

Multimedia im Netz
Online Multimedia
Winter semester 2015/16

Tutorial 01 – Major Subject



Welcome!

Today's Agenda

- Organization & modalities
- Client side scripting: JavaScript (repetition)
 - Drawing on a canvas
 - DOM access & manipulation
 - Event handling
- Quiz
- Git tutorial

Organization & Modalities

Dates

- **For major subject students (Master)**
Medien-/Informatik, Mensch-Computer-Interaktion:

Day	Time	Tutor
Monday	16 – 18 h	Peter Juras
Monday	18 – 20 h	André Schmidt
Wednesday	14 – 16 h	Tobias Stockinger
Wednesday	18 – 20 h	Thomas Weber

- **For minor subject students (Bachelor)**
Kunst und Multimedia, Pädagogik, Statistik, Lehramt:

Day	Time	Tutor
Wednesday	16 – 18 h	Thomas Weber

Programming Trainings

- Depending on the demand, we offer programming trainings instead of regular tutorials
- Individual consultation is optional. Please contact your tutor to arrange an appointment.

Tutorials – Why are we doing this?

- Application and **immersion** of lecture content
- **Hands-on** activities and discussion
- Opportunity to ask **questions**
- **Preparation** of the upcoming assignment
- **Discussion** of the solutions to exercises

Procedure – Part 1

- Slides and assignment online prior to tutorial
- Due dates for assignments: one **or** two weeks.
Monday to Monday.
- News, updates, and important announcements on the official website:
<http://www.medien.ifi.lmu.de/lehre/ws1516/mmn/>

Procedure – Part 2

- Doing the assignments is completely **voluntary**.
- We recommend you do the assignments.
 - They're fun and challenging.
 - They go beyond the lecture content.
 - They prepare you to pass the exam.
- Assignments are turned in via UniWorX
 - Make sure to check the due date
 - You can't hand in an assignment after the deadline.
 - Individual- or group submission
 - Make sure to do the right assignment:
 - Assignment 01 (HF) = Hauptfach, major subject students
 - Assignment 02 (NF) = Nebenfach, minor subject students

Sample Solutions



- We do **not** provide sample solutions.
- This year, we want to try something else: GitHub.
<https://github.com/MIMUC-MMN/assignments-ws-15-16>
 - There's a git repository to collaborate on sample solutions.
 - We (can) provide a skeleton for the solutions after the deadline
 - Alternatively, we invite a student to push their solution.
 - All members of the GitHub team can improve and discuss the solution.
- Code from the tutorials also goes on GitHub:
<https://github.com/MIMUC-MMN/tutorials-15-16>

Exam

- Date and time: 11.02.2016 10-12 a.m.
- Location: M118 & A240, main building Geschwister-Sholl-Platz
- Most likely open-book.
- They exam includes tasks from both the **lecture and tutorial!**

Semester Plan (subject to change)

Dates	Topics
19.10. & 21.10.	Organization, Client-Side Scripting, git
26.10. & 28.10.	Server-side scripting with PHP – Basics
02.11. & 04.11.	PHP: Sessions and Data Storage
09.11. & 11.11.	PHP & MySQL, AJAX
16.11. & 18.11.	jQuery
23.11. & 25.11.	NodeJS Basics, Express Framework
30.11. & 02.12.	NodeJS: Routing, Database Access
07.12. & 09.12.	NodeJS: Authentication
14.12. & 16.12.	Digital Rights – Watermarking Techniques
21.12. & 23.12.	Christmas Tutorial – Programming Consultation
11.01. & 13.01.	Multimedia Content Description, Introduction to AngularJS
18.01. & 20.01.	AngularJS2, Webcomponents with Polymer
25.01. & 27.01.	Repetition
01.02. & 03.02.	Repetition / Cancelled (depending on final exam date)

News, Readings, Q&A via Twitter

https://twitter.com/MMN_WS1516

Startseite Über uns Twitter durchsuchen Hast Du einen Account? Anmelden

MMN LMU
@MMN_WS1516
Munich, Germany
medien.ifi.lmu.de/lehre/ws1415/m...

