

# Stichwortverzeichnis für Modulabschlussprüfung 3041 Informatik

Kapitel	Stichwort	Beschrieb	Skript	Vorlesung	Übung
1	GitHub	Bedienung GitHub	1		
2	Plotten	Einführung Matplotlib	2,2	01_example.py	02_Aufgabe_1.py 02_Aufgabe_2.py
2	Numpy	Einführung Numpy, Arrays	2,3	01_example.py, 04_numerik.py 05_arrays.ipynb	02_Aufgabe_1.py 02_Aufgabe_2.py
2	Numpy	Mathematische Funktionen und Konstanten in Numpy	2,4	02_plotten.ipynb, 03_sinuskurve.py, 06_numpyplot.py, 07_subplot.py	02_Aufgabe_2.py Kap. 9
2	Plotten Funktionen	Kombi numpy und matplotlib	2.5.1	02_plotten.ipynb, 03_sinuskurve.py, 06_numpyplot.py, 07_subplot.py	02_Aufgabe_1.py 02_Aufgabe_2.py Kap. 9
2	Plotten mehrere Fkt.	mehrere Fkt. in einem Plot	2.5.2	06_numpyplot.py	
2	Plotten Subplots	mehrere Fkt. in einem Fenster in einzelnen Plots	2.5.3	07_subplot.py	Kap. 10
2	Plotten von Figuren	Bogen, Rechtecke, Kreise...	2,6	figuren.py	02_Aufgabe_1.py Kap. 10
	Plotten von random-Fkt.				02_Aufgabe_1.py
2	Plotten Meshs von Fkt.	np.meshgrid(x, y), plt.pcolormesh(X, Y, Z)			02_Aufgabe_2.py
2	Plots speichern	plt.savefig("passender_dateiname.png")			02_Aufgabe_1.py 02_Aufgabe_2.py
2	Objektorientierung	Definition	3,1		
3	Klassen	Klassen anlegen	3,2	punkt.py, 02_temperatur.py Kap. 4, 5, 6, 7	03_Aufgabe_1.py 03_Aufgabe_2.py Kap. 4, 5, 6, 7, 8, 9, 12
3	Instanzen	Instanz anlegen	3,3	01_beispiel.py, 02_temperatur.py, Kap. 4, 5, 6, 7	03_Aufgabe_1.py 03_Aufgabe_2.py Kap. 4, 5, 6, 7, 8, 9, 12
3	Konstruktor	__init__ Klasse automatisch aufrufen	3.4, 3.7	punkt.py, 02_temperatur.py Kap. 4, 5, 6, 7	03_Aufgabe_1.py 03_Aufgabe_2.py Kap. 4, 5, 6, 7, 8, 9, 12
3	Destruktor	del Klasse löschen	3,5		
3	Setter- und Getter-Methoden	Attribute verändern und holen ohne direkten Zugriff	3,8	02_temperatur.py Kap. 4	03_Aufgabe_2.py Kap. 7, 8, 9
3	Set und Get Werte prüfen	Warnung bei Nichterfüllung (z.B. Koordinatenbereich, falscher Datentyp)		02_temperatur.py Kap. 4, 5	03_Aufgabe_2.py Kap. 4, 5, 8, 9
3	Property Attribute	Get und Set automatisch aufrufen	3.9, 3.10	02_temperatur.py	03_Aufgabe_2.py
3		if name == "main":		punkt.py	
4	MM Magische Methoden	Definition			
4	MM Typenumwandlungen	__str__, __float__, __int__, __bool__ ...	4,1	temperatur.py, tempervice.py, vector2.py Kap. 5, 6, 7	04_Aufgabe_1.py Kap. 5
4	MM Vergleichsoperatoren	>, <, =, !=	4.2.1	temperatur.py, tempervice.py, vector2.py Kap. 6	04_Aufgabe_1.py Kap. 5
4	MM Math. Operatoren	add, sub, mul	4.2.2	vector2.py	04_Aufgabe_1.py
5	UML	Objektoreintierung und UML -> Defintion	5,1		
5	UML Klassendarstellung	Abbildung Klassen unabhängig von Programmiersprache	5,2		
