**PHP-Syntax**

**Start** + **U** + **cmd** Wechsel zur Kommandozeile (DOS-Fenster)

**Start** + **cmd** Wechsel zur Kommandozeile unter **Windows**

**exit [Return]** Kommandozeile verlassen, Rückkehr zum Windows-Desktop

**<?php** leitet ein PHP-Programm ein

**?>** beendet ein PHP-Programm

**Kommentare, Ausgabe, Zuweisungen**

**//** Kommentar (gilt nur für eine Zeile)

**# (Raute)** alternatives Zeichen für einen Zeilenkommentar

**/\* \*/** mehrzeiliger Kommentar

**; (Semikolon)** jede Zeile wird mit einem Semikolon (Strichpunkt) abgeschlossen

**echo** Ausgabe auf dem Monitor; Text muss in Anführungszeichen stehen

**$x** Jede Variable beginnt mit dem Dollarzeichen;  
Groß- und Kleinschreibung beachten

**=** Zuweisungsoperator

**NULL;** im Beispiel: $x = NULL; hat die Variable $x keinen Wert

**. (Punkt)** Verknüpfungsoperator oder Punktoperator;   
auch "Textkleber" genannt

**Datentypen**

**integer** Datentyp für eine Ganzzahl

**short, long** Datentyp für eine Ganzzahl (short und long in php **nicht** verwendet)

**float, double,** Datentypen für eine Kommazahl (Fließkommazahl)

**real**

**char** Datentyp für einzelneBuchstaben(in PHP **nicht** verwendet)

**String** Datentyp für Text; muss in doppelten Anführungszeichen stehen

**Escape-Sequenzen**

**\n** Zeilenumbruch (Das "n" muss klein geschrieben werden)

**\t** Tabulatorsprung (Escape-Sequenz)

**\r** Wagenrücklauf; der Cursor kehrt auf die erste Zeilenposition zurück

**\\** Backslash; den Backslash als Textzeichen erzeugen

**\$** Das Dollarzeichen als Textzeichen

**\"** Das Anführungszeichen als Textzeichen

**%%** Prozentzeichen als Textzeichen **(Ausnahme: mit % escapen!)**

**fgets(STDIN)** Tastatureingabe, Beispiel: $liter = fgets(**STDIN**);   
"STDIN" immer **groß** schreiben!

**DEFINE** Eine Konstante wird definiert; Bsp: DEFINE ("PI", 3.14159265)

**round()** Liefert eine Zahl auf bestimmte Genauigkeit gerundet;  
Beispiel: round(3.56, 1) = 3.6

**Formatzeichen (Formatstrings)**

**printf()** Formatierte Ausgabe

**%0.2f** Zahl wird mit zwei Nachkommastellen (Fließkommazahl) ausgegeben

**%s** Eine Zeichenkette (String) wird ausgegeben

**%d** Eine Ganzzahl (Dezimalwert mit Vorzeichen) wird ausgegeben

**%u** Dezimalwert ohne Vorzeichen

**%2d** Vorkommastellengenaue (Mindestbreite 2 Stellen)  
Ausgabe einer Ganzzahl

**%b** Variable wird mit seinem Binärwert ausgegeben

**%x** Eine hexadezimale Zahl wird ausgegeben (klein geschrieben)

**%X** Eine hexadezimale Zahl wird ausgegeben (groß geschrieben)

**%c** Das Zeichen, das der Position in der ASCII-Liste entspricht,  
wird ausgegeben

**%o** Eine oktale Zahl wird ausgegeben

**%%** Prozentzeichen als Textzeichen **(Ausnahme: mit % escapen!)**

**trim()** Entfernt überflüssige Zeichen ("whitespace")   
am Anfang und Ende eines Strings

**strlen()** Ermittelt die Zeichenanzahl eines Strings (String length)

**(int), (double)** Datentypvorgaben; z.B. bei Tastatureingabe: (int) fgets(STDIN)

**modulo** Der Rest (**Modulo)** einer Division wird berechnet (Bsp: 13 **%** 6 = 1)

**Fallunterscheidung (Auswahl)**

**if (Ausdruck)** Die Bedingung muss immer in runden Klammern stehen

**if/else** Bei Nichtzutreffen der Bedingung wird der else-Zweig ausgeführt

**if-else if-else** Zweite Bedingung wird geprüft

**true, false** "wahr", "falsch" (Wahrheitswerte – boole’scher Term)

**switch-case** Mehrfachverzweigung (nur bei Prüfung auf Gleichheit möglich)

**break** Zwischen den case-Zweigen muss eine break-Anweisung stehen

**default** Die Standard-Anweisung (Default),  
wenn die Variable **keinem** der case-Werte entspricht

**Vergleichsoperatoren**

**==** "genau gleich"

**===** Wert und Datentyp gleich

! Das Ausrufezeichen bedeutet "NICHT"

**!=** "ungleich" (nicht gleich)

**>** größer als

**>=** größer oder gleich

**<** kleiner als

**<=** kleiner oder gleich

**Logische Operatoren**

**AND** Logischer Operator "UND"

**&&** Logischer Operator "UND" (höhere Priorität)

**OR** Logischer Operator "ODER"

**||** Logischer Operator "ODER" (höhere Priorität)

**String-Funktionen**

**strtolower()** Setzt einen String in Kleinbuchstaben um; z.B. strtolower("M") = "m"

**strtoupper()** Setzt einen String in Großbuchstaben um

**substr()** Kopierfunktion; Beispiel: substr("BFW-Wuerzburg", 1, 3) = "BFW"

**strrev()** "Dreht" einen String um (Palindrom); z.B. „Sonne“ -> „ennoS“

**chr()** liefert das zu einer ASCII-Position gehörende Zeichen

**ord()** liefert die zu einem Zeichen gehörende ASCII-Position

**Funktionen Kommandozeile**

**Strg + C** Laufendes Programm abbrechen (z. B. bei Endlosschleifen)

**Strg + S** Laufendes Programm wird angehalten und fortgesetzt   
(wie eine **Pause**-Taste) (ggf. veraltet)

**Programmschleifen**

**while-Schleife** Kopfgesteuerte, abweisende Schleife

**do-while-Schleife** Fußgesteuerte, nicht abweisende Schleife

**for-Schleife** Syntax: for (Initialisierung, Testbedingung, Inkrement)

**continue** Schleifendurchgang wird vorzeitig beendet

**break** Schleife wird verlassen

MS