

## **Руководство пользователя DialogOS**

(Диалоговая платформа для виртуальных ассистентов «NLab» (DialogOS))

Москва, 2023

## **АННОТАЦИЯ**

Программа DialogOS (далее Система) предназначена для обучения чат-ботов, применяемых для обработки текстовых и голосовых запросов пользователей, поступающих по каналам связи на естественном языке в режиме связанного диалога, для автоматической генерации ответов на них в контексте диалога.

Система предоставляет веб-интерфейс для работы с конфигурационными данными чат-ботов и реализует следующий функционал:

- 1) создание базы знаний чат-ботов;
- 2) создание и управление шаблонами чат-бота;
- 3) создание и управление словарями чат-бота;
- 4) компиляция шаблонов и словарей (обучение чат-бота);
- 5) просмотр логов компиляции.

В данном документе также представлено описание основных возможностей и синтаксиса языка DL, предназначенного для подготовки сценариев типовых диалогов, применяемых для обучения диалоговых систем на базе искусственного интеллекта (ботов).

## СОДЕРЖАНИЕ

АННОТАЦИЯ .....	2
СОДЕРЖАНИЕ .....	3
1. Введение .....	5
1.1. Лицензионное соглашение .....	5
1.2. Предмет договора .....	5
1.3. Имущественные права .....	5
1.4. Условия использования .....	5
1.5. Срок действия договора .....	6
1.6. Ограниченная гарантия .....	6
1.7. Требования к системе: .....	6
2. Установка .....	7
2.1. Установка и запуск Системы .....	7
2.2. Импортирование шаблонов для созданного профиля .....	7
2.3. Импорт словарей в созданный профиль .....	7
2.4. Вход в Систему .....	7
3. Работа с интерфейсом Системы .....	8
3.1. Элементы интерфейса Системы .....	8
3.1.1. Панель инструментов .....	8
3.1.2. Область навигации .....	8
3.1.3. Рабочая область .....	8
3.2. Создание/выбор профиля .....	8
3.3. Создание/выбор набора .....	8
3.4. Создание нового правила в IDE .....	9
3.5. Компиляция .....	9
4. Работа с Диалоговым процессором .....	10
4.1. Описание языка Диалогового процессора (DL) .....	10
1 Общие сведения .....	10
1.1 Элементы языка .....	10
1.1.1 Общий вид данных .....	10
1.1.2 Структура шаблона .....	10
1.1.3 Правила-условия .....	11
1.1.4 Правила-вопросы .....	11
1.1.5 Правила-ответы .....	13
1.2 Операторы .....	15

1.2.1	Ссылка на словарь [dict()]	15
1.2.2	Тег [if()]	15
1.2.3	Рекурсивные ссылки-вопросы [inf]	16
1.2.4	Гипертекстовая ссылка [href()]	16
1.2.5	Другие html-тэги	17
1.3	Переменные	17
1.3.1	Переменные Диалога	17
1.3.2	Переменные Пользователя	17
1.3.3	Переменные Бота	18
1.3.4	Использование переменных	18
1.3.5	Переменная %that_anchor	19
1.3.6	Супер якорь +%that_anchor="CIAS"	19
1.3.7	Переменные %last_mark и %last_mark2, %last_set1, %last_set2, %last_set3	19
1.4	Словари	20
1.4.1	Словари вопросов	20
1.4.2	Словари ответов	21
1.5	Функции в языке DL	22
1.5.1	Что такое функции в языке DL	22
1.5.2	Типы функций	22
1.5.3	Типы аргументов	23
	Перечень принятых сокращений	25

## **1. ВВЕДЕНИЕ**

### **1.1. Лицензионное соглашение**

#### **ВНИМАНИЕ!**

Данное программное обеспечение (ПО) (включая носители и печатные материалы) передается Вам на условиях Лицензионного договора. Перед установкой ПО внимательно ознакомьтесь с Договором передачи авторских прав на использование ПО (Лицензионный договор). Установка ПО рассматривается как Ваше полное согласие со всеми условиями Лицензионного договора.

**ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРЕДНАЗНАЧЕНО ДЛЯ  
УСТАНОВКИ ТОЛЬКО НА ОДНОМ КОМПЬЮТЕРЕ.**

### **ДОГОВОР О ПЕРЕДАЧЕ АВТОРСКИХ ПРАВ НА ИСПОЛЬЗОВАНИЕ ПРОГРАММ ДЛЯ ЭВМ И БАЗ ДАННЫХ (ЛИЦЕНЗИОННЫЙ ДОГОВОР)**

Настоящий Лицензионный договор является юридическим соглашением между Вами, конечным пользователем (физическим или юридическим лицом), и Открытым Акционерным обществом «Лаборатория Наносемантика» (ООО «Лаборатория Наносемантика»).

