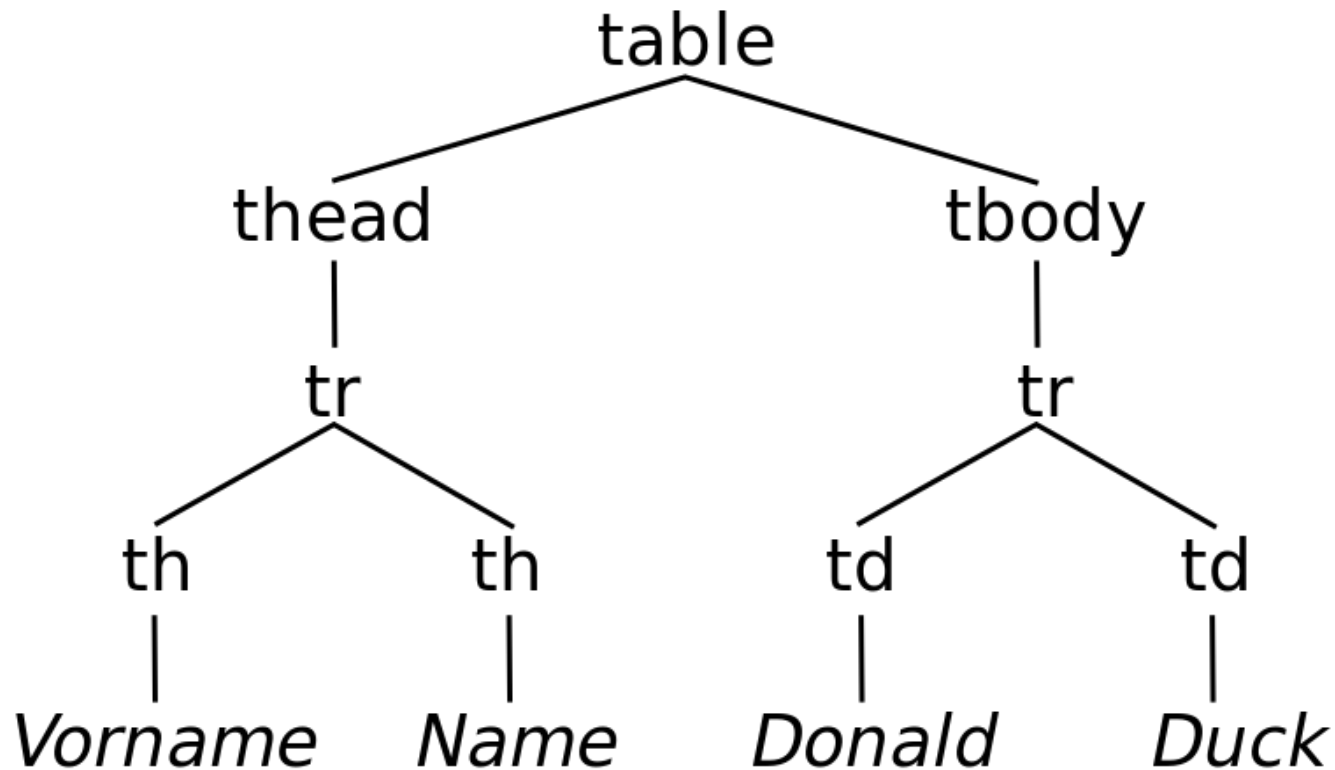
Document Object Model

```
<table>
  <thead>
    <tr>
      <th>Vorname</th>
      <th>Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Donald</td>
      <td>Duck</td>
    </tr>
  </tbody>
</table>
```

https://de.wikipedia.org/wiki/Document_Object_Model

PROGRAMMIERSPRACHEN

Document Object Model



https://de.wikipedia.org/wiki/Document_Object_Model

PROGRAMMIERSPRACHEN

Document Object Model

- Mithilfe des DOM
 - kann zwischen einzelnen Knoten des Dokuments navigiert werden (Springen...)
 - können Knoten erzeugt, verschoben und gelöscht werden
 - können Textinhalte ausgelesen, geändert und gelöscht werden
- Das DOM wurde in 4 Levels standardisiert (Level 0-4)

