1. Die allgemeine Syntax von turtlescript lautet:
[TYPE]@[VARIABLE NAME] = [VALUE];
das gleiche könnte man auch als:
[TYPE]@[VARIABLE NAME]= [VALUE];
schreiben.
2. Ein Type(im obrigen Code [TYPE]) ist dabei:

ID	TYPE	C++ Typ
1	INTEGER	TUR_INTEGER
2	BOOLEAN	TUR_BOOLEAN
3	STRING	TUR_STRING
4	BIGNUM	TUR_BIGNUM
5	BYTE	TUR_BYTE
6	FLOAT	TUR_FLOAT

3. *INTEGER*
Ein Integer wird als TUR_INTEGER(int) gespeichert.
Es hat einen Wertebereich von -2147483648 bis 2147483647.
Dabei wird das Minuszeichen direkt vor der Zahl hingeschrieben.
4. *BOOLEAN*
Ein Boolean wird als TUR_BOOLEAN(int) gespeichert.
Es kann entweder den Wert true(wahr) oder false(unwahr/falsch) annehmen.
5. *STRING*
Ein String wird als TUR_STRING(string) gespeichert.
Der Text des Strings wird in Anführungszeichen geschrieben.
Dabei wird das doppelte Anführungszeichen(") verwendet werden.
6. *BIGNUM*
Eine Bignum wird als TUR_BIGNUM(long long) gespeichert.
Es hat einen Wertebereich von $-(2^{63})$ bis $(2^{63})-1$.
Dabei wird das Minuszeichen direkt vor der Zahl hingeschrieben.
7. *BYTE*
Ein Byte wird als TUR_BYTE(unsigned char) gespeichert.
Es hat einen Wertebereich von 0 bis 255.
Es sollte vermieden werden höhere Werte als 255 in einem Byte zu speichern.
Der Wertebereich eines Byte ist mindenes so groß das, das ASCII-Alphabet(auch ASCII-Code) dastellbar ist.
8. *FLOAT* Ein Float wird als TUR_FLOAT(string) gespeichert.
Dabei wird bis zur Auswertung bzw. Auslesung der Daten das Float als String gespeichert.
Erst dann, wenn es gelesen wird, wird es in ein float umgewandelt.

9.(C++) Um die Dateien mit einem C++ Compiler zu kompilieren benötigt man das C++ Boost Library.
10.(C++) In C++ wird es erfolgreicher inkludierung der Header Datei und erfolgreichem Kompilimiren der turtlescript-Dateien ein Namensbereich(namespace) turtlescript existieren.
In diesem Namespace befinden sich die Funktionen editvar, eval, context, stack, getinteger, getboolean, getstring, getbignum, getbyte, getfloat und clear.
11.(C++) *editvar*
Nicht für den Benutzer/Entwickler vorgesehen.
12.(C++) *eval*
Interpretiert turtlescript Code.
Fordert als erstes Argument den turtlescript Code, der interpretiert werden soll.
Gibt immer eins zurück.

13.(C++) *context*
Nur teilweise für den Benutzer/Entwickler vorgesehen.
Interpretiert turtlescript Code-Value.
Fordert als erstes Argument den Value der interpretiert werden soll.
Gibt das Ausgewertete Value zurück.
Als Value wird bei der Zuweisung der nach den Zuweisungszeichen(=) stehenden Code gemeint.
14.(C++) *stack*
Gibt den derzeit verwendeten Stack als TUR_STACK zurück.
15.(C++) *getinteger*
Wandelt eine Variable in den Typ TUR_INTEGER um.
Als erstes Argument wird der Variablen namen, von der Variable von der der Inhalt umgewandelt soll, gefordert.
16.(C++) *getboolean*
Wandelt eine Variable in den Typ TUR_BOOLEAN um.
Als erstes Argument wird der Variablen namen, von der Variable von der der Inhalt umgewandelt soll, gefordert.
17.(C++) *getstring*
Wandelt eine Variable in den Typ TUR_STRING um.
Als erstes Argument wird der Variablen namen, von der Variable von der der Inhalt umgewandelt soll, gefordert.
18.(C++) *getbignum*
Wandelt eine Variable in den Typ TUR_BIGNUM um.
Als erstes Argument wird der Variablen namen, von der Variable von der der Inhalt umgewandelt soll, gefordert.
19.(C++) *getbyte*
Wandelt eine Variable in den Typ TUR_BYTE um.
Als erstes Argument wird der Variablen namen, von der Variable von der der Inhalt umgewandelt soll, gefordert.
20.(C++) *getfloat*
Wandelt eine Variable in den Typ TUR_FLOAT um.
Als erstes Argument wird der Variablen namen, von der Variable von der der Inhalt umgewandelt soll, gefordert.
21.(C++) *clear*
Leert den Variablen Stack.
Gibt immer eins zurück.
22. *Stack*
Variablen in turtlescript werden in einem Stack vom Typ TUR_STACK gespeichert.
(C++) Ein TUR_STACK ist eine Struktur(struct) die drei Vektoren(vector) enthält.
Im ersten Vektor, mit den Namen name und den Speichertyp string werden die Namen der Variablen gespeichert.
Im zweiten Vektor, mit den Namen value und den Speichertyp any(aus boost) werden die Inhalte der Variablen gespeichert.
Im dritten Vektor, mit den type und den Speichertyp int werden die Typen(siehe Punkt 2) der Variablen gespeichert.
23.(C++) *TUR_EXCEPTION*
Die TUR_EXCEPTION Klasse in eine Exception-Klasse (die nicht von der Klasse exception(Aus std) abstammt), die gewurfen wird, wenn ein Interpretations Fehler auftritt.
Sie beinhaltet die Funktion what die nichts als Argument fordert und die Fehler Meldung zurückgibt.
Sie beinhaltet die Funktion line die nichts als Argument fordert und Zeilennummer in der, der Fehler passiert ist zurück gibt.

24. Um den Wert von einer Variable einer anderen zuzuweisen wird das Dollarzeichen($) verwendet.
Hinter das Dollarzeichen($) schreibt man den Variablen namen, von der der Inhalt ein eingeblendet werden soll.
Rechenoperationen, wie +, -, *, / oder ähnliches, sind nicht möglich.
Die Zusammenketten von Strings ist auch nicht möglich.
25. Ein turtlescript Datei hat die Dateierweiterung .tur oder .turtle und wird normalerweise mit der UTF-8-Codierung gespeichert.