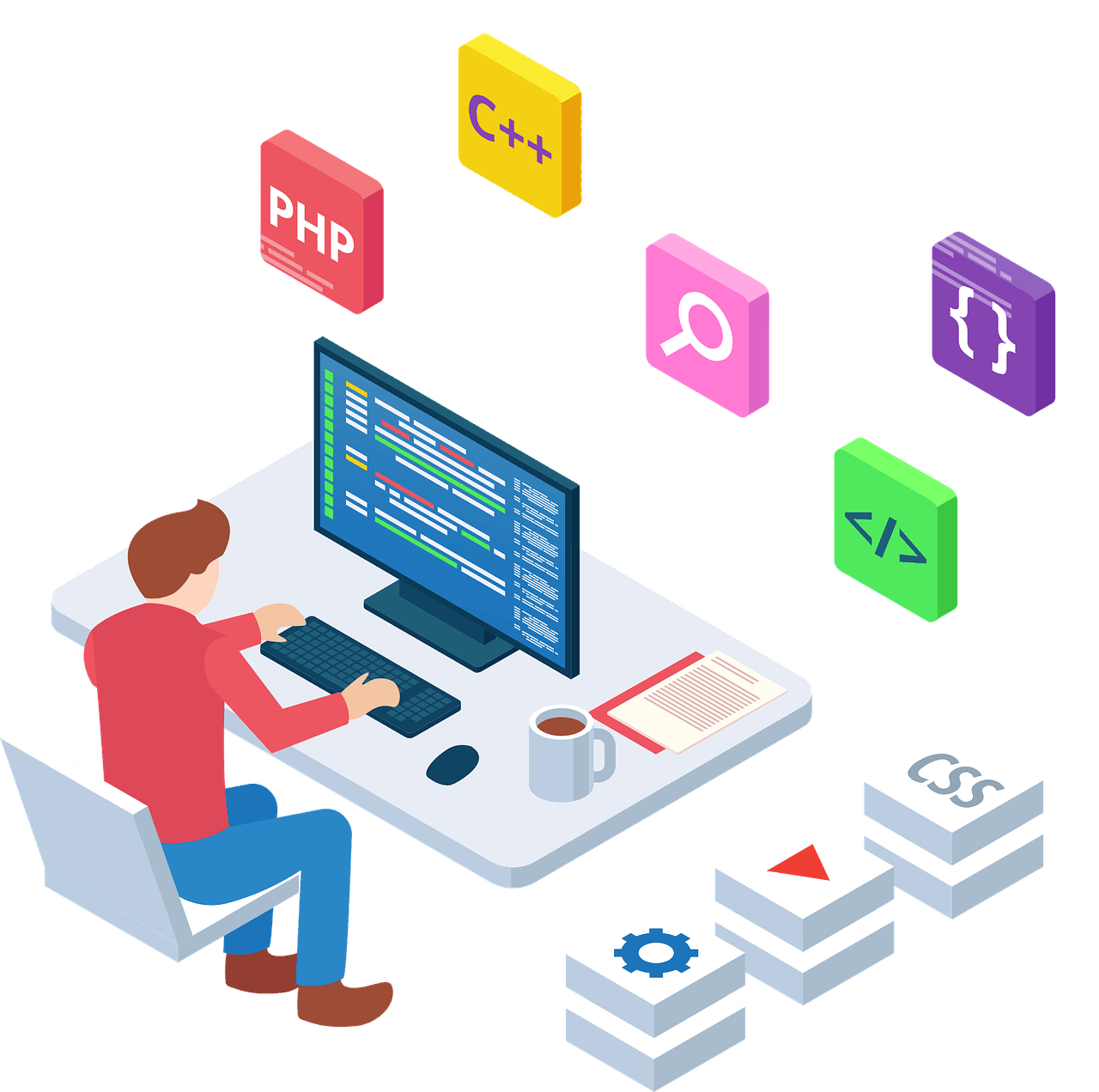
**Webentwicklung**



Artem Galan

MINT 24/25

# Einleitung

Die Webentwicklung unterscheidet sich in gewissen Punkten erheblich von der normalen Applikationsentwicklung. Der Hauptunterschied liegt darin, dass die Abläufe über mindestens zwei Orte verteilt werden. Klassischerweise ist dies der Server (auf welchem die eigentliche Applikation läuft) und der Client (der Computer des Benutzers), wo die Ein- uns Ausgaben stattfinden.