Данное Программное обеспечение (далее по тексту ПО) – это комплекс программ для ЭВМ, баз данных и документации, который является объектом авторского права и охраняется законом.

### **1.2. Предмет договора**

Предметом настоящего Лицензионного договора является передача ООО «Лаборатория Наносемантика», как Правообладателя, Вам, как Конечному Пользователю, неисключительного авторского права на использование ПО (далее по тексту – Лицензия).

Все условия, оговоренные далее, относятся как к ПО в целом, так и ко всем его компонентам в отдельности.

### **1.3. Имущественные права**

Имущественные права на данное ПО принадлежат исключительно ООО «Лаборатория Наносемантика»

Лицензия дает Вам, Конечному Пользователю, право на использование ПО в любых целях при соблюдении приведенных ниже условий.

### **1.4. Условия использования**

Вы можете использовать ПО одновременно только на одном компьютере.

Для использования ПО в многопользовательской системе Вы должны иметь лицензию на каждое рабочее место.

Вы не имеете права осуществлять самостоятельно или допускать других осуществлять следующую деятельность:

1. Пытаться деассемблировать, декомпилировать (преобразовывать объектный код в исходный текст) программы, базы данных и другие компоненты ПО;
2. Вносить какие-либо изменения в объектный код программ или баз данных к ним за исключением тех, которые вносятся средствами, включенными в комплект ПО и описанными в документации;
3. Предоставлять права на использование программ или другие права на ПО третьим лицам;
4. Совершать относительно ПО другие действия, нарушающие Российские и международные нормы по авторскому праву и использованию программных средств.

### 1.5. Срок действия договора

Настоящий Лицензионный договор вступает в силу с момента установки ПО на Ваш компьютер и действует на протяжении всего периода его использования.

В случае нарушения любого из условий настоящего Лицензионного договора Вы лишаетесь предоставленного настоящим Лицензионным договором права на использование ПО. ООО «Лаборатория Наносемантика» вправе защищать свои интересы в соответствии с законодательством Российской Федерации в области авторских и смежных прав.

### 1.6. Ограниченная гарантия

ПО поставляется «таким, каково оно есть». ООО «Лаборатория Наносемантика» не гарантирует, что ПО не содержит ошибок, а также не несет никакой ответственности за прямые или косвенные последствия применения ПО, в том числе возникшие из-за возможных ошибок или опечаток в комплекте ПО.

#### ООО «Лаборатория Наносемантика»

Почтовый адрес:	121357, г. Москва, ул. Верейская, д. 29, стр. 134, этаж 7, помещение Н2
Телефон, факс	+7 (495) 995-58-72
E-mail:	<a href="mailto:hello@nanosemantics.ai">hello@nanosemantics.ai</a>
WWW:	<a href="http://www.nanosemantics.ai">www.nanosemantics.ai</a> <a href="https://nanosemantics.ai/dialog-os">https://nanosemantics.ai/dialog-os</a>

### 1.7. Требования к системе:

Для выполнения Системы требуется:

1. доступ к интернет;
2. веб браузер;
3. персональный компьютер под управлением Linux или MacOS с установленным приложением Docker Desktop последней версии (не ниже 2.1).

## 2. УСТАНОВКА

### 2.1. Установка и запуск Системы

Для запуска NLab IDE в стандартном режиме, нужно выполнить следующие действия:

Установка Докер:

```
docker-compose build
```

Запуск контейнеров:

```
docker-compose up -d
```

Затем надо дождаться старта всех сервисов и создания базы данных.

Создание базового чат-бота для веб-виджета:

```
./scripts/command.sh bot:create --profile test
```

Вместо названия «test», уточните профиль бота, создаваемый в IDE.

### 2.2. Импортирование шаблонов для созданного профиля

Для импортирования шаблонов из файла в созданный профиль воспользуйтесь следующей командой, где «test» - имя созданного в IDE профиля, и «file» - путь к шаблону:

```
./scripts/import_templates.sh test file
```

### 2.3. Импорт словарей в созданный профиль

Для импорта словарей из файла в созданный профиль воспользуйтесь следующей командой, где «test» - имя созданного в IDE профиля, и «dicts» - путь к директории словарей:

```
./scripts/import_dicts.sh test dicts
```

Для запуска установленной Системы в адресной строке браузера<sup>1</sup> укажите адрес сайта программы <http://localhost:3000> и нажмите «Enter», интерфейс виджета будет доступен по адресу <http://localhost>. При успешном запуске пользователю будет показан экран авторизации Системы.

### 2.4. Вход в Систему

Поставляемая версия Системы предполагает одного пользователя.

Для входа в Систему введите логин и пароль<sup>2</sup> в форме авторизации по адресу: <http://localhost> и нажмите кнопку «Войти».

Если логин и пароль были введены верно, откроется главная страница Системы.

<sup>1</sup> Для запуска Системы воспользуйтесь актуальной версией браузера Google Chrome или Chromium.