Neu bei Twitter?
Melde Dich jetzt an, um Deine eigene, personalisierte Timeline zu erhalten!
[Registrieren](#)

Vielleicht gefällt Dir auch -
Aktualisieren

- Sebastian Loehmann** @loehmuc
- uxcite GmbH** @uxcite
- Alexander Wiethoff** @awiethoff
- Medieninfo München** @mimuc
- Fabian Hennecke** @FabianHennecke

TWEETS 35 FOLLOWER 40 [Folgen](#)

Tweets Tweets & Antworten

MMN LMU @MMN_WS1516 · 2 Std.
We will continue with this Twitter account during the winter term 15/16. We'll also try to recommend readings and interesting content.

MMN LMU @MMN_WS1516 · 22. Jan.
Next week, there will be a last lecture. We will have guests from Google Munich talking about User Experience Design. Please come!

MMN LMU @MMN_WS1516 · 22. Jan.
The podcasts of today's lecture ore online.
medien.ifi.lmu.de/lehre/ws1415/m...

MMN LMU @MMN_WS1516 · 21. Jan.
There will be one more lecture next week (Jan 29), with external guest speakers!

MMN LMU @MMN_WS1516 · 21. Jan.

Client Side Scripting: JavaScript + HTML5 = ♥

HTML5

- HTML5 introduced a couple of new features:
 - New Elements:
 - **<canvas></canvas>**
 - **<audio></audio>**
 - **<video></video>**
 - More: http://www.w3schools.com/html/html5_new_elements.asp
 - Form features (examples):
 - Wildcards
 - Validation
 - Drag and Drop



HTML5: Document Structure

```
<!DOCTYPE html>  
<html lang="de">  
  
<head>  
  <meta charset="UTF-8" />  
  <title>HTML5 Structure</title>  
</head>  
  
<body>  
</body>  
  
</html>
```


HTML5: Canvas

- The <canvas> element is a ***container*** that's embedded into the HTML markup

```
<canvas width="400" height="400" style="border:1px solid #000000;">
  Browser does not support the canvas tag.
</canvas>
```

- HTML5 uses the *immediate mode* for the <canvas> element and not the *retained mode*.

HTML5: Context

- The drawing is done via JavaScript. In order to draw, the context is required: **getContext () ;**
- The context is an object that has its own attributes and methods that you can use to draw on the canvas.
- There are two types of contexts:
 - 2D
 - 3D (WebGL)

JavaScript

- JavaScript is a dynamic scripting / programming language
- Code is interpreted by the web browser
- Code can be embedded into HTML

```
<script>  
/*  
    Here goes your script!  
*/  
</script>
```

- Alternatively, the code can be imported from a file
<script src="myScript.js"></script>

DOM (Document Object Model)

- The DOM references every element and its content in an HTML (or XML) document.
- Elements, contents and structure can be modified:
 - **document**: Content of the browser window
 - **getElementById ()** : gets an HTML element with a unique identifier
 - **getElementsByTagName ()** : gets all elements by a specific tag
 - **querySelector ()** : Find first node that matches a CSS selector
 - **Node.firstChild** : returns the first child node
 - **Node.nodeValue**: gets or sets the value of a node
- <http://wiki.selfhtml.org/wiki/JavaScript>
<http://de.selfhtml.org/javascript/index.htm>

DOM and JavaScript

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"/>
  <title>HTML 5</title>
</head>
<body>
<canvas id="canvas" width="400" height="400"
  style="border:1px solid #c3c3c3;">
  Your browser does not support the HTML5 canvas tag.
</canvas>

<script>
  var canvas = document.getElementById("canvas");
</script>
</body>
</html>
```

Retrieve the context of the canvas

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8"/>
  <title>HTML 5</title>
</head>
<body>
<canvas id="canvas" width="400" height="400"
  style="border:1px solid #c3c3c3;">
  Your browser does not support the HTML5 canvas tag.
</canvas>

<script>
  var canvas = document.getElementById("canvas");
  var context = canvas.getContext("2d");
</script>
</body>
</html>
```