PROGRAMMIERSPRACHEN

JavaScript

- Einbinden/Ausführen von JavaScript in eine HTML-Seite direkt in HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>JavaScript: Hallo Welt</title>
    <script>
      alert("Hallo Welt!");
    </script>
    <noscript>
      Sie haben JavaScript deaktiviert.
    </noscript>
  </head>
  <body>
    <p>Diese Seite tut nichts weiter, als eine Hinweisbox auszugeben.</p>
  </body>
</html>
```

PROGRAMMIERSPRACHEN

JavaScript

- Einbinden/Ausführen von JavaScript in eine HTML-Seite extern 1/2

quadrat.js

```
function Quadrat() {  
    var Eingabe = document.getElementById('Eingabe');  
    var Ergebnis = Eingabe.value * Eingabe.value;  
    alert("Das Quadrat von " + Eingabe.value + " = " + Ergebnis);  
    Eingabe.value = 0;  
}  
  
var los = document.getElementById('los');  
los.addEventListener('click', Quadrat, true);
```


PROGRAMMIERSPRACHEN

JavaScript

- Einbinden/Ausführen von JavaScript in eine HTML-Seite extern 2/2

index.html

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <title>externes JavaScript in HTML einbinden</title>
</head>
<body>
  <h1>externes JavaScript in HTML einbinden</h1>

  <main>
    <input type="number" id="Eingabe" value="0" size="3">
    <button type="button" id="los">Quadrat errechnen</button>
  </main>
  <script src="quadrat.js"></script>
</body>
</html>
```

PROGRAMMIERSPRACHEN

JavaScript - Variablen - IFs - Switch

```
1 'use strict';
2 var name = '';
3 var alter = '';
4 var eingabe = '';
5 var text = '';
6
7 name = prompt('Bitte geben Sie Ihren Namen ein!', name);
8 alter = prompt('Bitte geben Sie Ihr Alter ein!', alter);
9
10 if (isNaN(alter)) {
11     eingabe = 'dumm';
12 }
13 else if (alter < 18) {
14     eingabe = 'jung';
15 } else {
16     eingabe = 'ok';
17 }
18
19 switch (eingabe) {
20     case 'jung':
21         text = 'Du bist leider zu jung!';
22         break;
23     case 'dumm':
24         text = 'Sie sind leider zu dumm, eine Zahl einzugeben!';
25         break;
26     default:
27         text = 'Hallo, ' + name;
28         break;
29 }
30
31 alert(text);
```

PROGRAMMIERSPRACHEN

JavaScript - While/For

```
var zahl,
    text;

zahl = prompt('Bitte geben Sie eine Zahl ein!', zahl);

while (zahl >= 1) {
    text = zahl + '- 1 = ' + (zahl-1) + '<br>';
    zahl = zahl -1;
    document.write(text);
}
```

```
var zahl,
    text;

confirm('Sind Sie bereit?');

for (var i = 1; i <= 10; i++) {
    zahl = i * i;
    text = zahl + ' = ' + i + ' * ' + i + '<br>';
    document.write(text);
}
```

PROGRAMMIERSPRACHEN

JavaScript - Funktionen

```
<script>
'use strict';
var eingabe,
    ergebnis,
    gerundet,
    text;

//Helferfunktionen
function wurzelZiehen(zahl) {
    ergebnis = Math.sqrt(zahl);
    runden(ergebnis);
    return gerundet;
}

function runden(zahl) {
    gerundet = zahl.toFixed(2);
    return gerundet;
}

eingabe = prompt('Bitte geben Sie eine Zahl ein!', eingabe);
eingabe = parseFloat(eingabe);
if (isNaN(eingabe)) {
    alert('Der eingegebene Wert: ' + eingabe + ' ist keine Zahl!');
}

wurzelZiehen(eingabe);
text = 'Die Wurzel von ' + eingabe + ' ist ' + ergebnis + ', gerundet ' + gerundet + '.';
document.write(text);
</script>
```

