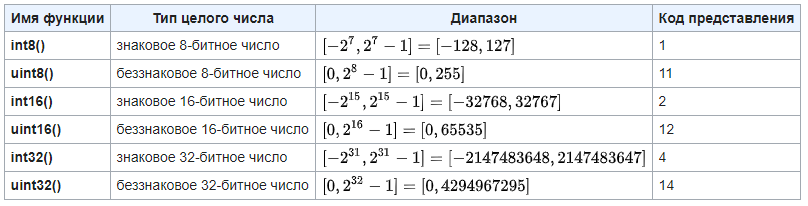
Справочник по математическим объектам и их представлению в Scilab

* **Числа**

**Целые числа**

Для хранения целого числа в памяти может быть использовано разное число битов, а именно 8, 16 и 32. От количества используемых битов зависит диапазон целых чисел. Кроме того, имеется возможность включения и отключения знакового бита, что бывает полезно, когда отрицательные целые числа не требуются.

В таблице ниже приведены функции для создания целых чисел и диапазоны возможных значений. Во всех случаях функция возвращает целое число, указанное в качестве аргумента, с определенным способом его хранения в памяти.



Для работы с целочисленными переменными существует две функции:

iconvert(X, itype) — меняет представление в памяти в общем случае матрицы из вещественных или целых чисел. Функции при этом следует передавать имя этой матрицы X и код внутреннего представления целого числа itype из последней колонки таблицы выше;

inttype(X) — возвращает код внутреннего представления в общем случае матрицы целочисленных данных.

#### **Комплексные числа**

Ввод комплексного числа в значительной мере можно упростить, если воспользоваться функцией **complex(a, b)**, в которой в качестве аргументов указать действительную и мнимую части соответственно. Согласно справочной информации, данная функция позволяет избежать проблем в объявлении комплексных переменных, в которых возможно появление констант %nan и %inf.

Для работы с комплексными числами также существует небольшой набор функций:

* conj(X) — возвращает комплексное число сопряженное числу X;
* real(X) — возвращает действительную часть комплексного числа X;
* imag(X) — возвращает мнимую часть комплексного числа X;
* isreal(X) — возвращает логическое %T, если передаваемый аргумент является комплексным числом, %F в противном случае.
* imult(X) — умножает мнимую единицу на аргумент. Согласно справочной информации, данную функцию рекомендуется использовать, когда приходится иметь дело с константами типа %nan и %inf, появляющихся в комплексном числе.
* **Матрица** — это двухмерный массив однотипных элементов. Можно понимать матрицу как несколько векторов-строк, записанных столбцом.

Создать матрицу в Scilab можно одним из нескольких способов:

* Матрицу можно создать из составляющих ее элементов;
* Из имеющихся векторов, упорядочив их строками или столбцами;
* Одной из специальных функций.

В общем случае синтаксическая конструкция имеет вид

[x11, x12, …, x1n; x21, x22, …, x2n; …; xm1, xm2, …, xmn]

Таким образом, вы создаете векторы-строки, которые отделяете точкой с запятой. Запятая в этом случае, как и с вектором, не обязательна.

-->A=[1 2; 3 4] *// создадим матрицу из составляющих ее элементов*

A =

1. 2.

3. 4.

#### **Математическая функция**

В Scilab есть предопределенные математические функции, такие как тригонометрические, экспонента и другие. Но что, если вам необходимо определить собственную функцию? Далее мы будем отличать математические функции и программируемые функции. Разница между ними заключается в том, что программируемые функции призваны реализовывать некоторый алгоритм, в то время как математическая функция отражает связь между множеством аргументов и множеством значений функции.

Отметим, что математическую функцию можно объявить через программирование, но так как обычно ее тело состоит из одной строчки, то рациональнее всего объявлять ее через специальную функцию **deff()**. Общий синтаксис имеет вид

deff('[Y1,Y2...]=Fname(X1,X2,...)',['Y1=выражение\_1';'Y2=выражение\_2;...'])