JavaScript and Canvas

- Colors, strokes, fills (attributes)
 - fillStyle
 - strokeStyle
- Draw rectangles (functions)
 - rect();
 - fillRect();
 - strokeRect();
- Draw images onto the canvas
 - drawImage()
- More functions:
http://www.w3schools.com/tags/ref_canvas.asp

Draw a Rectangle

```
...  
<script>  
  var canvas=document.getElementById("canvas");  
  var context = canvas.getContext("2d");  
  context.fillStyle="#00ff00";  
  context.fillRect(0,0, 150, 100);  
  
</script>  
</body>  
</html>
```


The arc() Function

- Create circles or parts of circles.

- Signature:

```
context.arc(x, y, r, sAngle, eAngle, counterclockwise) ;
```

- x,y: coordinates on the canvas

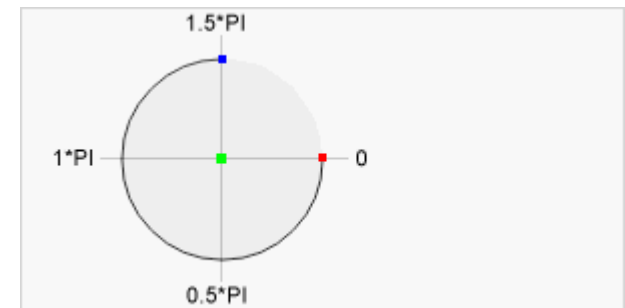
- r: radius of the arc

- sAngle: starting angle (rad)

- eAngle: end angle (rad)

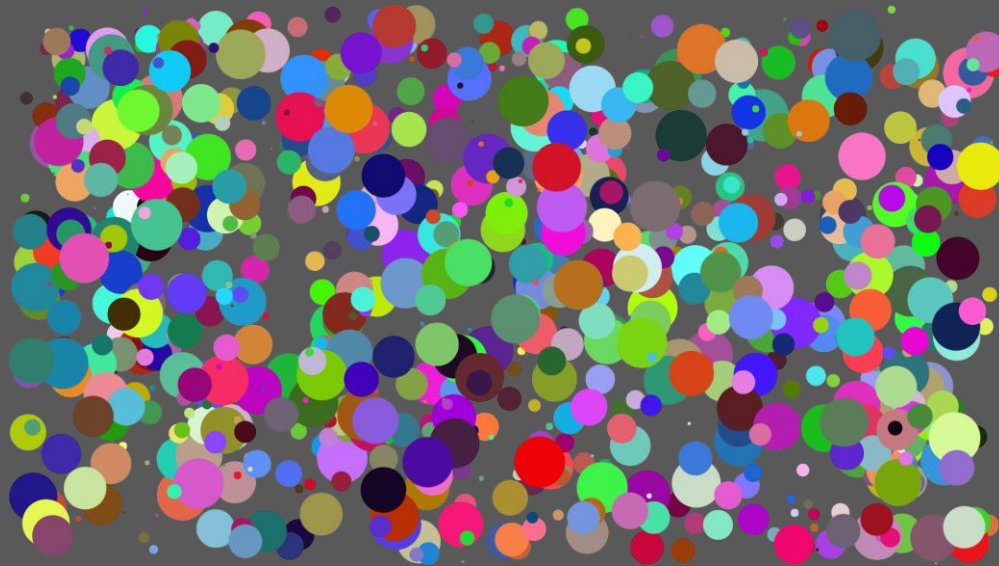
- counterclockwise: flag to determine the direction, default: false

- You need to call context.fill() or context.stroke() to actually see the drawing



