

Spickzettel für den zdi-BeST-Kurs *Programmieren für Einsteiger*

C boiler plate

```
#include <stdio.h>

int main(void)
{
    return 0;
}
```

Kommentare

```
// Bis zum Ende der Zeile
/* Manuell beendet */
```

Variablen

```
int ganzzahl=42;
float kommazahl=4.2f;
double doppeltegenauigkeit=4.2;
char zeichen='x';
char *string="I bims 1 strink";
int *pi = &ganzzahl;

ganzzahl = ganzzahl + 4;
ganzzahl+=1;
ganzzahl++;
```

Standard-I/O in C

```
printf("Ganzzahl: %d, Kommazahl: %f, Zeichen: %c, String: %s, Pointer (deref.): %i",
      ganzzahl, kommazahl, zeichen, string, *pi);
scanf("%d", &ganzzahl);
// usw...
```

Kontrollstrukturen

Wenn

```
if ( leben == 0 )
{
    beendeSpiel();
}

if ( geld > 9000 )
{
    kaufeItem();
}
else
{
    weine();
}
```

Schleifen

While

```
int geld = 0;
while ( geld < 9000 )
{
    arbeite();
    geld++;
}
kaufeItem();
```

For

```
for (int geld=0; geld < 9000; geld++)
{
    arbeite();
}
```

While fußgesteuert

```
do
{
    arbeite();
    geld++;
}
```

```

}
while(geld <9000);

kaufeItem();

```

Funktionen

```

Typ name(Typ parameter1, Typ parameter2, Typ parameter3, ...)
{
    // Anweisungen ...
}

```

Standard-I/O in C++

```

#include <iostream>
#include <string>

using namespace std;

int main()
{
    string supergeheimesPasswort = "ilovedonaldtrump";
    string passwort;
    cout << "Wie heißt das Paßwort?" << endl;
    cin >> passwort;
    if (passwort != supergeheimesPasswort)
    {
        cout << "Verschwinde" << endl;
    }
}

```

Klassen und Objekte

```

class Klasse
{
    private:
        int attribut;
    public:
        void methode() { };
};

```

```

int main()
{
    Klasse objekt;
    objekt.att ribut = 42;
    objekt.methode();
    method e( );
}

```

Javascript

Ausgaben:

```

textfield1.text = qsTr("Ich habe mich verändert");
console.log("Ich komme auf die Konsole");

```

Variablen

```

var ganzzahl=42;
var kommazahl=4.2;
var zeichen='x';

ganzzahl = ganzzahl + 4;
ganzzahl+=1;
ganzzahl++;

```

Funktionen

```

funciton name(parameter1, paramter2, ...)
{
    // Anweisungen
}

```

QML

Controls

Bibliothek hinzufügen:

```

// ...
import QtQuick.Controls 2.0
// ...

```

Fenster

```
ApplicationWindow{
    width: 640
    height: 480
    title: qsTr("Coolstes Programm")
    property int ZahlAlsAttribut: 45
}
```

Control-Event

```
Button {
    // ...
    onClicked: { funktion(); }
}
```

Sichtbare Mouse-Area

```
Rectangle {
    MouseArea {
        anchors.fill: parent
        onClicked: { funktionBeiKlick(); }
        onEntered: { funktionBeiBeruehrung(); }
    }
}
```

Eigenes Item

```
Item {
    id: name
    property alias eigenschaftAussen : inneresElement.eigenschaftInnen

    Rectangle {
        color: "red"
        id: inneresElement
        eigenschaftInnen: false
    }
}
```