

Bevor wir anfangen können, müssen wir ein paar grundlegende Begriffe kennenlernen

Browser	Das ist ein Programm, mit der man Internetseiten besuchen kann. Solche Programme sind zB FireFox, Opera, Google Chrome, Apple Safari, Microsoft Edge, Microsoft Internet Explorer
Client	Damit ist der Computer des Anwenders gemeint. Das ist quasi jedes Handy, Tablet, Laptop, PC, Mac etc. Der Client ist das Gerät, in dem ein Browser geöffnet ist, mittels dem man im Internet surft.
Server	Damit sind große Rechner bei Firmen wie die Telekom, Strato, Hosteurope, Checkdomain etc. gemeint, in der die Webseiten liegen. Jede Internetseite liegt auf irgendeinem Großrechner, bei irgendeiner Firma.
Client-Server-Beziehung	Wenn der Anwender auf seinem PC oder Handy zB bei Google irgendein Begriff eintippt und dann auf ENTER drückt, dann wird diese „Anfrage“ von seinem Browser, der Client ist, an den Server geschickt, in der Google liegt. Der Server nimmt die Anfrage und verarbeitet das im Hintergrund und schickt das Ergebnis der Anfrage wieder zurück zum Clienten. Der Client, also der Browser auf dem Desktop des Anwenders, liest die Antwort des Servers und bringt diese Information wieder auf dem Bildschirm. So erscheint bei Google beispielsweise das Suchergebnis-Liste einer Suche.
Domain bzw. Domainadresse	<p>Jede Internetseite auf der Welt wird über eine sogenannte Domainadresse erreicht. Eine Domainadresse sieht zum Beispiel so aus: www.meineseite.de</p> <p>Hinter dieser Domainadresse verbirgt sich eine echte Adresse aus Zahlen, den sich kaum ein Mensch merken kann. Sie ist quasi die Verbindungsadresse zu dem Server, auf dem die Webseite liegt.</p> <p>Das ist wie wenn du im Adressbuch deines Telefons einen neuen Eintrag für FRANK anlegen würdest. Der Eintrag heißt FRANK. In dem Eintrag FRANK liegen mehrere Informationen, mittels denen die eindeutige Adresse von FRANK ermittelt wird. Das wären dann die Straße, die Hausnummer, die Postleitzahl, der ORT und das Land.</p> <p>Es ist aber einfacher, sich den Namen FRANK zu merken, als diese Ganzen tatsächlichen Informationen zur Franks Adresse.</p>
URL	<p>Eine URL ist ein Pfad, das nicht nur auf eine bestimmte Domainadresse einer Webseite zeigt, sondern gegebenenfalls auch auf einen bestimmten Inhalt auf einer Webseite.</p> <p>Im Beispiel von FRANK wäre das zusätzlich zur FRANKS Adresse, die Etagennummer und das Zimmernummer in der Wohnung, in der er sich gerade aufhält. Zum Beispiel Etage 2, Wohnzimmer.</p>
Quellcode	Das ist Text, der jedoch den Code einer Programmiersprache enthält.
Quellcode-Datei	Das ist eine Textdatei, dass nur Code einer Programmiersprache enthält.
Ressource	In der Programmierung werden alle Dateien, die zur Realisierung einer Webseite, eines Programms benötigt werden, Ressourcen genannt. Denn sie sind notwendig, damit die Seite oder das Programm überhaupt ladet.

Dateiendung bzw. Datei-Erweiterung bzw. Dateityp Jede Datei auf dem Computer, bzw. Handy oder Server hat einen ganz bestimmten Typ. Dieser Typ ist notwendig, damit das Gerät weiß, wie er mit der Datei umgehen kann.

Zum Beispiel MP3 ist eine Musikdatei. Wenn du also eine MP3 Datei öffnen willst, dann schaut dein Gerät nach, ob es dafür ein geeignetes Programm gibt, der die Datei laden kann. Ein Audio-Abspielprogramm wird geladen und dieser spielt dann die Musik.

PNG oder JPG sind zum Beispiel zwei verschiedene Grafikdatei-Arten. Wenn du so eine Datei öffnen willst, dann schaut dein Gerät nach, ob es dafür ein geeignetes Programm gibt, um die Grafik auch auf dem Bildschirm dazustellen.

