

# Representation of functional Data

Before beginning to use the functionalities of the package, it is necessary to represent the data in functional form, using one of the following classes, which allow the visualization, evaluation and treatment of the data in a simple way, using the advantages of the object-oriented programming.

## Discrete representation

A functional datum may be treated using a non-parametric representation, storing the values of the functions in a finite grid of points. The `FDataGrid` class supports multivariate functions using this approach. In the [discretized function representation example](#) it is shown the creation and basic visualisation of a `FDataGrid`.

<code>skfda.representation.grid.FDataGrid</code> ( <code>data_matrix</code> )	Represent discretised functional data.
---	--

Functional data grids may be evaluated using interpolation, as it is shown in the [interpolation example](#). The following class allows interpolation with different splines.

<code>skfda.representation.interpolation.SplineInterpolator</code> ([...])	Spline interpolator of <code>FDataGrid</code> .
--	---

## Basis representation

The package supports a parametric representation using a linear combination of elements of a basis function system.

<code>skfda.representation.basis.FDataBasis</code> ( <code>basis, ...</code> )	Basis representation of functional data.
--	--

The following classes are used to define different basis systems.

<code>skfda.representation.basis.BSpline</code> ([...])	BSpline basis.
<code>skfda.representation.basis.Fourier</code> ([...])	Fourier basis.
<code>skfda.representation.basis.Monomial</code> ([...])	Monomial basis.

## Generic representation

Functional objects of the package are instances of `FData`, which contains the common attributes and methods used in all representations. This is an abstract class and cannot be instantiated directly, because it does not specify the representation of the data. Many of the package's functionalities receive an element of this class as an argument.

<code>skfda.representation.FData</code> (extrapolation, ...)
--

Defines the structure of a functional data object.
--

## Extrapolation

All representations of functional data allow evaluation outside of the original interval using extrapolation methods.

- [Extrapolation](#)
  - [Extrapolation Methods](#)
    - [skfda.representation.extrapolation.BoundaryExtrapolation](#)
    - [skfda.representation.extrapolation.ExceptionExtrapolation](#)
    - [skfda.representation.extrapolation.FillExtrapolation](#)
    - [skfda.representation.extrapolation.PeriodicExtrapolation](#)
  - [Custom Extrapolation](#)
    - [skfda.representation.evaluator.EvaluatorConstructor](#)
    - [skfda.representation.evaluator.Evaluator](#)