**Section 21: Introduction to GUIs**

**30.03.**

**147. Interact Functionality with GUIs**

In Jupyter Notebooks the GUI can be generated with ipywidgets

Interact can be used as a decorator or a wrapper for functions

Depending on the type of arguments, different widgets are created

Values or widgets instances can be passed to arguments for interact

For example: bool – checkbox, text – textbox, numeric values – slider, lists / dicts – drowdown

If some arguments need to be fixed – fix(argument) will set the argument

Interactive is useful to easily access the properties of the widgets

The return value won’t be displayed automatically, thus the display function is needed

The arguments and return values are accessible through. kwargs and .result

**148. GUI Widget Basics**

The widgets are stateful, thus their state can be accessed inside other cells

A widget is displayed with display() and closed with close()

The proprieties of widgets can be retrieved and set in real time even after the widget execution

To access all the properties of a widget use: widget\_name.keys

Two widgets can be linked such that they display the same value, or other attributes are linked

This link is done with jslink

The unlinking is done with .unlink

**149. List of Possible Widgets**

There are a lot of widgets such as:

Numeric: sliders, range sliders, progress bars, numeric text boxes

Boolean: toggle button, checkbox, valid / invalid

Selection: dropdown, radio buttons, selection slider, selection range slider, toggle buttons

String: text, text area, label, html

Images

Widgets can have linked special events like:

On click, on submit and so on

Widgets have callbacks too

Trailet are eventful and are the properties of widgets

These trailets on different widgets can be linked with traitlets.link

The unlinking is done with unlink

There are more advanced widgets as:

Animations, date picker, color picker, accordion, tabs etc.

**150. Widget Styling and Layouts**

Widgets have a layout that can be manipulated with CSS and refer to position and margins

The layout attributes are the same for all the widgets

The style is particular to each independent type of widget

The layout is accessible through widget\_name.layout.

The layout / parts of the layout from some widgets can be used in another widgets

There are some predefined styles that gives the colors: primary, success, info, warning, danger

For the button this predefined style can be set in button\_style attribute

More individual style characteristics can be set for widget\_name.style.

All the possible attributes for style can be displayed with widget\_name.style.keys