Dazu kommt noch, dass in den meisten Fällen HTML verwendet wird, um ein GUI (graphical user interface) zu erstellen. Aus diesem Grund beginnen wir mit HTML

# HTML Allgemein

HTML steht ausgeschrieben für Hypertext Markup Language und ist keine Programmiersprache. Es gibt weder die Möglichkeit Entscheidungen zu fällen (z.B. mit if then) noch werden Daten direkt verarbeitet (z.B. Rechnungen durchzuführen).

HTML ist eine Auszeichnungssprache und sagt dem Browser (z.B. Firefox), wie er Inhalte auf dem Bildschirm darstellen soll. So gesehen ist HTML eine äusserst simple Sache. Nichtsdestotrotz ist es die Grundlage für die meisten Web-Applikationen.

|  |  |  |
| --- | --- | --- |
| **Computersprachen** | | |
| **Programmiersprachen** | **Datenbanksprachen** | **Auszeichnungssprachen**  **(Engl.: markup language)** |
| * C * C++ * Java   … | * SQL   … | * HTML * LaTeX   … |

Der Begriff Hypertext wurde 1965 von Ted Nelson geprägt. Das Konzept für Hypertext entstand und wurde vom W3C Gremium (=Fachgruppe von Experten) definiert:

* Hypertext ist ein Text, der nicht linear sein muss.
* Hypertext ist ein Text, der Links (Hyperlinks) zu anderen Texten enthält.
* HyperMedia ist ein Hypertext, der auch Grafiken, Videos oder Klänge enthalten kann.
* Hypertext und HyperMedia sind Konzepte, keine Produkte.

# Die eigene Homepage

Erstellen Sie eine eigene Homepage, welche Sie für Ihre Bewerbungen nutzen können. Sie soll folgende Punkte enthalten:

**Inhaltlich:**

• Titelblatt

• Infos aus dem Lebenslauf

• Ihre Stärken und Vorzüge

• Beispiele ihrer Arbeiten (Bild und Beschreibung)

• Auf Ihrer Seite sollen folgende Dinge mindestens einmal vorkommen

• Eine Seitennavigation

• Es ist ein Layout ersichtlich (z.B. Alle Schriften gleich, Farben haben einen Sinn und Funktion, etc.)

• Bilder

Erweiterte Ziele:

• Ein Effekt wie Accordion oder Collapse aus Bootstrap

**Technisch**:

* Haben Sie eine saubere Ordnerstruktur
* Formatieren Sie ihren Code leserlich
* Halten Sie sich an guten HTML Style
* Überschriften
* Absätze
* Hyperlinks
* Listen
* Tabellen
* Bilder
* Geben Sie ihren HTML Tags sinnvolle Attribute und Classes/ID
* Integrieren Sie ihr eigenes CSS im header

**Erweiterte Ziele:**

* Formatieren Sie ihre Seite mit CSS
  + Grid Layout
  + https://css-tricks.com/snippets/css/complete-guide-grid/
* Integrieren Sie Bootstrap auf Ihrere Seite (oder ein anderes Framework Ihrer Wahl
* Responsive HTML
* HTML Formulare
* Favicon

# Ressourcen

Nutzen Sie folgende Seite zum Lernen:

<http://www.html-seminar.de/>

Bootstrap:

<http://www.w3schools.com/bootstrap/>

Weitere Nachschlagewerke / Referenzen:

HTML:

<https://developer.mozilla.org/en-US/>

<https://www.w3schools.com/html/default.asp>

<https://wiki.selfhtml.org/wiki/HTML/Tutorials>

CSS:

<https://wiki.selfhtml.org/wiki/CSS>

<https://www.w3schools.com/css/default.asp>

**Lernziele**

• Sie kennen das HTML Grundgerüst (DOCTYPE, HTML, HEADER, BODY)

• Sie haben eine saubere Codedarstellung

Sie können folgende HTML Tags und Kenntnisse anwenden

• Formatierung: Überschriften (h1, h2 usw.) fett, unterstrichen, kursiv, hoch und tiefstellen

• Links

• Listen und Aufzählungen

• Tabellen

• Bilder

• Klassen und IDs verwenden

• CSS allgemein

* Einbinden (extern, intern, internal)
* Selektoren
* Einfache Formatierungen (Färbungen, Schrift, Schriftgrösse, Ausrichtungen)

# Aufbau eines HTML-Dokuments

<!DOCTYPE html>

<html **lang**="...">

<meta **charset**="UTF-8">

<title>Page Title</title>

<meta **name**="viewport" **content**="width=device-width,initial-scale=1"> <link **rel**="stylesheet" **href**="">

<script **src**="">…</script>

<body>  
  
<img **src**="img\_la.jpg" **alt**="LA" **style**="width:100%">  
  
<div **class**="">

<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

</div>  
  
</body>

</html>

## Textformatierungen

Um Texte grob zu formatieren, können HTML-Tags verwendet werden. Diese bieten eine einfache Möglichkeit, Struktur und Stil hinzuzufügen. Hier sind einige häufig genutzte Tags:

**1. Überschriften**

Die Tags <h1> bis <h6> dienen zur Erstellung von Überschriften, wobei <h1> die größte und <h6> die kleinste Schriftgröße hat.

<h1>Das ist eine Hauptüberschrift</h1>

<h2>Das ist eine Unterüberschrift</h2>

<h3>Das ist eine kleinere Unterüberschrift</h3>



**2. Fettgedruckter Text**

Mit <b> oder <strong> wird Text hervorgehoben (fettgedruckt).

<b>Das ist fetter Text</b>

<strong>Das ist ebenfalls fetter Text</strong>



**3. Kursiver Text**

Für kursive Schrift nutzt man <i> oder <em>.

<i>Das ist kursiver Text</i>

<em>Das ist ebenfalls kursiver Text</em>



**4. Unterstrichener Text**

Text wird mit <u> unterstrichen.

<u>Das ist unterstrichener Text</u>



**5. Hochgestellter Text**

Für hochgestellte Zeichen nutzt man <sup>.

x<sup>2</sup> ergibt x²



**6. Tiefgestellter Text**

Mit <sub> erstellt man tiefgestellte Zeichen.

H<sub>2</sub>O ergibt H₂O



**Zusammenfassung in einem Beispiel**

<h1>Das ist eine Hauptüberschrift</h1>

<h2>Das ist eine Unterüberschrift</h2>

<h3>Das ist eine kleinere Unterüberschrift</h3>

<b>Das ist fetter Text</b>

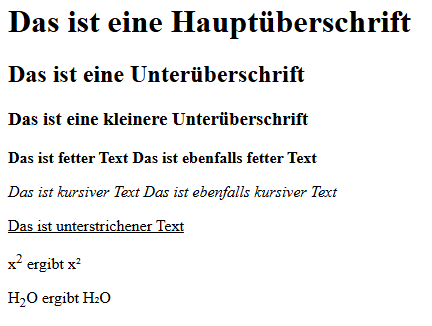
<strong>Das ist ebenfalls fetter Text</strong>

<p><i>Das ist kursiver Text </i><em>Das ist ebenfalls kursiver Text</em></p>

<p><u>Das ist unterstrichener Text</u></p>

<p>x<sup>2</sup> ergibt x²</p>

<p>H<sub>2</sub>O ergibt H₂O</p>



## Links

Links werden mit dem <a>-Tag erstellt. Der grundlegende Aufbau eines Links ist:

<a **href**="URL">Linktext</a>

Hierbei ist href das Attribut, das die Ziel-URL des Links angibt. Der Linktext ist der Text, auf den der Benutzer klicken kann.

