jsMath

**Getting documentation and code**

**M=MatrixSpace(QQ,3) A=M([1,2,3,4,5,6,7,8,9])**

**A**

|  |  |
| --- | --- |
|  | **[1 2 3]****[4 5 6]****[7 8 9]** |

**A.determinant()**

**#This gives the value of A's determinant - remember determinant is a FUCNTION and so there are ( )s**

|  |  |
| --- | --- |
|  | **0** |
|  |  |

**A.determinant?**

 **# This ? without any parentheses gives the documentation on the determinant function**

|  |  |
| --- | --- |
|          | **File:** /sagenb/sage\_install/sage-4.7.2/devel/sage/sage/matrix/matrix\_rational\_dense.pyx**Type:** <type ‘builtin\_function\_or\_method’>**Definition:** A.determinant(algorithm=’default’, proof=None)**Docstring:**Return the determinant of this matrix.INPUT:* proof - bool or None; if None use proof.linear\_algebra(); only relevant for the padic algorithm.
* algorithm:

“default” – use PARI for up to 7 rows, then use integer“pari” – use PARI“integer” – clear denominators and call det on integer matrixNoteIt would be *VERY VERY* hard for det to fail even with proof=False.ALGORITHM: Clear denominators and call the integer determinant function.EXAMPLES:sage: m = matrix(QQ,3,[1,2/3,4/5, 2,2,2, 5,3,2/5])sage: m.determinant()-34/15sage: m.charpoly()x^3 - 17/5\*x^2 - 122/15\*x + 34/15 |

**A.determinant??**

**#This ?? without any parentheses gives the code for the determinant function**

|  |  |
| --- | --- |
|          | **File:** /sagenb/sage\_install/sage-4.7.2/devel/sage/sage/matrix/matrix\_rational\_dense.pyx**Source Code** (starting at line 766):def determinant(self, algorithm="default", proof=None): """ Return the determinant of this matrix. INPUT: - ``proof`` - bool or None; if None use proof.linear\_algebra(); only relevant for the padic algorithm. - ``algorithm``: "default" -- use PARI for up to 7 rows, then use integer "pari" -- use PARI "integer" -- clear denominators and call det on integer matrix .. note:: It would be \*VERY VERY\* hard for det to fail even with proof=False. ALGORITHM: Clear denominators and call the integer determinant function. EXAMPLES:: sage: m = matrix(QQ,3,[1,2/3,4/5, 2,2,2, 5,3,2/5]) sage: m.determinant() -34/15 sage: m.charpoly() x^3 - 17/5\*x^2 - 122/15\*x + 34/15 """ det = self.fetch('det') if not det is None: return det if self.\_nrows <= 2: # use generic special cased code. return matrix\_dense.Matrix\_dense.determinant(self) if algorithm == "default": if self.\_nrows <= 7: algorithm = "pari" else: algorithm = "integer" if algorithm == "pari": det = self.\_det\_pari() elif algorithm == "integer": A, denom = self.\_clear\_denom() det = Rational(A.determinant(proof=proof)) if denom != 1: det = det / (denom\*\*self.nrows()) else: raise ValueError("unknown algorithm '%s'"%algorithm) self.cache('det', det) return det |

|  |  |
| --- | --- |
|          |  |