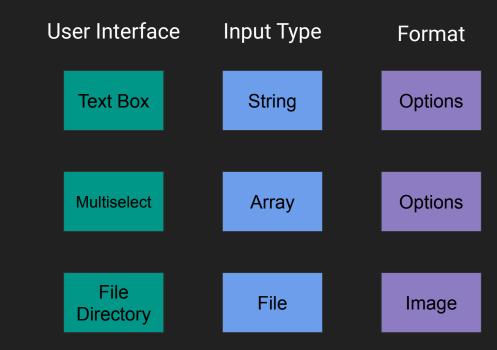# User Interfaces at NCATS

Sunny Yu
Dec 2023

NCATS Needs

- Support a diverse group of users
  - Programmers
  - Technical
  - Advanced domain experts
  - Domain experts
- Integrate with existing architecture and tooling
  - Data storage resources, cloud compute, on-premise HPC
- Build complex and reusable workflows

Inputs/Outputs vs. User Interface

- The user interface standard is used to provide input or application specific context to an input value.
- Distinct context and utility of input/output, format, and user interface fields.

User Interface	Input Type	Format
Text Box	String	Options
Multiselect	Array	Options
File Directory	File	Image

User Interface Types

- How can ensure portability across different user interfaces?

Text	Any arbitrary length string
Number	Any numerical value
Checkbox	Boolean operator
Select	Single string value from a set of options
Multiselect	One or more string values from a set of options
Color	Color values passed as RGB color values
Datetime	Standardized date and time values
Path	Absolute or relative path to file/directory using Unix conventions
File	User uploaded binary data

Conditional Logic

- Complex tools with numerous input parameters and configuration options can easily translate into a long and incomprehensible UI form.
- How can add custom logic into a standardized manifest?
  - Boolean and logical operators
    - AND, OR, NOT
    - EQUALS, GREATER THAN, IF/THEN
  - Overloaded configurations

```yaml
ui:
- key: inputs.first_parameter
  type: checkbox
- key: inputs.second_parameter
  type: select
  fields: ['option1', 'option2', 'option3']
  condition: "inputs.first_parameter==true"
- key: inputs.second_parameter
  type: select
  fields: ['option1', 'option3']
  condition: "inputs.first_parameter==false"
```

Conditional Logic - Examples

- Use Case 1: Context-based options for a single field
  - Based on input selection of one or more other fields, certain options for a given field may no longer be valid. These fields or options within a field can be conditionally hidden (using the overload feature) to ensure that selected input parameters are valid.
- Use Case 2: Hidden fields
  - Given a tool with many input parameters and configuration options, it may be useful to conditionally hide/show various fields based on selection of other inputs.