<sup>2</sup> Логин и пароль по умолчанию: admin, admin.

### **3. РАБОТА С ИНТЕРФЕЙСОМ СИСТЕМЫ**

#### **3.1. Элементы интерфейса Системы**

Интерфейс Системы состоит из трех основных областей.

- 1) панель инструментов;
- 2) область навигации;
- 3) рабочая область.

##### **3.1.1. Панель инструментов**

Панель инструментов содержит:

- 1) строку «Меню» - позволяет выполнить переход в основные разделы Системы;
- 2) кнопку «Запустить компиляцию» - позволяет запустить перекомпиляцию;
- 3) форма для работы с набором – позволяет выбрать или создать набор.

##### **3.1.2. Область навигации**

Область навигации содержит:

- 1) основное меню Системы - позволяет выполнить переход в основные разделы Системы;
- 2) область списков – позволяет просматривать и изменять списки созданных наборов/шаблонов/словарей.

##### **3.1.3. Рабочая область**

Рабочая область позволяет просматривать и изменять содержимое наборов/шаблонов/словарей.

#### **3.2. Создание/выбор профиля**

Перед началом работы выберите профиль, с которым собираетесь работать или создайте новый. Профиль – это именование набора чат-ботов, доступного данному. Выбор производится в выпадающем меню в верхнем левом углу в панели инструментов.

- 1) для создания нового профиля нажмите пиктограмму «Плюс», введите название профиля;
- 2) для сохранения ввода нажмите на пиктограмму «Дискета»;
- 3) для отмены ввода нажмите на пиктограмму «Перечеркнутый круг».

#### **3.3. Создание/выбор набора**

После создание/выбора набора на главной странице в левой панели откроется список, содержащий наборы шаблонов и словарей, которые содержат правила, образующие базу знаний. Деление на наборы не несет никакой технической нагрузки, но позволяет логически сгруппировать правила, что облегчает работу с базой знаний.



Для создания набора нажмите пиктограмму «Плюс» в нижнем левом углу экрана под списком наборов, введите название набора после чего:

- 1) для сохранения ввода нажмите на пиктограмму «Дискета»;
- 2) для отмены ввода нажмите на пиктограмму «Перечеркнутый круг»;
- 3) для удаления одного или нескольких наборов, выберите их в списке наборов и нажмите пиктограмму «Крест» в нижнем левом углу экрана под списком наборов;
- 4) для включения/выключения одного или нескольких наборов, выберите их в списке наборов и нажмите пиктограмму «Красная точка» в нижнем левом углу экрана под списком наборов.

### 3.4. Создание нового правила в IDE

Наполнение базы знаний чат-бота происходит в режиме редактирования.

Редактирование правил подразумевает их создание, правку, сохранение и удаление. Чтобы перейти к правилам, создайте/выберите один набор в левой панели.

Для создания нового правила нажмите пиктограмму «Плюс» в правом верхнем углу, придумайте и введите название нового правила, нажмите на пиктограмму «Дискета» для его сохранения, после чего вы увидите форму для ввода контента создаваемого правила. Напишите новое правило:

- 1) для сохранения ввода нажмите на пиктограмму «Дискета»;
- 2) для выхода из режима без сохранения ввода нажмите на пиктограмму «Перечеркнутый круг»;
- 3) для создания копии правила нажмите на пиктограмму «Копия»;
- 4) для удаления правила нажмите на пиктограмму «Корзина».

Важно!

Не забывайте сохранить новое правило, иначе вы не сможете его скомпилировать и ввести в состав базы знаний чат-бота. Точно также не забывайте сохранять все изменения, которые вы делаете в правилах.

### 3.5. Компиляция

Для записи выполненных изменений в базу знаний чат-бота запустите компиляцию нажатием пиктограммы «Компиляция».

Для просмотра журнала компиляции перейдите в раздел «Компиляция». Результаты компиляции выделены в журнале цветом:

- 1) зеленым с меткой «FINISHED» – успешно завершённые процессы компиляции;
- 2) красным с меткой «FINISHED» – процессы, не завершённые из-за ошибок;
- 3) желтым с меткой «WORKING» – процесс компиляции, который находится в обработке.

## 4. РАБОТА С ДИАЛОГОВЫМ ПРОЦЕССОРОМ

### 4.1. Описание языка Диалогового процессора (DL)

#### 1 Общие сведения

Язык DL предназначен для подготовки сценариев типовых диалогов, применяемых для обучения диалоговых систем на базе искусственного интеллекта (ботов), имеет облегченный синтаксис, что позволяет при формировании правил минимально затенять размечаемый текст на естественном языке.

Язык DL является встроенным языком – важнейшей частью платформы, отвечающей за создание и сопровождение ботов.

