

## # Vorbereitung

Öffnen Sie ein neues Html-Dokument innerhalb von EditPlus und verwenden Sie in den nachfolgenden Aufgaben das bekannte minimale HTML-Gerüst:

```
<html><body><pre><script type="text/javascript">

</script></pre></body></html>
```

## # Aufgabe 1

Wir gehen von folgenden gegebenen Variablen aus.

```
var a=5, b="42", c=84, d=-2, e="false", f=true;
```

Sagen Sie nun das Ergebnis nachstehender Ausdrücke aufgrund der jeweiligen Datentypen und der beteiligten Operatoren voraus. Eine etwaige Änderung der Variablenwerte bleibe unberücksichtigt. Geben Sie die Ergebnisse dann mittels `document.write(... + "\n")` aus. Sehen Sie sich die [Einträge über Operatoren](#) an.

```
a > b
2*b == c
(a+b)/d
d += c % a
a + c/2 < b || f == 1
!(true == false)
d<=a && c%5 >= 2
"true" < "True"
"heino" >= "hein"
"hal" + "lo" == "hallo"
```

## # Aufgabe 2

Geben Sie für die mathematische Funktion

$$y = x^2 - 2 \cdot x - 4$$

im Bereich  $-4 \leq x \leq 4$  eine Wertetabelle in folgender Darstellung aus.

```
x | y
=====
. . | . .
```

Nutzen Sie hierfür den Algorithmus (Pseudocode):

```
define new variables x=-4, y;

while (x is less or equal 4) {
  calculate y;
  output y and x;
  increment x;
}
```

*Hinweise:*

- Die while-Schleife ist [hier dokumentiert](#).
- Benutzen Sie `\n` (*newline*) und `\t` (*tabulator*).

### # Aufgabe 3

Wir verwenden diesmal in einem neuen Dokument ein Grundgerüst ohne `<pre>` Element

```
<html><body><script type="text/javascript">

</script></body></html>
```

und geben die Wertetabelle aus *Aufgabe 2* als [HTML-Tabelle](#) aus.

### # Aufgabe 4

Mittels einer [while-Schleife](#) soll folgende ASCII-Figur zeilenweise ausgegeben werden,

```
*
**
***
****
*****
```

wobei die Anzahl der Zeilen  $n$  vorher vom Benutzer zu erfragen ist.

*Hinweise:*

- Verwenden Sie wieder das [obige Grundgerüst](#).
- Das Steuerzeichen `\n` (*newline*) erzeugt einen Zeilenvorschub.

### # Aufgabe 5

Ändern Sie Ihr Programm aus der vorangegangenen *Aufgabe 4* so, dass nun folgende Figur

```
*
 *
  *
   *
    *
```

entsteht.

## # Aufgabe 6

Schreiben Sie Ihr Programm aus *Aufgabe 5* unter Verwendung der [for-Schleife](#) so um, dass dieselbe Ausgabe entsteht.