PHP
früher Personal Home Page
heute Hypertext Preprocessor

Serverseitiger Script Interpreter
Webserver das PHP Modul
Dateien mit der .php verarbeitet werden

Namenskonvention
Variablen- und Funktionsbezeichner gelten
Nur Buchstaben, Ziffern und Unterstrich
erste Zeichen keine Ziffern
Groß/Kleinschreibung wird unterschieden
Beispiel
Zahl, zahl, ZAHL, _Zahl, _2 sind verschiedene gültige Bezeichener
2Zwei, -Zwei sind keine gültigen Bezeichner

Kommentare
Zeilenkommentar // oder
Blockkommentar /* */

Gültigkeitsbereich
Datei
Block {}

Trennzeichen
Semikolon, Komma, Operatoren, Schlüsselwörter, Leerzeichen

Variablen
Definiert mit $Bezeichner
Implizite Datentyp Zuweisung
Datentyp wird anhand des zugewiesenen Wert festgelegt
Datentypen
Object, Zahl/Number(Ganze- oder Kommazahl), Logisch (true/false) und Text(String) eingeleitet durch "" oder ''

Beispiel:
```
$Text="Hallo, from PHP World"; // Text
$Zahl=1 // Ganzezahl
$Waehrung=12.12 // Kommazahl
$Bedingung=true // logischer Typ
$Objekt=new Datentyp(); oder =Objecttyp// Objekttyp
$Waehrung=$Waehrung." Euro"; // => Text
```

Operatoren
Arithmetische Operatoren +,-,*,/ und %
Vergleichsoperatoren >,<,>=,<=,==,!= vergleichen nur Inhalt
```
  $Zahl=12;
  $Text="12";
  $Zahl==$Text => true es wird $Zahl in auto Text umgewandelt
  !== oder === Vergleichen Inhalt und Typ
```

```
  $Zahl===$Text => false
Logische and(&&), or(||) und ! verknüpft zwei logische Ausdrücke miteinander

Stringoperatoren
  . der verknüpft zwei Strings miteinander
  $Text="Hallo"."from PHP World"; => Hallofrom PHP World
  $Text=$Text.$Zahl;  => Hallofrom PHP World12 // in javascript $Text+"Text"
  $Zeichenkette='$Text'; => $Text                  Die Zeichen werden Zeichenkette zugewiesen
  $Kette       ="$Text"; => Hallofrom PHP World12 der Inhalt von $Text wird Kette zugewiesen

Sichtbarkeit
lokale Variablen die nur innerhalb eines Block in dem diese definiert sind sichtbar
{ $Text="lokal"; {} }
globale Variable diese ist ausserhalb eines Blocks definiert und kann mit global
                 innerhalb eines Blocks sichtbar gemacht werden

$Global=123;
{global $Global; $Global=456;}

super globale Variablen die ohne global überall sichtbar sind
$GLOBAL
$_SERVER     Liste mit Server Informationen
$_REQUEST, $_POST, $_GET Liste mit Parameter der URL ?Parameterliste Nachname=Maier
                         $_REQUEST["Nachname"] liest den Wert des Parameters Nachname
$_FILES      Liste mit Dateien
$_SESSION    Liste mit Informationen zur aktuellen Session
$_COOKIES    Liste mit Cookies
$_ENV        Liste mit den Umgebungseinstellungen

Kontrollstrukturen
Bedingung
  einfach  Wert Operator Vergleichswert
  komplex  logische Verküpfung von einfach Bedingung
Verzeigungen
  einfach Verzeigung if (Bedingung) {} else {}
  komplex
    verschachtelten einfach Verzeigungen
    Fallunterscheidung if - else if - ... -else
                       switch - case Fall - .... - default
                       Fall kann hier auch 'Text'
Schleifen
  kopfgesteuert
    for (init; Bedingung; Nachbearbeitung) {}
      for ($Index=0; $Index < $MaxElement; $Index=$Index+1 /* ++$Index, $Index+=1 oder $Index++ */) {
        print $Index;
      }
    while (Bedingung) {}
    foreach ($Liste as $Element) {echo $Element;}
  fussgesteuerte
    do {} while(Bedingung);
```

```
  Schleifensteuerung
    continue und
    break

Ausgabe
  echo  $Wert;  // in HTML Dokumenten
  print $Wert;  // PHP Console
  print_r($Liste);  // die Liste in struktorierter From aus

Listen
  es gibt drei Typen von Listen
    Indexbasierte Liste
      //$Liste=array(Werteliste durch Kommagetrennt); // Objekttyp
      $Liste=array(1,2,3,4); // enthält vier Element
      // $Liste[$Index] kann man auf die Element zugreifen $Index läuft 0 und MaxElement-1
      echo $Liste[3]; // liefert den Wert 4
      $Liste[0]=10; // ändert das erste Element auf den Wert 10
      Liste erweitern
        $Liste[]=5; // fügt der Liste ein weiteres Element hinzu
        // 10,2,3,4,5
        // array_push($Liste, Werteliste durch Komma getrennt);
    Assoziative Liste
      Schlüsselbasierte Liste "key" => "value"
      $Person=array("Nachname" => "Maier", "Vorname" => "Paul", "Alter" => 30);
      echo $Person["Nachname"]; // liefert Maier
      foreach($Person as $Key => $Value) {echo $Key."-".$Value;}
      $Person["Vorname"]="Paula";
    Listen aus Listen Felder
      // $Feld=array(array(...),array(...),...); /Liste mit Listen
      $Personen=array(array("Nachname"=>"Maier","Vorname"=>"Egon", "Alter"=20), $Person);
      echo $Personen[0]["Nachname"];  // Maier
      $Werte=array(array(1,2,3), array(4,5,6));
      echo $Personen[0][1]; // liefert den Wert 2
    sortierten Listen
      // sort(array)  sortiert die indexbasierte Liste in aufsteigender Reihenfolge
      // rsort(array) sortiert die indexbasierte Liste in absteigender Reihenfolge
      // (r)asort(array) sortiert die Werte der assoziative Liste in aufsteigender Reihenfolge
      // (r)ksort(array) sortiert die Schlüssel der assoziative Liste in aufsteigender Reihenfolge
      // r bedeutet absteigend

Funktionen
  function Bezeichner(Parameterliste) : Datentyp { }
  Datentyp ist optional
  Parameterliste Liste mit Parametern die mit Komma getrennt sind
  Defaultwerte für die Parameter setzen von links nach rechts
  Parameter Typen
    call by value     mit $Bezeichner
    call by reference mit &$Bezeichner
    call by value with reference mit $Bezeichner
  function berechnen(&$Zahl1, $Zahl2) {
```

```
      $Zahl3=$Zahl1+$Zahl2; // 12+5
      $Zahl =$Zahl+1;        // 13
      return $Zahl3;
    }
    $Zahl=12;
    $Ergebnis=berechnen($Zahl,5);
    echo $Zahl;     // enthält 13
    echo $Ergebnis; // enthält 17

  Text Funktionen
    strlen($Text)  liefert die Anzahl der Zeichen innerhalb der Zeichenkette
    str_replace(suchtText,Ersetzungstext,$Text) ersetzt suchtext durch ersetzungstext in $Text
    str_replace(",", ".", $Preis);
    $Neu=substr($Text, start, Anzahl) gibt aus $Text ab start Anzahl Zeichen aus
```