

Funktionen Einstieg

1.

1.1 quadrat

Schreiben Sie eine Funktion die einen Parameter annimmt und das Quadrat, die Multiplikation dieser Zahl mit sich selbst, zurückgibt.

1.2 difference

Schreiben Sie eine Funktion die zwei Parameter annimmt und die Differenz der beiden Zahlen zurückgibt.

1.3 product

Schreiben Sie eine Funktion die zwei Parameter annimmt und das Produkt der beiden Zahlen zurückgibt.

1.4 numberCompare

Schreiben Sie eine Funktion die zwei Zahlen als Parameter annimmt.

Ist die erste größer als die zweite

return "Erster Wert x ist größer als zweiter Wert y"

Ist die zweite größer als die erste

return "Zweiter Wert y ist größer als erster Wert x"

Sind beide Werte gleich

return "Zahlen sind gleich"

Beispiele:

```
numberCompare(1,1); // "Numbers are equal"
```

```
numberCompare(2,1); // "First is greater"
```

```
numberCompare(1,2); // "Second is greater"
```

1.5 singleLetterCount

Schreiben Sie eine Funktion die zwei Strings als Parameter annimmt. Der erste Parameter ist ein String mit einem Wort und der zweite Parameter ist ein string mit einem Buchstaben. Ermitteln Sie wie oft der Buchstabe (Parameter 2) im Wort (Parameter 1) vorkommt und geben Sie die ermittelt Häufigkeit zurück. Prüfen Sie case insensitive, Groß-/Kleinschreibung wird beim Vergleich ignoriert.

TIPP: S. 161 6.2.1 Operationen mit Zeichenketten

2. Schreiben Sie eine Funktion die eine Zahl als Parameter annimmt und umdreht. Der Rückgabewert ist eine Zahl.

Beispiel x = 12345;

Erwartet Ausgabe : 54321

3. Schreiben Sie eine Funktion die einen string als Parameter annimmt und den ersten Buchstaben in diesem String in Großbuchstaben umwandelt.

TIPP: S. 161 6.2.1 Operationen mit Zeichenketten

Beispiel string = "programmieren";

Erwartet Ausgabe : "Programmieren"

4. Schreiben Sie eine Funktion die einen Parameter annimmt und den Datentyp des Wertes zurückgibt. Mögliche Rückgabewerte: object, boolean, function, number, string, and undefined.

5. Was gibt folgendes Code aus und warum? Erklären Sie jeden Baustein dieses Codes.

```
(function() {  
  var a = b = 5;  
})();  
console.log(b);
```

6. Legen Sie im Ordner 01_grundlagen > pages eine Datei „06_math_objekt.html“ an.

6.1 Erzeugen Sie zuerst eine Zufallszahl zwischen 1 und 6 inklusive 1 und 6.

6.2 Schreiben Sie eine Funktion die zwei Parameter (Datentyp number) annimmt.

Die Funktion soll eine zufällige Zahl von Parameter 1 bis Parameter 2 erzeugen.

Parameter 1 ist die Untergrenze (inklusive), kann also ermittelt werden.

Parameter 2 ist die Obergrenze (inklusive), kann also ermittelt werden.

7. Testen Sie die Funktion aus Aufgabe 6.

Erzeugen Sie 10000 zufällige Zahlen von 1 bis 10. Ermitteln Sie wie häufig jede der Zahlen in den 10000 Durchläufen vorgekommen ist.

TIPP: Da wir noch keine Arrays behandelt haben müssen Sie für jede Zahl eine Variable anlegen.

zahl1, zahl2 ... zahl10

Mit einer Schleife, die 10000 Durchläufe hat, erzeugen Sie in jedem Schleifendurchlauf eine neue zufällige Zahl von 1 bis 10. Die vorher angelegten Hilfsvariablen zahl1 bis zahl10 dienen dazu die Häufigkeit zu zählen.

Geben Sie nachdem die Schleife fertig ist die Häufigkeit der einzelnen Zahlen wie folgt aus:

Die Zahl 1 kam: 946 mal vor.

Die Zahl 2 kam: 1025 mal vor.

Die Zahl 3 kam: 1000 mal vor.

Die Zahl 4 kam: 1037 mal vor.

Die Zahl 5 kam: 989 mal vor.

Die Zahl 6 kam: 1039 mal vor.

Die Zahl 7 kam: 972 mal vor.

Die Zahl 8 kam: 970 mal vor.

Die Zahl 9 kam: 1042 mal vor.

Die Zahl 10 kam: 980 mal vor.

Test: Häufigkeit aller Zahlen 10000