PROGRAMMIERSPRACHEN

JavaScript - Objekte

```
var data = {  
  id : "17283",  
  firstName : "Hans",  
  name : "Meier",  
  type : "Bestellung",  
  items : [2231, 188832, 231231, 3123, 123]  
}
```

```
var data = new Object();  
  
data.id = "17283";  
data.firstName = "Hans";  
data.name = "Meier";  
data.type = "Bestellung";  
data.items = [2231, 188832, 231231, 3123, 123];
```

PROGRAMMIERSPRACHEN

JavaScript - DOM Manipulation

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

AUFGABEN

JavaScript

- Arbeiten Sie sich durch das Tutorial auf <http://www.jshero.net/home.html>
- Erstellen Sie mit JavaScript ein Dropdown-Menü für Ihre Webseite

PROGRAMMIERSPRACHEN

Serverseitige Programmierung - PHP

- PHP ist ein Akronym PHP : Hypertext Preprocessor
- In PHP-Dateien können vorkommen
 - HTML
 - CSS
 - JavaScript
 - PHP



PROGRAMMIERSPRACHEN

Serverseitige Programmierung - PHP

- Erzeugung dynamischer Content
- Dateioperationen auf dem Server
- Verarbeitung Formulareingaben
- Verarbeitung Cookies
- Datenbankoperationen
- Benutzerverwaltung
- Sicherheitsaspekte

PROGRAMMIERSPRACHEN

Warum PHP?

- PHP funktioniert auf den unterschiedlichsten Plattformen/OS (Windows, Linux, MacOS...)
- Kompatibel mit allen gängigen Servern (Apache, IIS...)
- Unterstützt die meisten Datenbanksysteme
- Kostenlos
- Gut erlernbar, trotzdem umfangreich
- Effiziente Verarbeitung auf dem Server

PROGRAMMIERSPRACHEN

PHP

- Einbinden/Ausführen von PHP innerhalb einer HTML-Seite

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

PROGRAMMIERSPRACHEN

PHP

- Einbinden/Ausführen von PHP extern 1/2

header.php

```
<header>
  
  <h1>Titel</h1>
  <nav>
    <ul>
      <li><a href="#link_1.html">Wiki</a></li>
      <li><a href="#link_2.html">Blog</a></li>
      <li><a href="#link_3.html">Forum</a></li>
    </ul>
  </nav>
</header>
```

PROGRAMMIERSPRACHEN

PHP

- Einbinden/Ausführen von PHP extern 2/2

index.php

```
<!doctype html>
<head>
  <meta charset="utf-8">
  <title>Meine erste HTML5-Seite</title>
</head>

<body>
  <?php include ("header.php"); ?>

  <main>
    <article>
      <h1>Überschrift</h1>
      <p>Dies ist meine erste HTML5-Seite</p>
      ... mehr Inhalt
    </article>

    <aside>
      <h2>Weiterführende Links</h2>
      <ul>
        <li><a href="#link_1.html">Wiki</a></li>
        <li><a href="#link_2.html">Blog</a></li>
        <li><a href="#link_3.html">Forum</a></li>
      </ul>
    </aside>
  </main>

  <?php include ("footer.php"); ?>
</body>
</html>
```

PROGRAMMIERSPRACHEN

PHP Variablen

- Datentyp muss nicht definiert werden
- Verschiedene Scopes: lokal, global, statisch

```
<?php
$x = 5;
$y = 10;

function myTest() {
    global $x, $y;
    $y = $x + $y;
}

myTest();
echo $y; // outputs 15
?>
```

PROGRAMMIERSPRACHEN

PHP Variablen

```
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}
```

```
myTest();
myTest();
myTest();
?>
```

PROGRAMMIERSPRACHEN

PHP Variablen

- String: `$x = "Hello world!";`
- Integer: `$x = 5985;`
- Array: `$cars = array(„Volvo“, "BMW", "Toyota");`
- Float: `$x = 10.365;`
- Boolean: `$x = true;`
- Object:

```
<?php
class Car {
    function Car() {
        $this->model = "VW";
    }
}

// create an object
$herbie = new Car();

// show object properties
echo $herbie->model;

?>
```


