

# Persistent Data and Serialization

## 1 Persistent Data

- string representations
- parsing and evaluating

## 2 Serialization

- saving and loading objects
- pickling objects

MCS 320 Lecture 9  
Introduction to Symbolic Computation  
Jan Verschelde, 25 June 2025

# Persistent Data and Serialization

## 1 Persistent Data

- string representations
- parsing and evaluating

## 2 Serialization

- saving and loading objects
- pickling objects

# Persistent Data and String Representations

A Jupyter notebook can store data. We may want to

- 1 process external data sets, and/or
- 2 store results of long computations.

Every object has a string representation.

- Writing an object to file takes two steps:
  - 1 Compute the string representation of an object.
  - 2 Write the string representation to file.
- Reading an object from file takes two steps:
  - 1 Read the string representation to file.
  - 2 Construct the object from the string representation.

# Persistent Data and Serialization

## 1 Persistent Data

- string representations
- parsing and evaluating

## 2 Serialization

- saving and loading objects
- pickling objects

# Parsing and Evaluating

On Python objects:

- `str()` is the method to convert an object to a string,
- `eval()` constructs the object given its string representation.

```
x = 3          # x is the object for the integer 3
sx = str(x)    # sx is the string representation of 3
y = eval(sx)   # y is the object for the integer 3
```

SageMath has a more elaborate number system than Python.

```
x = pi.n(digits=32) # x is a real_mpf.RealNumber
sx = str(x)         # the string representation of x
px = preparse(sx)   # prepare sx for the interpreter
```

`px` is the string

```
"RealNumber('3.1415926535897932384626433832795')"
```

With `eval(px)` we convert `px` to a `RealNumber`.

# Persistent Data and Serialization

## 1 Persistent Data

- string representations
- parsing and evaluating

## 2 Serialization

- saving and loading objects
- pickling objects

# Saving and Loading Objects

String representations are human readable,  
but are not the most efficient methods to store data.

SageMath objects have

- the `save()` method to write an object to a `.sobj` file, and
- the `load()` function to read a `.sobj` file into an object.

This simplifies the storing of data, but

- the `.sobj` files are not human readable, and
- require SageMath to read.

# Persistent Data and Serialization

## 1 Persistent Data

- string representations
- parsing and evaluating

## 2 Serialization

- saving and loading objects
- pickling objects



# Pickling Objects

Serialization is the conversion of an object into a byte stream.

The `pickle` module of Python offers *pickling* of objects.

- Saving a pickled object to file takes two steps:
  - 1 An object is converted into a `bytes` type.
  - 2 The string representation of those bytes is written to file.
- Reading a pickled object from file takes two steps:
  - 1 The bytes are read from file into a string.
  - 2 The string with the bytes is evaluated into an object.

Pickled objects preserve the type information better than plain strings.