

Workshop: Frontend Development

Mit Astrid Schwarz
Qualcomm – Wiener Töchtertag

Technologien

- Git
- Figma
- Html
- Css
- Javascript

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly  
-- OPERATOR CLASSES ----  
types.Operator):  
on X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
context):  
context.active_object is not
```

Technologien

- Git
- Figma
- Html
- Css
- Javascript

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly  
-- OPERATOR CLASSES ----  
types.Operator):  
on X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
context):  
context.active_object is not
```


Übung mit Git und Terminal

- Holt euch das Repository auf:
https://github.com/StridS/Wien_ToechtersTag
- Kreiert euch euren eigenen Branch: „group-<Name1>_<Name2>“
- Installiert die Module in „Wien_ToechtersTag/package.json“
- Ändert das „author“-Property im package.json

Terminal/Command Line – Cheat Sheet

Für Mac:

Öffnen:

-  + Leertaste
- „terminal“ eingeben

Ordner und Dateien im Pfad: „ls“

Derzeitiger Pfad: „pwd“

- Liefert z.B. „/home/<name>/Desktop“

Ordner wechseln: „cd <Pfad>“

- Z.B. „cd ../Documents“

Pfad-Eigenschaften

- „..“ = Übergeordneter Ordner
- „.“ = Derzeitiger Ordner
- „~“ = Heimatverzeichnis
- „/“ = Root-Verzeichnis

Kombinationen:

- „/home/astrid/Documents/my_folder“
- „../Desktop/another_folder“

Für Windows:

Öffnen:

- Starttaste (田)
- „cmd“ eingeben

Ordner und Dateien im Pfad: „dir“

Derzeitiger Pfad: „echo %cd%“

- Liefert z.B. „C:\Users<name>\Desktop“

Ordner wechseln: „cd <Pfad>“

- Z.B. „cd ../Documents“

Pfad-Eigenschaften

- „..“ = Übergeordneter Ordner
- „.“ = Derzeitiger Ordner
- „cd %userprofile%“ = Heimatverzeichnis
- „/“ = Root-Verzeichnis

Kombinationen:

- „C:\Users\astri\Downloads“
- „../Desktop/another_folder“

Git – Cheat Sheet

„git clone <url>“: Ladet das Repository herunter

„git branch“: Liste der existierenden Branches

„git branch <neuer branch name>“: Kreiert neuen Branch

„git checkout <branch name>“: Geht zu neuem Branch

„git add“: „Staged“ alle neuen Änderungen

„git commit -m‘message‘“: Checkt gestagte Änderungen ein

„git push“: Ladet Änderungen ins Repository hoch

„git status“: Zeigt derzeitigen Stand an

Weitere Ressourcen

- <https://training.github.com/downloads/de/github-git-cheat-sheet/>
- https://ndpsoftware.com/git-cheatsheet.html#loc=local_repo
- <https://ohshitgit.com/de>
- Or just google it...

Technologien

- Git
- Figma
- Html
- Css
- Javascript

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES ----  
  
types.Operator):  
on X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```


Übung mit Figma

- Designed eure eigene Website auf <https://www.figma.com/>
- Importiert die .fig Datei in euer Projekt
- Mobile oder Desktop ist euch überlassen
- Anforderungen
 - Mindestens ein Button
 - Mindestens ein Bild
 - Mindestens ein Kartenelement
- Bonus: Fügt eine Prototyping-Interaktion dazu
- Ressourcen für Medien:
 - <https://fonts.google.com/>
 - <https://www.flaticon.com/>
 - <https://www.pexels.com/de-de/>

Technologien

- Git
- Figma
- **Html**
- Css
- Javascript

Übung mit HTML

- Implementiert euer Design in HTML
- Verwendet dafür entweder VS Code oder <https://app.edublocks.org/>
 - Ihr könnt dafür einen neuen Ordner im Projekt anlegen
- Achtet auf die richtige Struktur
 - Was wäre im Header?
 - Was im Main?
 - Wann verwendet man am besten H1/H2/p/...

HTML - Cheat Sheet - Grundgerüst

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

HTML – Cheat Sheet - Tags

Grundstruktur

- `<header>...</header>`, `<main>...</main>`, `<footer>...</footer>`
- `<section>...</section>` - Für Unterteilungen im Main

Schrift

- `<h1>Titel</h1>` - Nur ein Mal in der Seite (z.B. Name der App)
- `<h2>Überschrift</h2>` - Je nach Level gibt es auch h3, h4, ...
- `<p>Text</p>` - Übliche Form von Standard-Text
- `mitten im Satz` - Inline Text (z.B. wenn was Unterstrichen ist)
 - Kann unabhängig von restlichen Text gestyled werden
 - Es gibt auch vor-formatierte Varianten (`<i>kursiv</i>`, `fett`, ...)
- `<a>Link` - Text mit Klick-Funktion
 - wird verwendet, um auf neue Seite zu kommen
 - Attribut „href“ darf nicht fehlen damit es auch funktioniert!

HTML – Cheat Sheet - Tags

Interaktion

- `<button>Tu Etwas</button>` - Nutzer:in erwartet sich ein Ereignis auf Click
 - Attribute: „type“, „title“, („onclick“)
- `<form>...</form>` - Dateneingabe, sammelt 1 oder mehrere Informationen
 - Attribute: „action“, „method“
- `<input>` - Dateneingabe für bestimmte Information (Self-Closing Tag!)
 - Input-Typ kann variieren je nach Information
 - Attribute: „type“, „name“, „value“, „placeholder“

Misc

- `` - Stellt ein Bild dar
 - Attribute: „src“, „alt“
- `<div>...</div>` - Regulärer Container für Struktur und Styling

Weitere Ressourcen

- <https://developer.mozilla.org/en-US/>
- <https://www.w3schools.com/>
- <https://wiki.selfhtml.org/wiki/SELFHTML>

Technologien

- Git
- Figma
- Html
- **Css**
- Javascript

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly  
-- OPERATOR CLASSES ----  
types.Operator):  
on X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
context):  
context.active_object is not
```

Übung mit CSS

- Implementiert euer Design in CSS
 - Erstellt eine neue .css-Datei und bettet sie in euer .html ein
 - `<link rel="stylesheet" href="style.css" />` <- HTML
- Überlegt, wie ihr eure Schriftart und Medien importiert
 - Lokal im assets-Ordner oder importiert via url()?
- Empfohlen: Legt am besten Klassen-Attribute für eure Elemente an!
 - `<header class="header">` <-HTML
 - `.header {...}` <- CSS

Weitere Ressourcen

- <https://css-tricks.com/>
- Ansonsten dieselben wie für HTML...

Technologien

- Git
- Figma
- Html
- Css
- Javascript

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly  
-- OPERATOR CLASSES ----  
types.Operator):  
on X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
context):  
context.active_object is not
```

Übung mit JavaScript - Gemeinsam

- Macht euren Button interaktiv!

Verwendete Medien

- placeholder_vr.mp4:
 - <https://www.pexels.com/video/virtual-reality-855770/>
- night_sky-wide.jpg:
 - <https://www.pexels.com/de-de/foto/kiefern-unter-sternenhimmel-1539225/>
- meteor.png:
 - https://www.flaticon.com/free-icon/meteor_925202
- night_sky-slim.jpg:
 - <https://www.pexels.com/de-de/foto/silhouette-von-menschen-und-haus-unter-mond-2820966/>
- image.svg:
 - <https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@48,400,0,0>