**1. Relativer Link auf eine Seite der eigenen Homepage**

Ein relativer Link verweist auf eine Seite innerhalb der gleichen Website. Der Pfad ist relativ zur aktuellen Seite.

Beispiel:

<a **href**="index.html">Kontaktseite</a>



In diesem Fall verweist der Link auf die Datei kontakt.html, die sich im gleichen Verzeichnis wie die aktuelle Seite befindet.

Falls die Datei in einem Unterordner liegt, könnte der relative Link so aussehen:

<a **href**="ueber-uns/geschichte.html">Unsere Geschichte</a>

**2. Absoluter Link auf eine fremde Seite**

Ein absoluter Link verweist auf eine externe Website, die durch die vollständige URL angegeben wird.

Beispiel:

<a **href**="https://extranet.bws-uster.ch">BWS Extranet</a>

In diesem Fall führt der Link zu einer externen Website (z.B. einer fremden Homepage).

**3. Link auf einen Anker innerhalb derselben Seite**

Ein Anker ist ein spezieller Punkt innerhalb einer Seite, zu dem der Benutzer springen kann. Zuerst muss ein Anker gesetzt werden, dann kann darauf verlinkt werden.

Zuerst definieren wir den Anker:

<a **name**="kontakt">Kontaktbereich</a>

Nun können wir auf diesen Anker innerhalb derselben Seite verlinken:

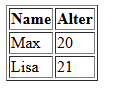
<a **href**="#kontakt">Zum Kontaktbereich</a>

Dies ermöglicht es dem Benutzer, direkt zum "Kontaktbereich" zu springen, ohne die Seite neu zu laden.

## Tabellen

In HTML wird eine Tabelle mit dem <table>-Tag erstellt. Sie besteht aus Zeilen (<tr>) und Zellen, die durch <td> für normale Zellen und <th> für Kopfzellen dargestellt werden.

Hier ist ein einfaches Beispiel:

<table **border**="1">

<tr>

<th>Name</th>

<th>Alter</th>

</tr>

<tr>

<td>Max</td>

<td>20</td>

</tr>

<tr>

<td>Lisa</td>

<td>21</td>

</tr>

</table>

Erklärung:

* <table>: Definiert eine Tabelle.
* <tr>: Eine Tabellenzeile.
* <th>: Eine Kopfzelle, die in der Regel fett und zentriert ist.
* <td>: Eine normale Zelle.
* border="1": Fügt einen Rand um die Tabelle hinzu.

## Listen und Aufzählungen

In HTML gibt es zwei Hauptarten von Listen: unnummerierte (geordnete) und nummerierte (ungeordnete) Listen.

**1. Unnummerierte Liste (geordnete Liste)**

Hier wird die Reihenfolge der Listenelemente nicht festgelegt.

<ul>

<li>Apfel</li>

<li>Banane</li>

<li>Orange</li>

</ul>

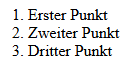
* <ul>: Definiert eine unnummerierte Liste.
* <li>: Ein Listenelement.

