**GO**

Für was brauche ich GO?

1. Performance:   
   Go wird direkt in Maschinencode kompiliert
2. Lesbarkeit:  
   Go hat eine klare und einfache Syntax
3. Skalierbarkeit:  
   Go wurde von Google entwickelt, um grosse, skalierbare Anwendungen zu unterstützen. Man kann mehrere Prozesse parallel laufen lassen.
4. Ideal für Netzwerk- und Cloud Anwendung:  
   GO ist besonders gut für Webserver und Microservices

Wann sollte Ich GO verwenden?

1. Server-Anwendungen und Backends
2. CLI-Programme
3. Microservices
4. Streaming

**Variablen in GO:**

Eine Variable wird immer mit «var» deklariert.

Beispiel: Beispiel:   
var «Variable Name» «Datentyp» var MyName string

Um einer Variable beim Deklarieren einen Wert zu geben:  
Beispiel: Beispiel:   
var «Variable Name» «Datentyp» = «Wert» var MyNum int = 2  
Oder:  
Beispiel: Beispiel:   
var «Variable Name» = «Wert» var MyNum = 2  
Oder:  
Beispiel: Beispiel:   
«Variable Name» := «Wert» MyNum := 2

**Konstanten in GO:**

Konstanten definiert man mit const:

Beispiel: Beispiel:   
const «Name» «Datentyp»=«Wert» const MyName string = «Luizz»

**Datentypen in GO:**

Folgende Datentypen gibt es in GO:

1. Zahlen

* Int
* Uint
* Float

1. Wahrheitswerte

* Bool

1. Zeichen

* rune

1. Zeichenketten

* String

**Ausgabe in GO:**

Mit der Funktion fmt und mit Fprintln() kann man dinge ausgeben.

Beispiel:

Fmt.Printf(“10 divided by 3 is %sn”, result)

**Strukturen in GO:**

Eine Struktur struct ist ein zusammengesetzter Datentyp, der mehrere Variablen in einer Einheit zusammenfasst.

**Struktur erstellen:**

Eine Struktur wird mit dem Schlüsselwort type und struct erstellt:

**Kontrollstrukturen in GO**

**Verzweigungen:**

* **If & elseSwitch/case**
* **Fallthrough**

**Schleifen:** - for  
 - while  
 - continue  
 - break  
 - range

**Beispiel**:

**Funktionen in GO:**

**Definieren**

Funktionen werden mir func definiert

Beispiel:  
 func f(x int) int {

y := 2 \* x  
 return y  
 }

**Aufrufen**

Die Funktion wird mit FunctionName(parameter) übergeben  
Beispiel:

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Betriebssystem enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Betriebssystem enthält.

Automatisch generierte Beschreibung y := f(2)  
 fmt.Println(y)