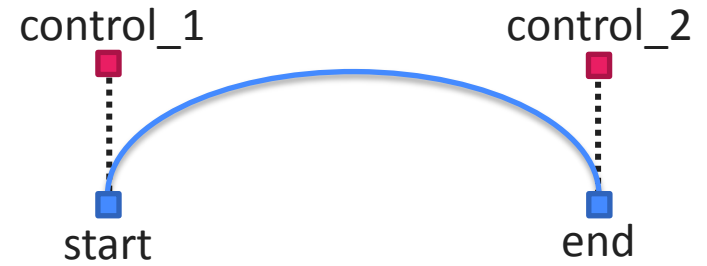
Break Out Task

- Generate a random image with the canvas object
- Example: A random number of circles of varying position, size, and color:



- Take 15 Minutes time

Bezier Curves



- Two control points to adjust the curvature

- Start with:

```
context.moveTo ( x, y ) ;
```

- Create the curve:

```
context.bezierCurveTo (  
    control_1x, control_1y,  
    control_2x, control_2y,  
    end_x,    end_y) ;
```

- Curves with only one control point:

```
context.quadraticCurveTo (  
    control_1x, control_1y, end_x, end_y) ;
```

User Interaction - EventListener

- Basically all HTML elements can trigger certain events
- JavaScript can listen for and consequently handle such events
- EventListener attribute:

```
<script>

function showTime () {
    document.getElementById ("output") .innerHTML =
        new Date () .toString () ;
}

</script>
<button onclick="showTime () ">Show Time!</button>
<div id="output"></div>
```

EventListener via JavaScript

```
<button id="timeButton">Show Time!</button>
<div id="output"></div>

<script>

    function showTime() {
        document.getElementById("output").innerHTML =
            new Date().toString();
    }

    document.querySelector('#timeButton')
        .addEventListener('click', showTime);

</script>
```

Events on the Canvas

Use case: We want to draw a circle wherever the user clicks

```
<canvas id="maincanvas"
  width="800" height="600"
  style="border: 1px solid gray"></canvas>
<script>

  function drawCircle(event) {
    var canvas = event.target;
    var context = canvas.getContext('2d');

    var x = event.clientX - canvas.offsetLeft;
    var y = event.clientY - canvas.offsetTop;

    var radius = Math.random()*50;
    context.beginPath();
    context.arc(x, y, radius, 0, 2*Math.PI);
    context.fill();
  }

  document.getElementById('maincanvas').onclick = drawCircle;
</script>
```

Quiz Part 1

1. Name 3 elements that are 'new' in HTML5!
2. Which document type is correct for HTML5:
 - a) `<!DOCTYPE html>`
 - b) `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0//EN" "http://www.w3.org/TR/html5/strict.dtd">`
 - c) `<!DOCTYPE HTML5>`
3. onclick is an...
 - a) element
 - b) event attribute
 - c) style attribute
4. Which attribute of `<script>` is no longer required?
rel | href | src | type

Quiz Part 2

1. Which **parameters** does the `arc()` function take?
2. How do you draw a **semi-circle** on a canvas?
3. What's a **potential** error source here?
 - `document.getElementById('#canvas');`
 - `document.querySelector('canvas').length;`
 - `canvas.onclick = drawShape();`
4. How many control points does a Bezier curve have?
5. How do you define the start point of a Bezier curve?

Link Collection

- <http://caniuse.com/>
- <http://www.w3schools.com/js/default.asp>
- <https://stackoverflow.com/>
- <https://www.coursera.org/learn/html-css-javascript>
- <http://www.html5rocks.com/>

Thanks!

What are your questions?

Let's begin with the Assignment!

- Download the assignment sheet
- Start with task 1
- You can collaborate with your neighbor
- Turn in the assignment by October 26, 12:00 noon via UniWorX