#### 1.1 Элементы языка

##### 1.1.1 Общий вид данных

Язык DL состоит из правил. Каждое правило описывает реакцию Бота на определенную реплику (или класс реплик) пользователя.

Наиболее крупные группы правил, объединенные одной темой, следует выделять в отдельные файлы (так называемые наборы) и присваивать им значимые имена.

##### 1.1.2 Структура шаблона

Каждый шаблон состоит из следующих частей, расположенных в описанном порядке:

###### 1.1.2.1 Группа «Правила-условия»<sup>3</sup>.

**Правило-условие** - описание состояния переменных на языке DL.

При выполнении шаблона-условия данный шаблон будет рассматриваться Ботом при выборе ответа на реплику пользователя. Иначе, при всех других состояниях переменных шаблон будет исключен из рассмотрения.

*Формат:*

Правило-условие начинается знаком плюс + (без пробела после него) и оканчивается концом абзаца. Правила-условия идут один за другим.

---

<sup>3</sup> Не обязана присутствовать.

### 1.1.2.2 Группа «Правила-вопросы».

**Правило-вопрос** - описание на языке DL реплики или класса реплик пользователя, реакцию на которые описывает шаблон.

*Формат:*

Правило-вопрос начинается знаком *доллара* \$ (с пробелом после него или без) и оканчивается концом абзаца. Правила-вопросы идут один за другим.

### 1.1.2.3 Группа «Правила-Ответы».

**Правило-ответ** - описание на языке DL возможных реакций Бота на реплику/класс реплик пользователя, описанных в шаблонах-вопросах данного шаблона.

Если шаблон включает в себя несколько правил-ответов, то при выборе этого шаблона как наиболее релевантного, случайным образом выбирается один шаблон-ответ.

*Формат:*

Правило-ответ начинается знаком решётки # (с пробелом после него или без) и оканчивается концом абзаца. Правила-ответы идут один за другим.

В шаблонах-вопросах нет разницы между прописными и строчными буквами. В шаблонах-ответах буквы выводятся для пользователя так, как они даны в шаблоне.

## 1.1.3 Правила-условия

Условие на переменную:

### Пример

```
+%Topic= "Bot"
$ * ты кто *
# Я Бот, который отвечает на ваши вопросы.
```

Этот шаблон будет рассмотрен, только если текущее значение переменной Topic = Bot.

## 1.1.4 Правила-вопросы

Символы, которые можно использовать:

- русские и английские буквы,
- пробелы, запятые, восклицательные и вопросительные знаки
- цифры,
- специальные знаки –звездочка, тильда,
- переменные и теги.

### 1.1.4.1 Варианты задания шаблона-вопроса

В шаблон-вопросе необходимо описывать возможные варианты реплики пользователя. Варианты можно задать несколькими способами:

#### а) Синонимичные шаблоны-вопросы

Варианты шаблона-вопроса можно записать несколькими подряд идущими строками, начинающимися со знака доллара \$.

#### Пример

```
$ Как тебя зовут?
  $ Как звать-то *?
  $ Как твоё имя *?
```

#### б) Словари

Варианты шаблона-вопроса или части шаблона-вопроса можно перечислить в отдельном словаре, а в шаблоне поставить ссылку на имя словаря в нужном месте шаблона. Подробнее см. ниже раздел «Словари».

#### в) Inline-словари

Варианты шаблона-вопроса можно записывать непосредственно в вопросе.

Синтаксис: {string/string /--/...}

-- обозначает пустой вариант.

#### Пример

```
$ {Скучно/Грустно/Одиноко} мне.
# Как после этого жить...
```

### 1.1.4.2 Кванторы в шаблонах-вопросах

Разрешены следующие виды кванторов:

#### а) Звёздочка \*

Звёздочка \* используется для обозначения любого слова или последовательности слов. Звёздочка может равняться нулю, то есть означать отсутствие слова.

#### Пример

Запись "\*" адрес" распознаётся как "какой ваш адрес", "дайте адрес" и т.д., либо просто "адрес".

**б) Суперзвёздочка \*\***

Суперзвёздочка \*\* означает наличие любого слова или последовательности слов, и не может равняться нулю. Суперзвёздочка при разборе шаблона имеет приоритет над звёздочкой. Конструкция \*\* понимается как одна суперзвёздочка.

**в) Тильда ~**

Тильда ~ служит для обозначения произвольного начала или окончания слова, пишется вплотную к слову. Бот трактует слово с тильдой как некоторый набор возможных слов.

**Пример**

запись «кнопк~» распознаётся как «кнопка», «кнопки», «кнопке» и т. д.

запись «к~» распознаётся как любое слово, начинающееся на букву к.

**1.1.5 Правила-ответы**

Символы, которые можно использовать:

- русские и английские буквы,
- пробелы, знаки препинания,
- цифры,
- кванторы «звездочка»,
- переменные и теги.