**2. Nummerierte Liste (ungeordnete Liste)**

In einer nummerierten Liste wird die Reihenfolge der Elemente angezeigt.

html

Code kopieren

<ol>

<li>Erster Punkt</li>

<li>Zweiter Punkt</li>

<li>Dritter Punkt</li>

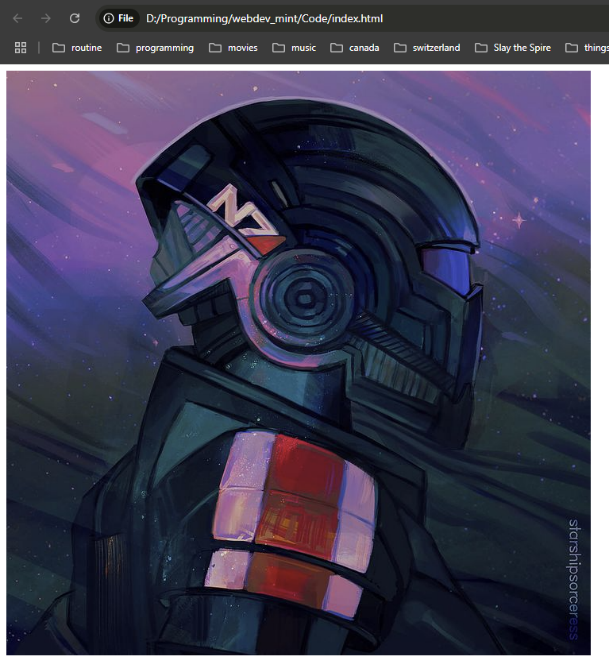
</ol>

* <ol>: Definiert eine nummerierte Liste.
* <li>: Ein Listenelement.

## Bilder

Bilder werden mit dem <img>-Tag eingefügt. Der src-Attribut gibt die Quelle des Bildes an. Für den relativen Pfad wird der Speicherort des Bildes auf der eigenen Website angegeben.

Beispiel für ein Bild mit einer relativen Adresse:

<img **src**="images/mass-effect" **alt**="shepard-drawn">

Erklärung:

* <img>: Das Tag zum Einfügen von Bildern.
* src="bilder/logo.png": Der relative Pfad zum Bild, hier im Ordner bilder und die Datei heißt logo.png.
* alt="Logo der Website": Ein alternativer Text, der angezeigt wird, wenn das Bild nicht geladen werden kann oder für Screenreader zur Barrierefreiheit.

# Formatieren mit CSS

CSS bedeutet Cascading Style Sheets. Es handelt sich um einzelne Dateien oder direkt eingefügter Code im HTML-Dokument.Stylesheets sind für die Gestaltung der Website verantwortlich. Geschichtlich gesehen ist CSS jünger. Sie entstanden aus dem Bedürfnis, Webseiten besser gestalten zu können. Während HTML eigentlich nur die Inhalte und die Struktur der Daten enthält, werden mit CSS die Formatierungen übernommen.

## Styles innerhalb eines Tags

Um Styles direkt innerhalb eines Tags zu definieren, verwendet man das style-Attribut. Dieses Attribut ermöglicht es, CSS-Regeln für ein spezifisches HTML-Element festzulegen.

Beispiel:

<p **style**="color: red; font-size: 16px;">Dies ist ein roter Text mit Schriftgröße 16px.</p>

* style="color: red; font-size: 16px;": Definiert den Text in der Farbe Rot und setzt die Schriftgröße auf 16 Pixel.

Weitere Beispiele für häufig genutzte Style-Eigenschaften:

* background-color: Hintergrundfarbe
* font-family: Schriftart
* text-align: Textausrichtung

## Farben mit Stylesheets (CSS)

In CSS wird die Farbe von Texten und Hintergründen mithilfe der Eigenschaften color und background-color definiert. Diese werden normalerweise in einem externen Stylesheet oder innerhalb eines <style>-Tags im Dokument definiert.

Beispiel:

<style>

p {

color: blue; /\* Textfarbe \*/

background-color: yellow; /\* Hintergrundfarbe \*/

}

</style>

<p>Dies ist ein Text mit blauer Schriftfarbe und gelbem Hintergrund.</p>



In diesem Beispiel wird der Text der <p>-Tags blau und der Hintergrund gelb gefärbt.