Nun zu JavaScript und PHP

JavaScript

PHP

Sowohl JavaScript, als auch PHP sind Programmiersprachen. Beide dienen dazu, verschiedene Abläufe zu realisieren. Ein Ablauf kann zum Beispiel das Klicken einer Schaltfläche sein oder das Anzeigen eines Informationsfensters.

Abläufe werden grob in zwei Kategorien aufgeteilt. Interaktionen und Hintergrundprozesse.

Interaktion	Hintergrundprozess
Eine Interaktion hat mit dem Desktop zutun.	Ein Hintergrundprozess ist auf dem Bildschirm nicht sichtbar. Aber dessen Ergebnis kann auf dem Desktop dargestellt werden, falls es das beabsichtigt.
Der Anwender klickt mit dem Mauszeiger auf etwas oder verschiebt den Mauszeiger auf etwas oder bewegt den Mauszeiger von etwas weg.	Hintergrundprozesse sind zum Beispiel solche Sachen wie Kopieren, Einfügen, einen Formular an dem Server schicken, Anmeldevorgang auf einer Webseite wie z.B. FaceBook oder eBay.
Der Anwender drückt auf eine Maustaste oder auf eine Taste auf dem Tastatur.	
Auf dem Handy bzw. Tablet berührt der Anwender ein Element auf dem Bildschirm mit einem oder mehrere Finger.	
JavaScript ist schwerpunktmäßig für Interaktionen ausgelegt. Sie kann jedoch auch Hintergrundprozesse auf dem Client realisieren.	PHP ist hauptsächlich für Hintergrundprozesse auf dem Server ausgelegt.
JavaScript-Code kann in HTML eingebettet werden oder in eine eigene Quellcode-Datei geschrieben werden. Eine JavaScript-Quellcode-Datei hat immer den Dateityp „.js“	PHP-Code muss immer in eine PHP-Quellcode-Datei geschrieben werden. Der zugehörige Dateityp ist immer „.php“
JavaScript läuft immer in einem Browser ab.	PHP läuft unabhängig vom Browser immer auf einem Server ab.

JavaScript und PHP Ähnlichkeiten

JavaScript

PHP

Beide Programmiersprachen unterstützen Funktionen. Eine Funktion ist ein Block mit einem Namen. In diesem Block kann man verschiedene Aufgaben hintereinander eingeben. Wird der Name der Funktion aufgerufen, so werden alle Aufgaben in dieser Funktion nacheinander ausgeführt. Eine Funktion erleichtert immer wiederkehrende Aufgaben.

```
function Bezeichner ()  
{  
    CODE  
}
```

```
function Bezeichner ()  
{  
    CODE  
}
```

Einer Funktion kann man „Parameter“ übergeben. Das heißt, die Funktion tut zwar immer dasselbe, aber sie kann variable Werte bekommen, mit denen sie arbeitet. Dadurch wird es zum Beispiel möglich eine Funktion zu erstellen, die zwei Zahlen addiert. Es addiert zwei Zahlen, aber diese zwei Zahlen können unterschiedliche Zahlen sein. Mal sind es 5 und 7, mal 8 und 3, ein andermal zB -6 und 166.

```
function addieren ( zahl1, zahl2 )  
{  
    CODE  
}
```

```
function addieren ( $zahl1, $zahl2 )  
{  
    CODE  
}
```

Im CODE-Teil werden „Anweisungen“ geschrieben. Eine Anweisung ist bereits eine Aufgabe, die getan werden soll. Jede Anweisung endet mit einem Semikolon.

```
X = 5;
```

```
$X = 5;
```

Um überhaupt Anweisungen realisieren zu können, benötigen wir Variablen. Eine Variable ist ein Bezeichner, den wir selbst definieren. Dieser Bezeichner wird von der Programmiersprache in eine echte Adresse im Arbeitsspeicher des Geräts umgewandelt und kann nun Daten zwischen speichern. Diese Daten werden benötigt, um die Nächste, dann die Nächste, dann die darauf folgende Anweisung umzusetzen.

