

Fakultät für Betriebswirtschaft
Munich School of Management

Basics in Programming for MMT

Session 1 – Quick Start

The logo for the Munich School of Management (MMT), consisting of the letters 'MMT' in a bold, sans-serif font. The 'M' is green and the 'MT' is blue.

www.mmt.bwl.lmu.de



Scope of the Session

1. What is Programming?

- Why to learn Programming?
- Programming Languages
- Computer vs. Human
- Processing
- Scope of the Lecture

2. Theory

- Processing IDE
- Commands
- Flip Book
- Variables
- “Assignment” vs “Equals Sign”

3. Next

- Session 2

4. Tutorial

- Reference
- Moving Circle
- Move from right to left
- Up and down?
- What else?

What is Programming?

What is Programming?

Why to learn Programming?

- Think of programming as a **tool** which helps you to create things that do not exist yet.
- That **solves problems** you cannot solve with of-the-shelf applications.

What is Programming?

Programming Languages

- Programming Language: Our way to **communicate** with computers.
- Communication requires **rules**. Otherwise, we do not understand each other.

What is Programming?

Computer vs. Human

- Some languages are closer to the human. **Easy to read and understand.**
- Others are closer to the machine. **Easy and fast to process.**

What is Programming?

Processing

- ... is a Java dialect. **Shorter instructions** for the same effect
- Enables us to **create visual output** with less code than Java
- But offers **platform independence** as Java (Mac, Windows, Linux)
- **Eases the transition** to major languages as Java

What is Programming?

Scope of the Lecture

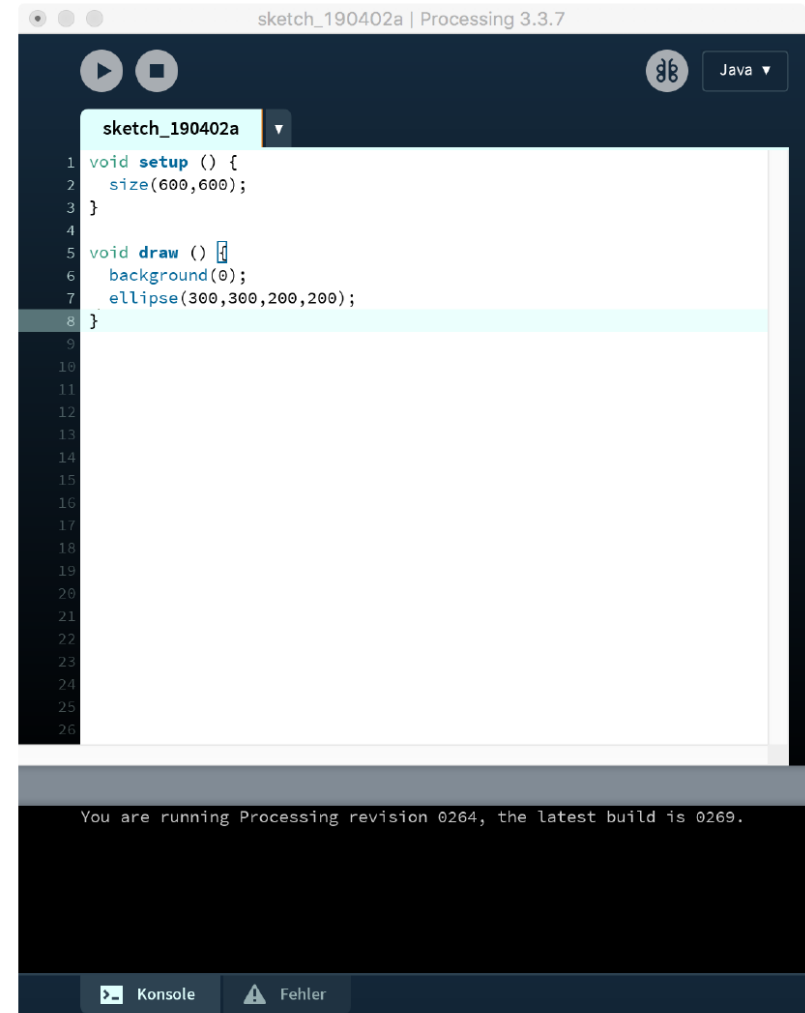
- Basics in **object-oriented** programming
- Basic **logical and structural** skills needed in programming
- **Toolset** to implement software **prototypes**

Theory

Theory

Processing IDE

- **Toolbar:** icons to run/stop your sketch
- **Text Editor:** area to enter your instructions/code
- **Console:** area to output (error-) messages



Theory

Commands (1/2)

What I want to do:

Draw a rectangle!

Print "Hello World!"

Create a random number between 0 and 1

Paint the background black

Open window with size w,h

How I say it:

```
rect(x,y,w,h);
```

```
print("Hello World!");
```

```
random(1);
```

```
background(0);
```

```
size(w,h);
```

Theory

Commands (2/2)

Command name:

Tell computer which command to perform

Arguments:

Values that are processed by the command

Command end:

Semicolon marks end of command

Theory

Flip Book (1/2)

setup ():

Prepare things once

```
void setup () {  
    size(600,600);  
}
```

draw ():

Repeat commands until sketch is closed

```
void draw () {  
    background(0);  
    rect(200,200,200,200);  
}
```

Theory

Flip Book (2/2)

- How to **animate/move** things?
- Movements are **values changing** over time.
- We need a method to get from **static** values to values that we can **manipulate**.

Theory

Variables (1/4)

Declaration:

Give values a keyword/name (variable) to make them “memorize-able”

```
int x;
```

Initialization:

Give these variables initial values

```
void setup () {  
    size(600,600);  
    x = 100;  
}
```

Usage:

Use variables instead of static values (e.g. arguments)

PC looks up the values of variables during execution

```
void draw () {  
    background(0);  
    rect(x,200,200,200);  
}
```

Theory

Variables (2/4)

`int`

`x`

`;`

Datatype:

Variables can be of different types.

Name:

Names can be single letters but also words. Always start with lowercase

End

Theory

Variables (3/4)

- There is no movement right now!
- How to manipulate variables?

Theory

Variables (4/4)

```
int x;
```

```
void setup () {  
    size(600,600);  
    x = 100;  
}
```

Overwrite:

Assign new value to variable

```
void draw () {  
    x = x+1;  
    background(0);  
    rect(x,200,200,200);  
}
```

Theory

“Assignment” vs. “Equal Signs” (1/2)

Math:

x can be removed on both sides.

Calculation does not resolve.

```
x = x+1;  
-> 0 = 1  
-> fatal error
```

Programming:

A new value is assigned to x .

Instructions on the right side are performed first.

Solution is saved as the new value of x .

```
x = x+1;  
-> "x = current value of x  
    plus one"  
-> x is increased by one
```

Theory

“Assignment” vs. “Equal Signs” (2/2)

What is the effect?

```
int x;

void setup () {
    size(600,600);
    x = 100;
}

void draw () {
    x = x+1;
    background(0);
    rect(x,200,200,200);
}
```

Next

Next

Session 2

- Colors
- If A then B!
- Slower speeds?

Tutorial

Tutorial

Reference

1. Go to: processing.org/reference
2. Find command for drawing circles

Tutorial

Moving Circle

1. Change our example code to draw a moving circle
2. What happens to the position?

Tutorial

Moving from right to left

Change the direction of the circle's movement from left-right to right-left

Tutorial

Up and down?

1. How to move the circle up?
2. How to move the circle down?

Tutorial

What else?

What else can we change? Be creative!

References

© Processing