* **color:** Ändert die Farbe des Textes.
* **background-color:** Ändert die Hintergrundfarbe eines Elements.

Es gibt viele Farbwerte, die Sie verwenden können, z.B.:

* Farbname: red, blue, green
* Hex-Code: #FF0000 (rot), #00FF00 (grün)
* RGB: rgb(255, 0, 0) (rot)

## Zusatz: Class, ID

Mit CSS können Sie mehrere Stellen im Dokument gleichzeitig verändern, indem Sie Klassen (class) und IDs (id) verwenden.

**Klassen (class)**

Mit Klassen können Sie mehrere HTML-Elemente im Dokument gruppieren und denselben Stil zuweisen. Sie können dieselbe Klasse auf mehreren Elementen anwenden.

Beispiel:

<style>

.highlight {

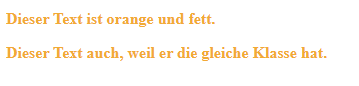
color: orange;

font-weight: bold;

}

</style>

<p **class**="highlight">Dieser Text ist orange und fett.</p>

<p **class**="highlight">Dieser Text auch, weil er die gleiche Klasse hat.</p>

* .highlight: Definiert die Klasse. Der Punkt (.) vor dem Namen zeigt an, dass es sich um eine Klasse handelt.
* **Vorteil:** Sie können viele Elemente mit der gleichen Klasse versehen, um den gleichen Stil anzuwenden.

**IDs (id)**

IDs sind einzigartig und werden verwendet, um ein einzelnes Element im Dokument zu stylen. Eine ID sollte nur einmal auf der Seite verwendet werden.

**Beispiel:**

<style>

#unique {

color: green;

font-size: 20px;

}

</style>

<p **id**="unique">Dieser Text hat eine einzigartige ID und ist grün und groß.</p>

* #unique: Definiert die ID. Das # vor dem Namen zeigt an, dass es sich um eine ID handelt.
* **Vorteil:** IDs sind eindeutig und können nur für ein einzelnes Element auf der Seite verwendet werden.

**Unterschied zwischen Class und ID:**

* **Class**: Kann auf viele Elemente angewendet werden. Wird mit einem Punkt (.) im CSS angesprochen.
* **ID**: Sollte nur einmal auf der Seite verwendet werden. Wird mit einem Rautezeichen (#) im CSS angesprochen.

## Zusatz: DIV, SPAN

Die Tags <div> und <span> sind beide allgemeine Container, die zur Gruppierung von HTML-Elementen verwendet werden, aber sie haben unterschiedliche semantische Bedeutungen und Verwendungszwecke.

**DIV-Tag**

Das <div>-Tag ist ein Block-Level-Element, das bedeutet, dass es einen eigenen Block auf der Seite bildet und die gesamte Breite des übergeordneten Elements einnimmt.

**Beispiel:**

<style>

.container {

background-color: lightblue;

padding: 20px;

}

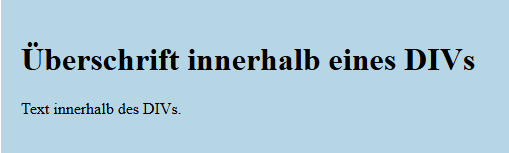
</style>

<div **class**="container">

<h1>Überschrift innerhalb eines DIVs</h1>

<p>Text innerhalb des DIVs.</p>

</div>

* **Verwendung:** <div> wird oft verwendet, um größere Bereiche einer Seite zu strukturieren und zu gruppieren. Es kann andere Block-Level-Elemente enthalten und wird typischerweise für Layouts genutzt.

**SPAN-Tag**

Das <span>-Tag ist ein Inline-Element. Es nimmt nur so viel Platz ein, wie der darin enthaltene Inhalt benötigt. Es wird verwendet, um kleinere Teile von Text oder Inhalt zu gruppieren und zu stylen, ohne den Fluss des Textes zu unterbrechen.

