

Ein Einblick in die GODOT Game Engine





# Lesson 01

Godot basics, 'Hello world', and Inspectors



# Regel Nr. 1 – Bei Unklarheiten? Fragen!

Gerne mehrmals <3

## A big part of coding is researching

Learn where to find what you need, what pages are useful

> English often has more results, bigger coding Community

GODOT  
DOCUMENTATION  
[docs.godotengine.org/  
en/stable/index.html](https://docs.godotengine.org/en/stable/index.html)



Godot 3 Recipes



[http://kidscancode.org/godot\\_recipes/3.x/](http://kidscancode.org/godot_recipes/3.x/)

# GDScript

○

○

## Native Language

Native to GODOT, optimized

## Blocks/Blöcke

**Blocks** in GODOT are indent based,  
Not using curly-brackets

## FORMATING MATTERS!

## Dynamische Sprache

Es ist keine kompilierung nötig um den  
Code zu testen; Variablen müssen nicht  
Typisiert werden/Nicht Typ gebunden

·  
·  
·  
·

|  
|  
|  
|

-----

# NOT GDScript

○

○

```
code {  
>|   int variable = 3;  
>|  
>|   void myFunction() {  
>|       code;  
>|       more code;  
>|  
>|       if (variable = 3) {  
>|           code;  
>|       }  
>|   }  
>| }  
}
```

```
code {  
    int variable = 3;  
    void myFunction() {  
        code;  
        more code;  
        if (variable = 3) {code;}  
    }  
}
```

⋮

⋮

# NOT GDScript vs. GDScript

- 
- 

```
code {  
  >| int variable = 3;  
  >|  
  >| void myFunction(){  
  >|   >| code;  
  >|   >| more code;  
  >|   >|  
  >|   >| if (variable = 3) {  
  >|   >|   >| code;  
  >|   >| }  
  >| }  
}
```

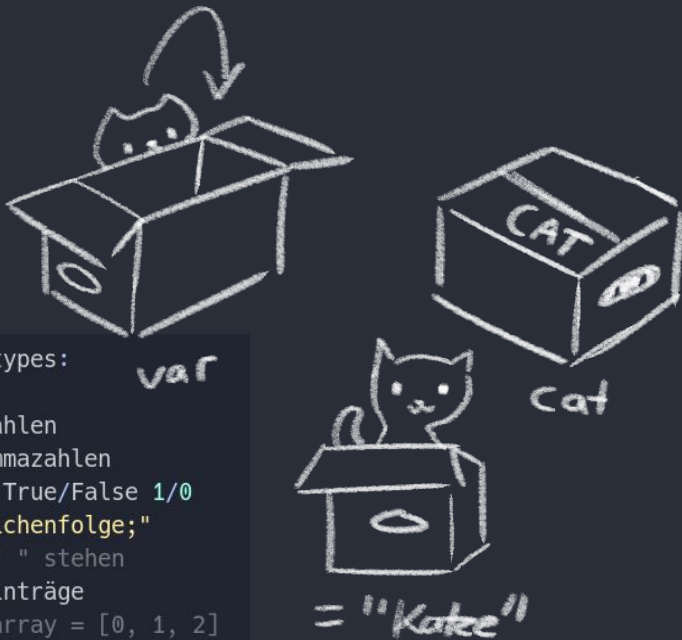
```
code  
var variable = 3  
  
func _my_function():  
  >| code  
  >| more code  
  >|  
  >| if (variable = 3):  
  >|   >| code
```

Formatierung in c++ (links) gehört zum guten Stil/Lesbarkeit, in GDScript (rechts) ist es ein Muss

# GDScript: Variable Initialisieren

○

○



Some supported types:

```
int » Ganze Zahlen
float » Fließkommazahlen
bool » Ja/Nein True/False 1/0
String "Text/Zeichenfolge;"
#Muss zwischen " " stehen
Array Mehrere Einträge
#z.B. var mein_array = [0, 1, 2]
UND ALLE NODES IN GODOT!!!
z.B. var my_node := Sprite.new()
```

C++

```
int x = 1;
```

GDScript

```
var x = 1
#oder
var y:int = 1
```

Dynamisch, keine Typisierung  
notwendig

Kann allerdings mit : typisiert  
werden

# Finde den Formatierungsfehler

○

```
var x = 1  
var y = 2  
var z = 3
```

⋮

⋮

○

—

—



# Finde den Formatierungsfehler

## Leerzeichen

```
var x = 1  
  var y = 2  
var z = 3
```

```
var x = 1  
var y = 2  
var z = 3
```

```
error(4,1): Unexpected indentation.
```

```
func _my_function():  
>| x = 1  
>| y = 2  
>| if (x == 1):  
>| x = 2
```

# Finde den Formatierungsfehler

## Leerzeichen

```
var x = 1  
var y = 2  
var z = 3
```

```
var x = 1  
var y = 2  
var z = 3
```

```
error(4,1): Unexpected indentation.
```

```
func _my_function():  
    x = 1  
    y = 2
```

## Fehlende Einrückung

```
func _my_function():  
>| x = 1  
>| y = 2  
>| if (x == 1):  
>| x = 2
```

```
func _my_function():  
>| x = 1  
>| y = 2  
>| if (x == 1):  
>| >| x = 2
```

```
error(11,1): Expected an indented block after "if".
```

# Finde den Formatierungsfehler

## Leerzeichen

```
var x = 1
var y = 2
var z = 3
```

```
var x = 1
var y = 2
var z = 3
```

error(4,1): Unexpected indentation.

## Fehlende Einrückung

```
func _my_function():
>| x = 1
>| y = 2
>| if (x == 1):
>| x = 2
```

```
func _my_function():
>| x = 1
>| y = 2
>| if (x == 1):
>| >| x = 2
```

error(11,1): Expected an indented block after "if".

## Mischen von Tab und Leerzeichen

```
func _my_function():
>| x = 1
>| y = 2
```

error(9,1): Mixed tabs and spaces in indentation.

```
func _my_function():
>| x = 1
>| y = 2
```

# What will we make?

○

## Small game

We'll go over: Basic coding,  
Sprites, Audio, Music

Doesn't have to be more than  
one room

## Example:

<https://pr0crastigam3s.itch.io/wgj196-strangerecipe-kettle>





# GODOT

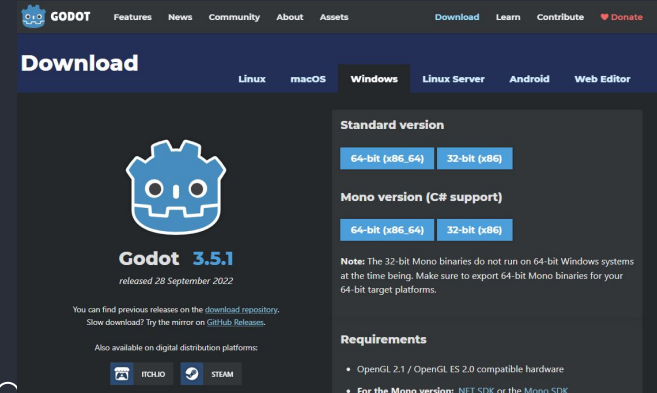
Game engine

## Download GODOT

<https://godotengine.org/>

- Version 3.5.1
- Keine Installation Notwendig
- WICHTIG: Gute Ordnerstruktur

F.ex. Projekte, Assets, Git upload als getrennte Ordner






```
print("Hello World")
```



# Multiple Buttons that give out different text

Task 01





**Two scenes that switch between  
each other through the click of a  
button**

Task 02