В группе «шаблоны-ответы» может быть несколько ответов, ответ пользователю выбирается по случайному счётчику.

**1.1.5.1 Варианты задания шаблона-ответа****а) Синонимичные шаблоны-ответы**

Варианты шаблона-ответа можно записать несколькими подряд идущими строками, начинающимися со знака решетки #.

**Пример**

\$ Спасибо

# Да не за что...

# Обращайтесь, если что.

# Всегда пожалуйста.

## б) Словари

Варианты шаблона-ответа или части шаблона-ответа можно перечислить в отдельном словаре, а в шаблоне поставить ссылку на имя словаря в нужном месте шаблона. Подробнее см. ниже раздел «Словари».

## в) Inline-словари

Варианты шаблона-ответа можно записывать непосредственно в ответе.

Синтаксис: {string / string / -- /...}

-- обозначает пустой вариант.

Область использования: шаблоны-ответы.

### Пример

§ \*

# Как {дальше/дальше с этим/после этого} жить...

## 1.1.5.2 Кванторы в шаблонах-ответах

### а) Вставка значения звёздочки в ответ

Звёздочка \* означает любой набор слов в некотором месте шаблона-вопроса. В реальном диалоге пользователь на месте, где стоит звёздочка, вводит конкретные слова. Эти слова называются **реализацией звёздочки**. Реализацию звёздочки можно использовать для формирования ответа.

Для вставки в ответ реализации звёздочки используются следующие конструкции:

[\*] - вставка реализации звёздочки (первой слева звёздочки, если их в вопросе много),

[\*1] – вставка первой слева реализации звёздочки,

[\*2] – вставка второй слева реализации звёздочки,

[\*3] – вставка третьей слева реализации звёздочки.

### Пример

§ \* получить техподдержку \*

# по вопросу [\*2] обращайтесь в нашу службу поддержки support@company.ru

Также существует конструкция, которая позволяет выделить конкретную часть запроса. В шаблоне-вопросе ту часть запроса, которую мы хотим использовать в ответе инфа, обрамляем квадратными скобками и тире:

[-{a/b/c}-]

[-a-]

[-\*-]

[-[dict(name)]-]

а в шаблоне-ответе ссылаемся на эту часть запроса следующим образом:

[&X]

где X – цифры. Номер той части запроса гостя, к которой идет отсылка в шаблоне-ответе, считается слева направо по открывающей квадратной скобке в шаблоне-вопросе.

[&1] – вставка первой слева реализации звёздочки,

[&2] – вставка второй слева реализации звёздочки,

[&3] – вставка третьей слева реализации звёздочки.

### **Пример**

```
$ * получить техподдержку по вопросу [-*-]
```

```
# по вопросу [&1] обращайтесь в нашу службу поддержки support@company.ru
```

## **1.2 Операторы**

### **1.2.1 Ссылка на словарь [dict()]**

**Синтаксис:** [dict(dict\_name)], где dict\_name – имя существующего словаря.

**Область использования:** шаблоны-вопросы, шаблоны-ответы.

**Описание:**

Для компактного вида и удобства использования варианты шаблона-вопроса и шаблона-ответа можно записывать в отдельные словари, а в шаблонах ставить ссылку на эти словари (подробнее про словари см. ниже).

### **Пример**

```
$ * [dict(dialog-privet)] *
```

```
# Привет! Как жизнь, как настроение?
```

В словаре dialog-privet.txt содержатся все варианты приветствий пользователя.

### **1.2.2 Тег [if()]**

**Синтаксис:**

[if(%var)]{string} - проверка на существование значения у переменной

[if(!%var)]{string} - проверка на несуществование значения у переменной

[if(%var="value")] - проверка на совпадение с некоторым значением

[if(%var!="value")]{string} - проверка на несовпадение с некоторым значением

Использование альтернативы [elsif]{string} и [else]{string} является факультативным.

**Область использования:** шаблоны-ответы.

**Описание:**

Если условие, наложенное на переменную, удовлетворяется, то вместо оператора IF вставляется текст, записанный в фигурных скобках.

Вместо значения переменной "value" можно вставлять референции [&1].

**Пример**

```
$ Как меня зовут
# [if(%user_name="Антон")] { Вас зовут Антон, как и меня! } [elseif(%user_name)] { Вас зовут [%user_name]! } [else] { Я не знаю, как вас зовут. }
```

### 1.2.3 Рекурсивные ссылки-вопросы [inf]

**Синтаксис:** [inf]{string}

**Область использования:** шаблоны-ответы.

**Описание:**

Для пользователя текст, записанный в фигурных скобках, будет отображаться в виде гипертекстовой ссылки. При нажатии на ссылку текст соответствующей ссылки будет задан в качестве очередного вопроса к Боту.

