## Вариативная самостоятельная работа №3

## Задание 3.2.1

Переменная в Scilab (англ. Variable) — это именованный массив всего с одним полем, которое хранит данные некоторого типа.

Среди типов данных можно выделить:

* Целые числа
* Вещественные числа
* Комплексные числа
* Строки
* Логические переменные

Создать переменную в среде не составляет труда. Для этого, как вы уже знаете, достаточно ввести ее имя и присвоить ей какое-либо начальное значение. Для переменной будет автоматически выделено место в памяти, а ее область видимости по умолчанию становится локальной.

Для того, чтобы посмотреть текущее значение переменной, достаточно просто обратиться к ней по имени, либо воспользоваться редактором переменных.

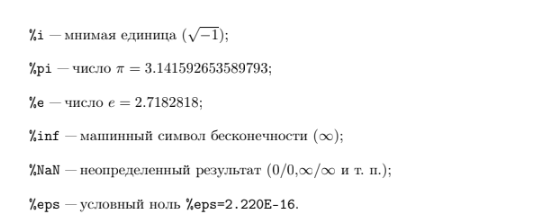
Запомните следующие правила, которым должны удовлетворять имена переменных и вообще любых объектов среды:

* Имя переменной может состоять из букв латинского алфавита (верхнего и нижнего регистра) и цифр;
* Имя переменной не может начинаться с цифры, но может начинаться с символов '%', '\_', '#', '!', '$', '?';
* Регистр в имени играет роль, то есть переменные с именами var,VAR ,Var и т. п. разные;
* Запрещено совпадение имени переменной с зарезервированными словами, такими как имена объявленных функций, констант и др.;

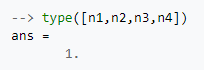
Создадим несколько переменных



В результате мы получили 4 переменных, 3 из которых хранят вещественные числа и одна — комплексное число. Знак процента указывает на то, что вы обращаетесь к *константе*.



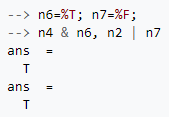
Несмотря на то, что в переменной n1 нет дробной части, с точки зрения хранения этого числа в памяти оно вещественно и имеет тип *double*. Убедимся, что переменные хранят числа, для чего вызовем внутреннюю функцию type(). Чтобы не вызывать функцию четыре раза, мы объединим наши переменные в вектор. Для объединения объектов в вектор необходимо перечислить их через запятую внутри квадратных скобок



В ответ функция возвращает число 1. Данным числом функция закодировала информацию о том, что в качестве аргумента была передана матрица, содержащая вещественные или комплексные числа удвоенной точности.  Не стоит заучивать наизусть что означает то или иное число, возвращаемое функцией type(), так как мы всегда можем воспользоваться справочной информацией.

Команда help() открывает окно справки и может перейти к разделу, описывающему функцию type().

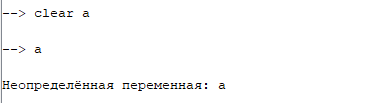
Логические переменные могут хранить в себе одну из двух предопределенных констант: %T (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *True* — *Истина*) и %F (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *False* — *Ложь*). Для логических типов данных применяются особые (логические) операции, тем не менее, этот тип данных совместим с числовым типом данных: любое ненулевое значение является эквивалентом %T, а соответственно нулевое значение — %F.



 Логическая операция успешно прошла даже с комплексной переменной.

Интерпретатор динамически реагирует на присваивание. Это означает, что любая объявленная переменная может хранить в любой момент времени любой тип данных.

Для очистки переменной используется команда clear.



Простейшие арифметические операции:

