

---

# **symbol\_table\_bst**

***Release 1.0***

**Stefan Stefanache**

**Oct 17, 2022**

## INDICES AND TABLES

### **class** `symbol_table.SymbolTable`

A binary search tree implementation of the Symbol Table. This class implements a subset of the functionality provided by the built-in Python dictionary class.

#### **`__contains__`**(*key*)

Return true if key is in the symbol table, False otherwise

#### **`__getitem__`**(*key*)

Return the value associated with key. Raises a `KeyError` if key is not in the symbol table.

#### **`__init__`**()

Construct an empty symbol table.

#### **`__iter__`**()

Return an in-order iterator over keys.

#### **`__len__`**()

Return the number of values stored in the symbol table.

#### **`__setitem__`**(*key*, *value*)

Implements `self[key] = value`. If key is already stored in the symbol table then its value is modified. If key is not in the table, it is added.

#### **`__str__`**()

Return the symbol table as a string.

#### **`get`**(*key*)

Return the value associated with key. Returns `None` if key is not in the symbol table.

#### **`pop`**(*key*)

Remove and return the value with the indicated key. Raise a `KeyError` if the key is not in the symbol table.

## Symbols

`__contains__()` (*symbol\_table.SymbolTable method*), 1  
`__getitem__()` (*symbol\_table.SymbolTable method*), 1  
`__init__()` (*symbol\_table.SymbolTable method*), 1  
`__iter__()` (*symbol\_table.SymbolTable method*), 1  
`__len__()` (*symbol\_table.SymbolTable method*), 1  
`__setitem__()` (*symbol\_table.SymbolTable method*), 1  
`__str__()` (*symbol\_table.SymbolTable method*), 1

## G

`get()` (*symbol\_table.SymbolTable method*), 1

## M

module  
    *symbol\_table*, 1

## P

`pop()` (*symbol\_table.SymbolTable method*), 1

## S

*symbol\_table*  
    module, 1  
*SymbolTable* (*class in symbol\_table*), 1