PROGRAMMIERSPRACHEN

PHP Variablen Stringoperationen

- **Suche:**

```
echo strpos("Hello world!", "world");
```

```
// outputs 6
```

- **Replace:**

```
echo str_replace("world", "Dolly", "Hello world!");
```

```
// outputs Hello Dolly!
```

PROGRAMMIERSPRACHEN

PHP Arithmetische Operationen

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power (Introduced in PHP 5.6)

PROGRAMMIERSPRACHEN

PHP Zuweisungs-Operationen

Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulus

PROGRAMMIERSPRACHEN

PHP Vergleichs-Operationen

Operator	Name	Example	Result
==	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y
===	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type
!=	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y
<>	Not equal	<code>\$x <> \$y</code>	Returns true if \$x is not equal to \$y
!==	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y
<	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y

PROGRAMMIERSPRACHEN

PHP Logische Operationen

Operator	Name	Example	Result
and	And	\$x and \$y	True if both \$x and \$y are true
or	Or	\$x or \$y	True if either \$x or \$y is true
xor	Xor	\$x xor \$y	True if either \$x or \$y is true, but not both
&&	And	\$x && \$y	True if both \$x and \$y are true
	Or	\$x \$y	True if either \$x or \$y is true
!	Not	!\$x	True if \$x is not true

PROGRAMMIERSPRACHEN

PHP IF-ELSE

```
<?php
$t = date("H");

if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

PROGRAMMIERSPRACHEN

PHP Switch

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

PROGRAMMIERSPRACHEN

PHP While

```
<?php
```

```
$x = 1;
```

```
while($x <= 5) {
```

```
    echo "The number is: $x <br>";
```

```
    $x++;
```

```
}
```

```
?>
```

```
<?php
```

```
$x = 6;
```

```
do {
```

```
    echo "The number is: $x <br>";
```

```
    $x++;
```

```
} while ($x <= 5);
```

```
?>
```


PROGRAMMIERSPRACHEN

PHP For Loops

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
?>
```

```
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```

PROGRAMMIERSPRACHEN

PHP Funktionen

```
<?php
function writeMsg() {
    echo "Hello world!";
}

writeMsg(); // call the function
?>
```

```
<?php
function sum($x, $y) {
    $z = $x + $y;
    return $z;
}

echo "5 + 10 = " . sum(5, 10) . "<br>";
echo "7 + 13 = " . sum(7, 13) . "<br>";
echo "2 + 4 = " . sum(2, 4);
?>
```

PROGRAMMIERSPRACHEN

PHP Arrays

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
$arlength = count($cars);

for($x = 0; $x < $arlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```

PROGRAMMIERSPRACHEN

PHP Arrays Sort

- `sort()` - sort arrays in ascending order
- `rsort()` - sort arrays in descending order
- `asort()` - sort associative arrays in ascending order, according to the value
- `ksort()` - sort associative arrays in ascending order, according to the key
- `arsort()` - sort associative arrays in descending order, according to the value
- `krsort()` - sort associative arrays in descending order, according to the key

PROGRAMMIERSPRACHEN

PHP Formular

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

AUFGABEN

- Machen Sie das Quiz auf https://www.w3schools.com/php/php_quiz.asp
- Testen Sie einige der Beispiele von https://www.w3schools.com/php/php_examples.asp auf Ihrem Webserver
- Erstellen Sie mit PHP ein Formular zur Eingabe von Daten. Das kann z.B. ein Kontaktformular sein.
https://www.w3schools.com/php/php_forms.asp

PRÄSENTATION 1. ZWISCHENERGEBNIS

- Präsentieren Ihr Design bzw. Konzeption der Webanwendung
- Zeigen Sie Ihre HTML-Struktur in Form einer Live-Demo

LINKTIPPS

- <https://wiki.selfhtml.org/wiki/JavaScript>
- https://de.wikipedia.org/wiki/Document_Object_Model
- https://wiki.selfhtml.org/wiki/JavaScript/Tutorials/Grundlagen_der_Programmierung
- <http://www.jshero.net/home.html>
- https://www.w3schools.com/howto/howto_js_dropdown.asp
- <https://www.w3schools.com/>
- https://www.w3schools.com/php/php_forms.asp