Eine Variable kann folgende Daten zwischen speichern:

- Ganzzahlen, Positive oder Negativ (mit einem – Zeichen davor)
- Kommazahlen, Positiv oder Negativ (Kommazahlen werden mit Punkt geschrieben)
- Text (Text wird zwischen ' ' oder zwischen “ ” geschrieben)

Unter JavaScript muss eine Variable immer vor der Verwendung angelegt werden. Das wird mit dem Schlüsselwort „var“ realisiert. ZB:

```
var x; var y; var dies; var das;
```

Unter PHP wird eine Variable automatisch angelegt, sobald sie das erste Mal namentlich verwendet wird. Eine PHP-Variable beginnt aber immer mit einem \$-Zeichen. Beispiel:

```
$info = 'Hallo, ich bin eine Variable';
```

Man kann Variablen „Global“ und „Lokal“ anlegen. Eine globale Variable wird außerhalb einer Funktion angelegt. Sie kann anschließend in allen Funktionen, die nachher erstellt werden, benutzt werden.

Eine lokale Variable wird immer in einer Funktion angelegt. Sie kann nur in der Funktion verwendet werden, in der sie angelegt wurde.

Lokaler Beispiel:

<pre>function Bezeichner () { var hallo; hallo = 'Wie gehts?'; }</pre>	<pre>function Bezeichner () { \$hallo = 'Wie gehts?'; }</pre>
---	---

Globaler Beispiel:

<pre>var hallo; function Bezeichner () { hallo = 'Wie gehts?'; }</pre>	<pre>\$hallo = ""; function Bezeichner () { global \$hallo; \$hallo = 'Wie gehts?'; }</pre>
---	--

Eine Funktion, die keine Parameter enthält, ruft man so auf:

Bezeichner();

Hat sie Parameter, dann muss man die Parameter angeben:

Bezeichner(2, 3, 'ok', 4.5, c, d);	Bezeichner(2, 3, 'ok', 4.5, \$c, \$d);
--------------------------------------	--

Manchmal muss man eine bestimmte Aufgabe in derselben Funktion mit abweichenden Werten mehrfach ausführen. Dafür gibt es Schleifen. Wir werden hier die einfachste Variante kennen lernen, die FOR-Schleife.

<pre>function Bezeichner (von, bis) { var z; for (z = von; zaehler < bis; z++) { alert(z); } }</pre>	<pre>function Bezeichner (\$von, \$bis) { var \$z; for (\$z = \$von; \$zaehler < \$bis; \$z++) { echo \$z; } }</pre>
---	---

Eine wichtige Aufgabe von Anweisungen ist es, das Verhalten der Webseite bzw. Programms zu steuern. Dazu müssen Daten ausgewertet und eine Entscheidung getroffen werden. Anhand dessen wird dann eine bestimmte Aufgabenstellung realisiert.

Dazu hilft uns der IF-Block. Wir werden hier die einfachste Form des IF-Blocks kennenlernen.

```
function Bezeichner ( wert )
{
    if ( wert == 0 )
    {
        alert( 'Das ist eine glatte NULL' );
    }

    if ( wert == 1 )
    {
        alert( 'Das ist schon eine EINS' );
    }

    if ( wert == 2 )
    {
        alert( 'Cool, glatte ZWEI :-)' );
    }
}

function Bezeichner ( $wert )
{
    if ( $wert == 0 )
    {
        echo 'Das ist eine glatte NULL';
    }

    if ( $wert == 1 )
    {
        echo 'Das ist schon eine EINS';
    }

    if ( $wert == 2 )
    {
        echo 'Cool, glatte ZWEI :-)';
    }
}
```

Eine IF-Auswertung kann unterschiedlich durchgeführt werden. Dazu sind zwei Operanden und ein Operator nötig. Die Operanden im obigen Code sind zum Beispiel Wert und 1, Wert und 2 sowie Wert und 3. Der Operator ==. Es gibt aber auch andere Operatoren und jede Operator hat eine andere Vergleichsfunktion:

==	Operanden werden auf GLEICHHEIT geprüft. Wenn beide Werte GLEICH SIND, wird der Inhalt des IF-Blocks ausgeführt, sonst nicht.
!=	Operanden werden auf UNGLEICHHEIT geprüft.
>=	Es wird geprüft, ob der rechte Operandwert GRÖßER oder GLEICH dem linken Operandwert ist.
<=	Es wird geprüft, ob der rechte Operandwert KLEINER oder GLEICH dem linken Operandwert ist.
>	Es wird geprüft, ob der rechte Operandwert GRÖßER dem linken Operandwert ist.
<	Es wird geprüft, ob der rechte Operandwert KLEINER dem linken Operandwert ist.