

Динамические формы: Исчерпывающее руководство  
для новичков

Кирилл Кривошеев

18 марта 2019 г.



# Оглавление

<b>1</b>	<b>Введение</b>	<b>1</b>
1.1	Что такое программа . . . . .	1
1.2	Запуск динамических команд . . . . .	1
1.3	Первая программа . . . . .	1
1.4	Чувствительность к регистру . . . . .	2
1.5	Арифметические функции . . . . .	2
1.6	Глоссарий . . . . .	2
1.7	Упражнения . . . . .	2
<b>2</b>	<b>Структура программы</b>	<b>3</b>
2.1	Значения и типы . . . . .	3
2.2	Идентификаторы . . . . .	3
2.3	Возвращаемые значения . . . . .	3
2.4	Порядок выполнения . . . . .	3
2.5	Функция call . . . . .	3
2.6	Операторы ветвления . . . . .	3
2.7	Циклы . . . . .	3
2.8	Упражнения . . . . .	4
<b>3</b>	<b>Динамическая форма</b>	<b>5</b>
3.1	Что такое динамическая форма? . . . . .	5
3.2	Текстовые поля . . . . .	5
3.3	Списки . . . . .	5
3.4	Чекбоксы . . . . .	5
<b>4</b>	<b>Группы</b>	<b>7</b>
4.1	Виды групп . . . . .	7
4.2	Копирование групп . . . . .	7
4.3	Удаление групп . . . . .	7
<b>5</b>	<b>Асинхронные команды</b>	<b>9</b>
5.1	Автоисполнитель . . . . .	9
5.2	Сопрограммы . . . . .	9



# Глава 1

## Введение

### 1.1 Что такое программа

### 1.2 Запуск динамических команд

Одним из первых вызовов, который бросает динамический язык начинающему разработчику, является необходимость настройки рабочего окружения HP Service Manager для редактирования шаблонов динамических форм. Однако, благодаря мощным возможностям динамического языка, сделать первые шаги в разработке можно гораздо проще: достаточно открыть шаблон обращения «Исполнитель» в dev-среде портала «Лицо ДРУГа». Это простая динамическая форма, состоящая всего из двух полей: «Команда» и «Результат».

### 1.3 Первая программа

По давней традиции, первая программа, которую начинающий разработчик пишет на новом языке, является «Hello, world!». На динамическом языке она выглядит так:

```
>>> setValue([Hello, world!],[result])
```

Для выполнения программы введите ее в поле «Команда» и переставьте курсор в поле «Результат». В итоге, в поле «Результат» вы увидите текст:

```
Hello, World!
```

Поздравляем! Вы сделали первый шаг на безумно интересном пути к освоению могучих возможностей создани интерактивных форм, который дает вам Динамический язык. Это пример команды **setValue**, которая устанавливает значение, указанное первым аргументом в квадратных скобках в поле, указанное во втором аргументе.

Здесь и далее, все команды, которые вводятся в поле «Команда» будут предваряться символами

```
>>>
```

За которыми следует результат выполнения в поле «Результат»

## 1.4 Чувствительность к регистру

Динамический язык нечувствителен к регистру. Это означает, что программы:

```
>>> setValue([Hello, world!],[result])
```

и

```
>>> setValue([Hello, world!],[result])
```

и даже

```
>>> sEtVaLuE([Hello, world!],[result])
```

С точки зрения интерпретатора динамического языка являются идентичными и произведут одинаковый результат. И несмотря на то, что последний пример выглядит, безусловно, наиболее круто, в примерах к книге мы будем придерживаться так называемого CamelCase, так как он наиболее удобочитаем:

```
>>> setValue([Hello, world!],[result])
```

## 1.5 Арифметические функции

Перейдем от «Hello, world!» к арифметике. Как и любой, уважающий себя язык программирования, динамический язык предоставляет возможность работы с основными арифметическими операциями, а именно **plus**, **minus** и **mul** для сложения, вычитания и умножения, а также **div** для деления:

```
>>> setValue([plus([68],[1])],[result])
69
>>> setValue([minus([69],[1])],[result])
68
>>> setValue([mul([6],[7])],[result])
42
>>> setValue([div([84],[2])],[result])
42
```

На самом деле, указанные функции обладают гораздо большим набором возможностей, но для первого знакомства этого будет достаточно.

## 1.6 Глоссарий

## 1.7 Упражнения

Откройте шаблон «Исполнитель» и решите следующие задачи:

1. Сколько будет в градусах Цельсия температура в 69 Фаренгейт?
2. Сколько спринтов потребуется для вывода в промышленную эксплуатацию шаблона «Совместная поездка» если каждый спринт исправляется 6 багов, и обнаруживается 4 новых?

## Глава 2

# Структура программы

### 2.1 Значения и типы

### 2.2 Идентификаторы

### 2.3 Возвращаемые значения

### 2.4 Порядок выполнения

### 2.5 Функция call

### 2.6 Операторы ветвления

### 2.7 Циклы

Вооружившись ранее изученным набором команд, вам не составит труда реализовать цикл на динамическом языке.

Напишем программу, которая выведет в поле «Результат» 100 точек:

Добавьте на динамическую форму скрытое поле с 'id' : 'counter'. Это поле будет служить счетчиком цикла.

Ниже приведен полный листинг программы:

```
if(
  [isnotin([getValue([counter]]),[100])],
  [run(
    [setValue(
      [plus(
        [getValue([result])],
        [.]
      )],
      [result]
    )],
    [setValue(
      [plus(
        [getValue([counter])],
```

```

        [1]
      ]],
      [counter]
    ]],
    [call(
      [getValue([cmd])],
      [this]
    )]
  )]
)]

```

Запустив программу, можно убедиться, что в поле «Результат» вывелось ровно 100 точек.

Теперь рассмотрим принцип работы программы подробнее:

```

if(
  [isnotin([getValue([counter])], [100])],

```

В данном фрагменте кода мы проверяем, не выполняется ли условие неравенства нашего счетчика значению 100.

В этом случае выполняем тело цикла, состоящее из 3-х команд внутри блока «run»:

```

setValue([plus([getValue([result])], [.] ]), [result])

```

Здесь происходит добавление новой точки в поле с результатом.

```

setValue([plus([getValue([counter])], [1] )], [counter])

```

Увеличиваем значение счетчика на 1.

```

call([getValue([cmd])], [this])

```

И, главная часть, обеспечивающая заикливание нашей программы - вызываем повторное выполнение скрипта при помощи функции «Call»

## 2.8 Упражнения

Откройте шаблон «Исполнитель» и решите следующие задачи:

1. Напишите программу, которая выводит на экран числа от 1 до 100. При этом вместо чисел, кратных трем, программа должна выводить слово «Fizz», а вместо чисел, кратных пяти — слово «Buzz». Если число кратно и 3, и 5, то программа должна выводить слово «FizzBuzz»



## Глава 3

# Динамическая форма

3.1 Что такое динамическая форма?

3.2 Текстовые поля

3.3 Списки

3.4 Чекбоксы



## Глава 4

# Группы

4.1 Виды групп

4.2 Копирование групп

4.3 Удаление групп



## Глава 5

# Асинхронные команды

### 5.1 Автоисполнитель

### 5.2 Сопрограммы