**Пример**

```
$ * поисковая система *
# Это довольно общий вопрос, пожалуйста, выберите более узкую тему:
[inf]{Что такое поисковая система} [br]
[inf]{Список ведущих поисковых систем Рунета} [br]
[inf]{Расширенный поиск в поисковых системах} [br]
[inf]{Поиск картинок, новостей, определений, дискуссий} [br]
[inf]{Кто работал и работает в поисковых системах (персоналии)} [br]
[inf]{Новости о поисковых системах}
```

### 1.2.4 Гипертекстовая ссылка [href()]

**Синтаксис:**

[href(url)(\_blank)]{string} – соответствует стандартной html-ссылке <a href=url target=\_blank">string</a>. url следует писать стандартным способом.

**Область использования:** шаблоны-ответы.

**Описание:**



На экране пользователя на месте ссылки отображается текст, записанный в фигурных скобках. При нажатии на данный текст браузер пользователя переходит на страницу, заданную url.

**Варианты синтаксиса:**

[href(url)]{string} – открытие новой страницы в том же окне.

[href(url)(\_blank)]{string} – открытие новой страницы в новом окне.

### 1.2.5 Другие html-тэги

Тексты ответов могут содержать символ перевода строки [br].

## 1.3 Переменные

Переменные используются для сохранения знаний об Боте или его собеседнике-пользователе в течение сеанса или на более долгое время. Значения переменных устанавливаются в ходе диалога и могут быть извлечены и использованы при необходимости.

*Формат:*

Устанавливать или извлекать значение переменных можно в тексте ответа.

Имя переменной может состоять только из латинских букв, цифр, знака дефиса «-» и подчёркивания «\_». Регистр букв в названии переменной имеет значение.

**Все переменные делятся на три следующих класса.**

### 1.3.1 Переменные Диалога

**Переменные Диалога** – это переменные, которые отображают текущее состояние диалога и помогают Боту в нем ориентироваться. В процессе диалога Бот может изменять и использовать их значения.

### 1.3.2 Переменные Пользователя

**Переменные Пользователя** хранят информацию о текущем собеседнике (пользователе) бота. Рекомендуется именовать переменные пользователя с использованием префикса user\_. Например:

- user\_name – имя пользователя
- user\_lastname – фамилия пользователя
- user\_age – возраст пользователя
- user\_gender – пол пользователя
- user\_city – город пользователя
- user\_phone – телефон пользователя
- user\_email – email пользователя

- и другие.

По желанию разработчиков могут быть введены любые другие переменные

### 1.3.3 Переменные Бота

**Переменные Бота** отображают информацию об Боте для использования в ответах на личные вопросы о персонаже бота:

- inf\_name – имя бота
- inf\_gender – пол бота

и другие.

### 1.3.4 Использование переменных

#### а) Присвоение значения

Синтаксис: [%var = "значение"]

Область использования: шаблоны-ответы.

#### Пример

```
$ Меня зовут *
# Приятно познакомиться! [%user_name="&1"]
```

#### б) Подстановка значения переменной

Синтаксис: [%var]

Область использования: шаблоны-ответы, операторы, функции (об операторах и функциях см. ниже).

#### Пример

```
$ пока, робот!
# [if(%user_name)]{ До свидания, [%user_name]! }{else}{ До свидания! }
```

Прежде чем выводить переменную в ответе, для начала может понадобиться убедиться, что она не пуста.

#### в) Удаление значения переменной

Синтаксис: [%var = ""]

#### Пример

```
$ забудь мое имя
# Хорошо, я больше не помню, как вас зовут. [%user_name=""]
```

### 1.3.5 Переменная `%that_anchor`

Синтаксис: `[%that_anchor = "значение"]`

Область использования: шаблоны-ответы.

Если на следующем после присваивания шаге переменной не присваивается новое значение, то переменная обнуляется.

#### Пример

```
$ *
# Ты {шутить/меня разыгрываешь/прикалываешься}. [%that_anchor = "шутить"]
```

#### **Проверка значения `that_anchor`**

Синтаксис:

`+%that_anchor="значение"` - проверка определенного значения

`+%that_anchor!="значение"` - проверка не соответствия определенному значению

`+%that_anchor` – проверка того, что переменная не пустая

`+%that_anchor` – проверка того, что переменная пустая

Область использования: шаблоны-условия.

### 1.3.6 Супер якорь `+%that_anchor="CIAS"`

Синтаксис: `+%that_anchor = "CIAS"`

или `+%that_anchor = "COMPARE_IS_ALWAYS_SUCCESSFULL"`

Так называемый супер якорь используется для повышения приоритета шаблона. Это нужно, например, для служебных и тестовых правил или правил с реакцией на ненормативную лексику.

CIAS не перебивает другие `that_anchor`, которые тоже подходят.