5	Objektorientierte Beziehungen	Assaiation (hat), Aggregation (besteht aus), Komposition (starke besteht aus)	5,3	01_HatBeziehung.py	05_Aufgabe_1.py
5	Vererbung	Objektorientierte Beziehungen (Umfang, Fläche etc.)	5.3.4	02_Vererbung.py, figur.py	05_Aufgabe_1.py
6	PyQt5	GUI-Prgrammierung Einleitung	6.1, 6.2	00_window.py	
6	Widgets GUI	wichtigste GUI-Elemente	6,3	alle Kap. 7, 8	06_Aufgabe_1.py Kap. 7, 8
6	Layout GUI	Layout erstellen und mit Widgets verknüpfen	6.4, 6.5	alle Kap. 7, 8	06_Aufgabe_1.py Kap. 7, 8
6	BoxLayout	HBoxLayout und VBoxLayout	6.5.1	01_hboxlayout.py, 02_widgets.py, 07_signal_slot.py	
6	QGridLayout		6.5.2	03_gridlayout.py	
6	QFormLayout	Formulare mit Label	6.5.4	04_formlayout.py	06_Aufgabe_1.py Kap. 7
6	MixLayouts	Kombination der verschiednen Layouts	6.5.5	05_multilayout.py	
6	Menubar	Einfache Menues und Verschachtelung	6,6	06_menu.py	06_Aufgabe_1.py Kap. 7
6	Signale und Slots für Widgets	pressed(), textChanged(), clicked()	6.7, 6.8, 6.10	07_signal_slot.py Kap. 7, 8	06_Aufgabe_1.py Kap. 7, 8
6	Signale für Menues	triggered()	6,9	06_menu.py	06_Aufgabe_1.py Kap. 7
6	Zugriff auf Widget-Inhalte		6,11	07_signal_slot.py Kap. 7, 8	06_Aufgabe_1.py Kap. 7, 8
6	GUI schliessen	exit(0)		06_menu.py	06_Aufgabe_1.py Kap. 7
7	QMessageBox	Informationsbox, About, Warnungen, kritische Fehler, Fragen	7,1	dialog.py, dialog2.py, dialog3.py	Kap. 8, 9
7	Dateidialoge	Datei über Dialog auswählen, abspeichern...	7,2	dialog2.py, dialog3.py	07_Aufgabe_1.py Kap. 8
7	Eingaben abfragen	spez. Form für Eingabe von Werten (Zahlen, Auswahl und Zeichenketten)	7,3	dialog2.py, dialog3.py	
7	Farbdialog	Farben aus Palette auswählen	7,4	dialog2.py, dialog3.py	
7	Fontdialog	Schrift definieren	7,5	dialog2.py, dialog3.py	
7	Druckdialog	Druck- und Druckvorschau	7,6		
7	Browseranzeige	QDesktopServices.openUrl(QUrl())			07_Aufgabe_1.py
8	Qt-Designer	Arbeiten im Qt-Designer -> speichern als .ui	8.1, 8.2	alle.ui	
8	Qt-Designer-File einbinden	loadUi	8,4	alle.py	08_Aufgabe_1.py
8	Qt-Designer-Elemente ansteuern		8.5, 8.6	alle.py	08_Aufgabe_1.py
8	Webbrowser erstellen	self.webEngineView.load(Qurl())	8,7		08_Aufgabe_1.py

Kapitel	Stichwort	Beschrieb	Skript	Vorlesung	Übung
9	Plotten mit Python Script	ähnlich Kapitel 2 (dort Jupyter Lab), mit und ohne QtDesigner	9.1, 9.2		09_Aufgabe_2.py
9	Daten mit Elementen einlesen	QLineEdit, QSlider oder QRadioButton	9.3		09_Aufgabe_2.py
10	Koordinatentransformation	pyproj, mit direkter Eingabe	10.1.1	geometrie.py	10_Aufgabe_1.py
10	Koord. Trans. mit CSV	Koordinaten aus CSV umwandeln	10.1.2	plz.py, geometrie.py	