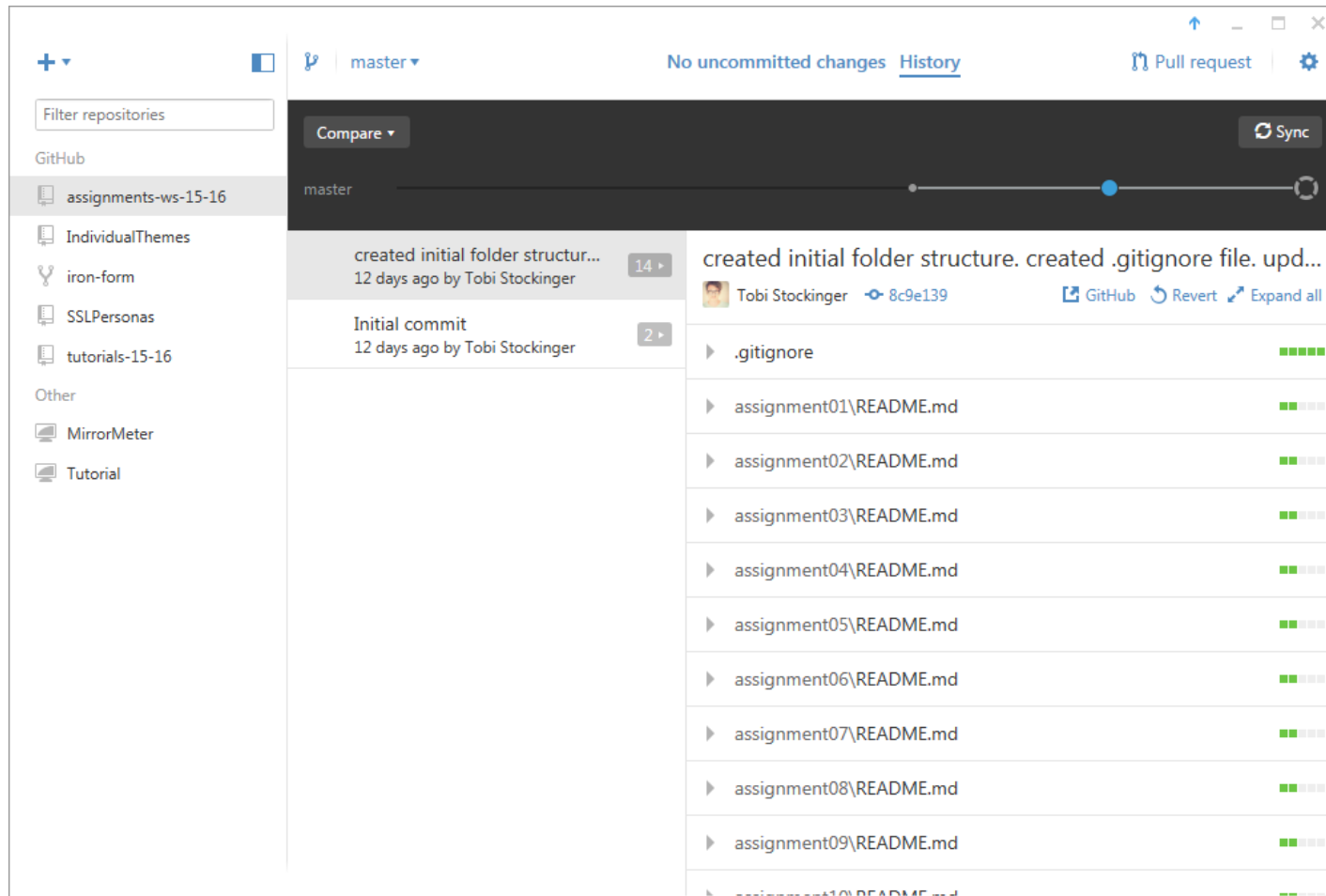
Introduction to GIT

Heavily inspired by Roger Dudler

<https://rogerdudler.github.io/git-guide/>

Recommendation: GitHub app

<https://desktop.github.com/>



Setup

- Download git
 - Mac - <https://git-scm.com/download/mac>
(included in the Xcode command line developer tools)
 - Windows: <https://git-for-windows.github.io/>
 - Linux: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
- On CIP pool machines git is already installed.