### 1.3.7 Переменные `%last_mark` и `%last_mark2`, `%last_set1`, `%last_set2`, `%last_set3`

Существуют переменные:

- `%last_mark`
- `%last_mark2`
- `%last_set1`
- `%last_set2`
- `%last_set3`

Если сработал ответ, в котором не присваивается значение такой переменной, её текущее значение сбрасывается.

Этим такие переменные похожи на `that_anchor`, но, в отличие от `that_anchor`, у шаблона, у которого в плюс-строке стоит условие на некоторое значение такой переменной, нет приоритета по

сравнению с остальными шаблонами, т.е. если есть подходящий более длинный шаблон, то сработает он, а не шаблон с условием на такую переменную.

### Пример

#### **Правила:**

\$ \* погода \*

# Не знаю ничего про погоду. [%last\_set1="погода"]

+%last\_set1="погода"

\$ \*

# Но вообще погода сегодня ничего так...

\$ \*

# Ничего не понимаю.

\$ \* музыка \*

# И про музыку ничего не знаю.

#### **Чаты:**

Гость: какая сегодня погода?

Бот: Не знаю ничего про погоду.

Гость: ылдовалыворп

Бот: Но вообще погода сегодня ничего так...

Гость: ыллллыв

Бот: Ничего не понимаю.

Гость: все-таки скажи, какая погода

Бот: Не знаю ничего про погоду.

Гость: ну ладно, про погоду не знаешь, а музыка интересует?

Бот: И про музыку ничего не знаю.

## **1.4 Словари**

### **1.4.1 Словари вопросов**

Варианты шаблона-вопроса можно записывать в отдельных словарях, а в шаблонах-вопросах ставить ссылку на эти словари.

В словаре каждый вариант шаблона-вопроса должен занимать одну строку.

#### **Замечания:**

Словари ответов и словари вопросов отличаются только использованием, то есть теоретически один и тот же словарь может использоваться как словарь правил-вопросов, так и как словарь правил-ответов.

### Пример

```
$ * [dict(vopros-anekdot)] *
# [dict(horosho)] [br] [dict(chapaev)]
# [dict(horosho)] [br] [dict(vovochka)]
```

Здесь использован словарь vopros-anekdot.txt, представляющий собой варианты просьбы рассказать анекдот:

```
расска~ анекдот~
дай~ анекдот~
знае~ анекдот~
анекдот~
```

В ответе использованы словари ответов – способ выразить согласие (horosho.txt) и различные наборы анекдотов по темам.

### 1.4.2 Словари ответов

Варианты шаблона-ответа можно записывать в отдельных словарях, а в шаблонах-ответах ставить ссылку на эти словари. Вместо ссылки на словарь в шаблон-ответ будет случайным образом подставляться один из вариантов ответа из словаря.

### Пример

```
$ * поисковая система *
# [dict(quest_search)] [br]
[inf]{Что такое поисковая система} [br]
[inf]{Список ведущих поисковых систем Рунета} [br]
[inf]{Расширенный поиск в поисковых системах}
```

Здесь словарь имеет имя quest\_search. В словаре quest\_search записаны следующие варианты шаблона-ответа:

```
Хотите узнать больше о поисковых системах? Выберите тему.
Поисковые системы - широкая область знаний. Вот какие есть разделы:
Хороший вопрос! Уточните, пожалуйста, что Вы имеете в виду:
Позвольте, я уточню? Выберите, пожалуйста, что-нибудь из предложенного.
Широкая тема, могу предложить ознакомиться с тематическими разделами.
```

## 1.5 Функции в языке DL

### 1.5.1 Что такое функции в языке DL

Функции — это теги вида `[@FunctionName()]`, которые могут использоваться в шаблон-ответах и условиях оператора `If`. Каждый такой тег заменяется на некоторый текст. Что именно это будет за текст зависит от самой функции.

#### Пример

Так, например, функция `Date` заменяется на текущую дату и если в шаблон-ответе написать:

```
Сегодня [@Date()]
```

то в ответе Бота будет выведен текст:

```
Сегодня 01.04.2020
```

Некоторые функции, в отличие от функции `Date`, могут принимать аргументы. Одной из таких является функция `ProperName`, которая всегда должна использоваться ровно с одним аргументом. Эта функция заменяется на переданный ей аргумент, в котором первый символ заменяется на заглавный, а все остальные - на строчные.