**Beispiel:**

<style>

.highlight {

color: red;

}

</style>

****<p>Der <span **class**="highlight">wichtige</span> Text ist rot.</p>

* **Verwendung:** <span> wird typischerweise für kleinere Textabschnitte oder einzelne Teile eines Absatzes verwendet, die stilistisch hervorgehoben oder bearbeitet werden sollen.

**Unterschied zwischen DIV und SPAN:**

* **DIV**: Block-Level-Element, das verwendet wird, um größere Bereiche zu gruppieren und eine neue Blockstruktur zu schaffen.
* **SPAN**: Inline-Element, das verwendet wird, um kleine Teile des Inhalts zu gruppieren oder zu stylen, ohne den Textfluss zu unterbrechen.

# PHP

## Einleitung

PHP steht für «PHP: Hypertext Preprocessor» und ist eine weit verbreitete und für den allgemeinen Gebrauch bestimmte Open Source-Skriptsprache, welche speziell für die Webprogrammierung geeignet ist und in HTML eingebettet werden kann. Die Abkürzung PHP ist ein rekursives Akronym. Das bedeutet, dass die Abkürzung «PHP» in der Erklärung «PHP: Hypertext Preprocessor» komischerweise gar nicht erklärt wird, sondern die Abkürzung als erstes Wort wieder vorkommt.

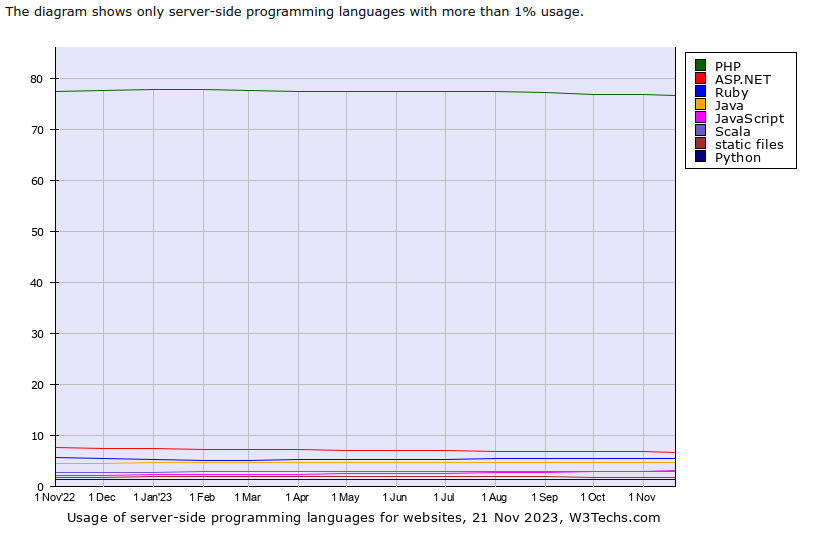
PHP unterscheidet sich von clientseitigen Sprachen wie Javascript dadurch, dass der Code auf dem Server ausgeführt wird und dort HTML-Ausgaben generiert, die an den Client gesendet werden. Der Client erhält also nur das Ergebnis der Skriptausführung, ohne dass es möglich ist herauszufinden, wie der eigentliche Code aussieht.

Das Beste an der Verwendung von PHP ist, dass es für Neueinsteiger extrem einfach ist, aber auch einen riesigen Funktionsumfang für den professionellen Programmierer bietet.

## Ist PHP noch relevant?

Laut W3Techs nutzen im 2023 knapp 80% der Website-Server PHP. Dies wird insbesondere durch beliebte Content-Management-Systeme wie WordPress erreicht. WordPress alleine wird von rund 33% der Websites genutzt.

(Bildquelle: W3Tech).



Diese Millionen von Websites werden nicht von heute auf morgen auf eine andere Programmierplattform umgestellt. Durch den aktuell hohen Marktanteil wird PHP also auch weiterhin weit im Web verbreitet sein.