Exercise 1

Warum führt folgender Code zu einem Fehler?

```
1 let try = "Let's try!";
```

Exercise 2

Welche Variablenname sind gültig bzw. ungültig?

let 2ndName;
let first%Name;
let first-name;
let first_name;
let _firstName;
let $firstName;

Exercise 3

Was bedeutet Lower-CamelCase? Nenne mindestens noch eine weitere Notationen inklusive Beispiel.

Exercise 4

Was bedeutet Loose Typing? Wie unterscheidet es sich von Strong Typing? Nenne jeweils 3 Programmiersprachen, die Loose bzw. Strong Typing verwenden.

Exercise 5

Welche Datentypen gibt es in JavaScript? Finde jeweils ein Beispiel.

Exercise 6

Was bedeuten die Datentypen null und undefinded? Gibt es einen Unterschied? Erkläre die Datentypen anhand von Beispielen.

Exercise 7

Beim unteren Beispiel hat sich ein Fehler eingeschlichen. Wie muss die Logik verändert werden, sodass alle Statements korrekt wiedergegeben werden? Implementiere die Änderung.

```
function orderMyLogic(val) {
    if (val < 10) {
      return "Less than 10";
    } else if (val < 5) {
      return "Less than 5";
    } else {
      return "Greater than or equal to 10";
    }
  }

  orderMyLogic(7);
```

Exercise 8

Implementiere für folgenden Code eine if/else-if-Logik.

```
  function testElseIf(val) {
    if (val > 10) {
      return "Greater than 10";
    }

    if (val < 5) {
      return "Smaller than 5";
    }

    return "Between 5 and 10";
  }

  testElseIf(7);
```

Exercise 9

Schreibe eine Funktion mit einem if/else-if Statement. Basierend auf dem Input, ändert sich der Rückgabewert der Funktion:

Nummer kleiner als 5 → Tiny
Nummer zwischen 5 und 10 → Small
Nummer zwischen 10 und 15 → Medium
Nummer zwischen 15 und 20 → Large
Nummer größer als 20 → Huge

```
  //Startpunkt
  function findTheRightSize(num) {
    // Starte mit dem Code

  }
```

Exercise 10

Eine Funktion soll basierend auf den Argumenten unterschiedliche Werte in einem Hinweisfeld ausgeben.

Argument (Statuscode)	Ausgabe in Hinweisfeld
1	Alpha
2	Beta
3	Gamma
4	Delta
Sonstiger Wert	Unbekannter Wert

Einschränkung: Das Schlüsselwort „if“ darf nicht verwendet werden

```
1  //Startpunkt
2  function showAlert(statusCode) {
3    let message;
4    // Starte mit dem Code
5
6  }
```

Exercise 11

Erstelle eine Funktion, die einen Integer als Argument nimmt und "Even" für gerade Zahlen und "Odd" für ungerade Zahlen zurückgibt.

Exersice 12

Vervollständigen Sie die Methode, die einen booleschen Wert als Argument erhält und eine "Yes"-Zeichenkette für "true" oder eine "No"-Zeichenkette für "false" zurückgibt.

```
//Startpunkt
function boolToWord(bool) {
  // Starte mit dem Code

}
```

Exercise 13

Implementiere eine Methode, die eine beliebige Basis-Zahl als Argument erhält. Diese Zahl soll anschließend jeweils mit den Werten 2,4,6,8,10,12,14 und 16 multipliziert werden. Die Ergebnisse der Multiplikationen sollen auf der Konsole ausgegeben werden.

Das Ergebnis könnte so aussehen (Basis-Zahl 2)

```
2 multipliziert mit 2 = 4
2 multipliziert mit 4 = 8
2 multipliziert mit 6 = 12
2 multipliziert mit 8 = 16
2 multipliziert mit 10 = 20
2 multipliziert mit 12 = 24
2 multipliziert mit 14 = 28
2 multipliziert mit 16 = 32
```

Exercise 14

Schreiben Sie eine Funktion take_umbrella(), die zwei Argumente annimmt: einen String, der das aktuelle Wetter darstellt (sunny, cloudy und rainy), und eine Zahl, die die Regenwahrscheinlichkeit für heute angibt (zwischen 0 und 1).

Ihre Funktion sollte True oder False zurückgeben, basierend auf den folgenden Kriterien.

- Man sollte einen Regenschirm mitnehmen, wenn es gerade regnet oder wenn es bewölkt ist und die Regenwahrscheinlichkeit über 0,20 liegt.
- Man sollte keinen Regenschirm mitnehmen, wenn es sonnig ist, es sei denn, die Regenwahrscheinlichkeit ist über 0,50.
- Wenn es regnet muss natürlich ein Regenschirm eingepackt werden.

Zum Beispiel sollte take_umbrella('sunny', 0.40) False liefern.

Exercise 15

Schreiben Sie eine Funktion rps(), kurz für „Rock, paper, scissors“, die zwei Argumente annimmt: den Wert für Spieler 1 (p1) und den Wert für Spieler 2 (p2). Gültige Werte sind: „rock“, „paper“ und „scissors“.

Die Funktion soll den Gewinner des Duells in der Konsole ausgeben. Bei Unentschieden soll „Draw!“ ausgegeben werden.
Beispiel: rps(„rock“, „paper“) und das Ergebnis ist „Player 2 won!“.