#### Пример

```
# Меня зовут [@ProperName("[%inf_name")]
```

и если в переменной `%inf_name` записано `глеб` или `гЛеБ`, то Бот скажет:

```
Меня зовут Глеб
```

### 1.5.2 Типы функций

Выше были приведены примеры функций `Date`, `ProperName`, являющихся **текстовыми**, то есть заменяемых на некоторый текст. Есть ещё один вид функций - **условные**. Эти функции используются только в условиях оператора `If` и могут принимать только *истинное* или *ложное* значение. Например, **условная** функция `IsEqual` принимает два аргумента, значения которых нужно проверить на равенство:

#### Пример

```
# [if([@IsEqual("[%inf_age","&1"])]){Ты угадал, мне [&1] лет} [else] {А вот и нет, мне не [&1] лет :-P}
```

**Условная** функция не может быть использована нигде кроме условия оператора `If`. Использование в условии оператора `If` **текстовых** функций не допускается. Зато текстовые функции могут использоваться в фигурных скобках, следующих за оператором `If`:

```
# [if([@IsEqual("[%inf_name]", "&1"])]{Да, меня зовут [@ProperName( "[%1]" )], по крайней мере последние [%inf_age] лет}
```

### 1.5.3 Типы аргументов

Для каждой функции строго определено, какие значения она может получать в качестве аргументов. Для этого существует пять типов аргументов:

- **Текстовый** - значение интерпретируется как текст.
- **Динамический** - значение может содержать текст, значения звездочек и референций, значения переменных.
- **Расширенный** - этот тип похож на динамических, но в нем можно использовать также и функции.
- **Произвольный** - значение может содержать любые элементы языка DL.
- **Переменная** - значение должно быть именем переменной, значение которой может использовать и/или изменять функция.

Аргументы **текстового** типа интерпретируются как простой текст и, даже если они содержат такие элементы языка DL как тэги, звёздочки или референции, эти элементы не обрабатываются.

В аргументах **динамического** типа подставляются значения звездочек, референций и переменных. Примеры функций с аргументами такого типа уже были приведены выше. Стоит запомнить, что для использования **переменных** в аргументах **динамического** типа их следует заключать в квадратные скобки:

#### Пример

```
# [@ToUpper("[%inf_name] и весь этот текст будут напечатаны заглавными буквами")]
```

функция *ToUpper* возвращает вместо себя свой аргумент, все буквы в котором заменены на заглавные. В результате получится:

ГЛЕБ И ВЕСЬ ЭТОТ ТЕКСТ БУДУТ НАПЕЧАТАНЫ ЗАГЛАВНЫМИ БУКВАМИ

Как видно, аргумент функции может состоять из нескольких различных элементов, в данном случае — это переменная и текст.

Значения аргументов **расширенного** типа отличаются от значений аргументов **динамического** типа только тем, что могут содержать в себе функции:

Значения аргументов **произвольного** типа, как и ожидается от его названия, могут быть любыми элементами языка DL. Не нужно забывать заключать **переменные** в прямоугольные скобки при использовании в аргументах этого типа.

Отдельное место занимают аргументы типа **переменная**. Такой аргумент, во-первых, может состоять только из имени переменной, причём только одного, а не из произвольного числа

различных элементов языка DL, и, во-вторых, имя переменной не должно заключаться в квадратные скобки. Кроме того функции, принимающие аргументы типа **переменная**, могут изменять значения переданных им переменных. Так, например, функция *SetValue* подставляет вместо себя переданное ей значение и, в то же время, присваивает его указанной переменной.

### Пример

```
# Очень приятно, [@SetValue("user_name", "[*1]" )], будем отныне знакомы!
```

Здесь, если пользователь в своей предыдущей фразе представился Глебом, мы получим следующий ответ Бота:

```
Очень приятно, Глеб, будем отныне знакомы!
```

При этом переменной *%user\_name* будет присвоено значение *Глеб*. Обратите внимание, что имя переменной здесь **не заключалось в прямоугольные скобки**.



### Перечень принятых сокращений

- язык DL - компилируемый язык программирования, DL используется в качестве аббревиатуры в текстовых сообщениях для представления Язык данных.
- диалоговые системы - система, работающая в режиме диалога, при котором она отвечает на каждую команду пользователя и по мере надобности обращается к нему за информацией
- искусственный интеллект (бот) - Виртуальный собеседник Бот
- Кванторы - (от лат. quantum сколько) в логике и математике логические эквиваленты слов все, каждый и т. п. (кванторы общности), некоторый, существует (кванторы существования) и др.;
- Синтаксис - набор правил построения фраз алгоритмического языка, позволяющий определить, осмысленные предложения в этом языке.
- Гипертекстовая ссылка - Гиперссылка (англ. hyperlink) часть гипертекстового документа, ссылающаяся на другой элемент (команда, текст, заголовок, примечание, изображение) в самом документе
- Комбинаторика - раздел математики, посвящённый решению задач выбора и расположения элементов из некоторого основного (обычно конечного) множества в соответствии с заданными правилами.
- функция - в математике - одно из основных понятий, выражение, определяющее регулярную зависимость между двумя множествами переменных величин, заключающуюся в том, что каждому элементу одного множества соответствует определенная, единственная величина из другого.
- значение аргумент- значение (число, указатель и т. д.), передаваемое функции, а также символьное имя (название переменной) в тексте программы, выступающее в качестве идентификатора этого значения.