Create a new Repository

1. Create a new directory inside a folder of your choice
 2. Open a terminal / shell and navigate to the folder, e.g.
`cd ~/myrepository`
 3. Initialize the **local** repository:
`git init`
- Not (really) necessary for the collaborative solutions in this course

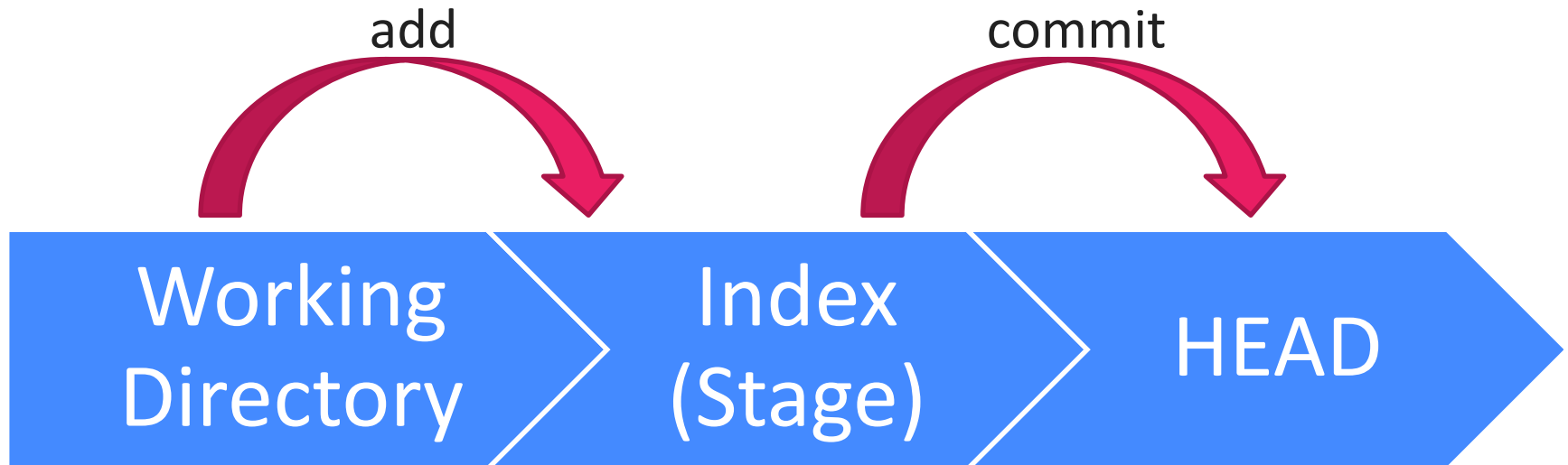
Clone a repository

1. Find out the URL from a (repository), e.g.
<https://github.com/MIMUC-MMN/assignments-ws-15-16>
 2. Clone the repository

```
git clone git@github.com:MIMUC-MMN/assignments-ws-15-16.git
```
- This implies the following SSH syntax:
 - Username: git
 - Server URL: github.com
 - Path to Repository: MIMUC-MMN/assignments-ws-15-16.git
 - To make this work, you need to create an SSH key
 - The GitHub app is helpful here.

<https://rogerdudler.github.io/git-guide/>

Workflow



Add & Commit

- All files that you changed need to be **added to the index**
- Once you finalized all the changes, you **commit** them

1. Adding a file / all files:

```
git add myfile.txt
```

```
git add *
```

2. Commit your changes:

```
git commit -m "My commit message describes what I did".
```

Pushing to Remote Server

- The changes are not in the HEAD of your working copy
- You can send them to the remote server, e.g. on GitHub

```
git push origin master
```

- origin: name of the server
- master: name of the branch.

In many projects, you can't push to master, so you need to create a different branch first.

- You can add multiple remote servers for one project
- ```
git remote add another_server URL
```

<https://rogerdudler.github.io/git-guide/>

# Branching

- Create a new branch and switch to it:  
`git checkout -b new_branch`
- Switch back to master  
`git checkout master`
- Delete a branch  
`git branch -d new_branch`
  
- A branch only becomes visible to others if you push it to a remote server!

# Update and Merge

- Update local files with the server version  
`git pull`
- Integrate changes from another branch:  
`git merge <branchname>`
- Compare branches  
`git diff <branch1> <branch2>`