10	Shapely Objektdarstellung	Darstellung Vektordaten mit WKT-Strings	10.1.3	geometrie.py, geometrie2.py	10_Aufgabe_1.py
10	Shapely räumliche Abfragen	contains, within, intersects, crosses, touches, equals...	10.1.4	geometrie2.py	
10	Shapely mehrere Polygone + Operationen	Differenz, Schnittmenge, Vereinigung...	10.1.5		
10	Delaunay Triangulierung		10.1.6		
11	Folium Kartendarstellung	bestehende Karten mit Folium darstellen		11a_GeoBasics-folium.ipynb	11_Aufgabe_2.ipynb
11	Folium Marker	Marker auf Karte darstellen		11a_GeoBasics-folium.ipynb	11_Aufgabe_2.ipynb
11	Folium Daten aus CSV	Marker aus CSV-Daten		11a_GeoBasics-folium.ipynb	
11	Folium Marker Cluster	Zusammenfassung Marker zu Wolke bei Überlagerung		11a_GeoBasics-folium.ipynb	
11	GeodataFrame aus CSV erstellen	Spalten auswählen, benennen, exportieren		11b_GeoBasics-folium.ipynb	11_Aufgabe_1.ipynb
11	GeodataFrame aus SHP erstellen	Daten betrachten, Plotten		11b_GeoBasics-folium.ipynb	11_Aufgabe_1.ipynb
11	Folium Datendarstellung in Grundkarte	Darstellen, Styling		11b_GeoBasics-folium.ipynb	
11	GeoDataFrame Attributabfragen	ascending, descending, query, shape (Anzahl Features)		11b_GeoBasics-folium.ipynb	11_Aufgabe_1.ipynb
11	GeoDataFrame Räumliche Abfragen	distance, contains		11b_GeoBasics-folium.ipynb	11_Aufgabe_2.ipynb
11	GeoDataFrame Histogramme			11b_GeoBasics-folium.ipynb	
11	Koordinatensystem bestimmen	gdal.Datensatz.crs			11_Aufgabe_1.ipynb
11	GeodataFrame aus JSON erstellen	direkte Integration aus Web			11_Aufgabe_2.ipynb
12	Defaultkartendatensätze laden	cartopy und matplotlib		12_Projektionen_Vektordaten.ipynb	
12	Projektionen	PlateCarree(), Mollweide(), Orthographic(), Mercator(),		12_Projektionen_Vektordaten.ipynb	12_Aufgabe_1.ipynb 12_Aufgabe_2.ipynb
12	Projektion Spezialfall	GOOGLE_MERCATOR hat keine Parameter (Start-lat etc. definiert)		12_Projektionen_Vektordaten.ipynb	12_Aufgabe_2.ipynb
12	Winkeltreue prüfen	Tissot Inikatrix -> Kreise = winkeltreu		12_Projektionen_Vektordaten.ipynb	
12	Punkte darstellen	aus lat, lon Punkte darstellen und stylen		12_Projektionen_Vektordaten.ipynb	
12	Linien generieren	Gerade, gekrümmt		12_Projektionen_Vektordaten.ipynb	
12	Ausschnitt festlegen	ax.set_extent([5.9, 10.5, 45.8, 47.8])		12_Projektionen_Vektordaten.ipynb	12_Aufgabe_2.ipynb
12	Geopandas	Daten als gpd einlesen, darstellen, random stylen (Kap. 11)		12_Projektionen_Vektordaten.ipynb	12_Aufgabe_1.ipynb 12_Aufgabe_2.ipynb
12	iloc	geometry = kantone.iloc[0][self.k]["geometry"]		12_Projektionen_Vektordaten.ipynb	12_Aufgabe_1.ipynb 12_Aufgabe_2.ipynb