

GAP tutorial

Alexander Konovalov

Centre for Interdisciplinary Research in Computational Algebra
University of St Andrews, Scotland

Fourth RISC/SCIENCE Training School in Symbolic Computation, Linz, June 29–July 10, 2009

*Supported by the EU FP6 project "**SCIENCE** – Symbolic Computation Infrastructure for **Europe**"*



Introduction

- What is GAP?
- What is it not?
- Things available in GAP
- Demo 1

Symbols and Characters

- Input Characters
- Symbols

Keywords, Identifiers and Expressions

- Keywords
- Identifiers
- Expressions

Statements

- Types of Statements
- IF Statement
- Loops
- Functions and Procedures

Summary

- Challenge
- Further

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

Acknowledgements

Thanks to:

- ▶ Temur Kutsia
- ▶ Steve Linton
- ▶ John McDermott
- ▶ Max Neunhöffer

Course outline

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

- ▶ Overview of the GAP system
- ▶ Basics of working with GAP
- ▶ Programming in GAP
- ▶ **SCSCP**: Symbolic Computation Software Composability Protocol:
 - ▶ for combining GAP instances and other CASs
 - ▶ for parallel computations in GAP

What is GAP?

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

- ▶ “Groups, Algorithms and Programming”
- ▶ A CAS directed at computational group and representation theory.
- ▶ Consists of a small(ish) kernel written in C (~ 170000 lines of code) ...
- ▶ ... an interpreted computer language called GAP ...
- ▶ ... and a library of algorithms written in GAP (~ 400000 lines of code) ...
- ▶ ... available via several thousands of user's functions
- ▶ The language has rather unique object oriented features invented to model mathematical objects:
 - ▶ objects learn during their lifetime and
 - ▶ change their type in the process.
- ▶ Method selection based on current type of all arguments.

What is GAP?

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

- ▶ mid-80s: development began in Aachen: J. Neubüser, M. Schönert, others
- ▶ since 1997 (J. Neubüser retired): international project coordinated from St Andrews
- ▶ since 2005: four equal Centres: St Andrews, Aachen, Braunschweig, Fort Collins
- ▶ GAP is a **platform to implement mathematical** (mostly discrete) **algorithms** (→ currently ~ 90 packages)
- ▶ GAP has a sizable **user and developer community**
- ▶ GAP contains large collections of **mathematical knowledge** in algorithms and databases
- ▶ GAP is **open source** (**GPL**): free, open, extendable
- ▶ available from `http://www.gap-system.org`

What is GAP not?

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

- ▶ A general purpose computer algebra system (like Maple, Mathematica or MuPAD)
- ▶ A numerical algorithms toolbox/system (like Matlab)
- ▶ A one-click solution for arbitrary problems involving groups

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and
Characters

Input Characters

Symbols

Keywords,
Identifiers and
Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

Algorithms and data available in GAP

- ▶ permutation groups,
- ▶ other representations for groups (matrix, polycyclic, finitely presented),
- ▶ semigroups/monoids
- ▶ linear algebra (in particular over finite fields),
- ▶ combinatorics (graphs → [GRAPE](#), codes → [GUAVA](#), designs → [DESIGN](#), cohomology → [cohomolo](#), [HAP](#))
- ▶ interfaces to other systems (→ SCIENCE project),
- ▶ [Small Groups Library](#), including, among others, all [423 164 062](#) groups of order not greater than 2000, except [49 487 365 422](#) groups of order 1024
- ▶ crystallographic groups (→ [CRYST](#) and [CRYSTCAT](#))
- ▶ character tables (→ [ctblib](#) package)
- ▶ access to the WWW Atlas of group representations (→ [AtlasRep](#) package).

Demo 1

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

- ▶ Now it's time to see **GAP** in action !!!
- ▶ <http://www.cs.st-andrews.ac.uk/~alexk/gap/risc/tutorial1.pdf>
- ▶ <http://www.cs.st-andrews.ac.uk/~alexk/gap/risc/quickref.pdf>
- ▶ <http://www.cs.st-andrews.ac.uk/~alexk/gap/risc/demo1.g>
- ▶ <http://www.cs.st-andrews.ac.uk/~alexk/gap/risc/demo2.g>

Allowed characters in GAP

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

- ▶ Digits
- ▶ Uppercase and lowercase letters
- ▶ Space, tab, newline, return
- ▶ Certain list of special characters

GAP special characters

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

+ - * ^ / \ < = >

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

() [] { } ~ # &

• , : ; ? ! " \ _

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

GAP symbols

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

+	-	*	/	^	~	! .
=	<>	<	<=	>	>=	! [
:=	.	..	->	,	;	! {
[]	{	}	()	:

Keywords for operations and data structures

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

▶ and

▶ or

▶ not

▶ in

▶ mod

▶ rec

Keywords for conditional statements

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

- ▶ `if`
- ▶ `then`
- ▶ `elif`
- ▶ `else`
- ▶ `fi`

Keywords for loops

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

- ▶ `for`
- ▶ `while`
- ▶ `do`
- ▶ `od`
- ▶ `repeat`
- ▶ `until`
- ▶ `break`
- ▶ `continue`

Keywords for functions

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

- ▶ `function`
- ▶ `local`
- ▶ `return`
- ▶ `end`

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

Keywords for leaving break loops and GAP

Introduction

- What is GAP?
- What is it not?
- Things available in GAP
- Demo 1

Symbols and Characters

- Input Characters
- Symbols

Keywords, Identifiers and Expressions

- Keywords**
- Identifiers
- Expressions

Statements

- Types of Statements
- IF Statement
- Loops
- Functions and Procedures

Summary

- Challenge
- Further

- ▶ `quit`
- ▶ `QUIT`

All GAP keywords in alphabetical order

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

and	break	continue	do	elif
else	end	fi	for	function
if	in	local	mod	not
od	or	repeat	return	then
until	while	quit	QUIT	rec

Identifiers implemented as keywords

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

- ▶ true
- ▶ false
- ▶ fail
- ▶ IsBound
- ▶ Unbind
- ▶ TryNextMethod
- ▶ Info
- ▶ Assert
- ▶ SaveWorkspace

Valid identifiers

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

- ▶ An identifier is used to refer to a variable
- ▶ An identifier consists of letters, digits, and underscores
- ▶ It must contain at least one letter or underscore sign

Examples of valid identifiers

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

a foo aLongIdentifier

hello Hello HELLO

x100 100x _100

some_people_prefer_underscores_to_separate_words

WePreferMixedCaseToSeparateWords

Expressions

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

- ▶ Simple examples of expressions:
 - ▶ variable
 - ▶ function call
 - ▶ integer
 - ▶ permutation
 - ▶ string
 - ▶ function
 - ▶ list
 - ▶ record
- ▶ You can construct more complex expressions using **operators**

Operators

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

► Comparisons:

= <> < <= > >= in

► Arithmetic operators:

+ - * / mod ^

► Logical Operators:

not and or

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and
Characters

Input Characters
Symbols

Keywords,
Identifiers and
Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

Types of statements

A statement is a construction of one of the following types:

- ▶ Assignment
- ▶ Procedure call
- ▶ `if ... then ... elif ... else ...
fi`
- ▶ Loops:
 - ▶ `for ... do ... od`
 - ▶ `while ... do ... od`
 - ▶ `repeat ... until`
- ▶ Return

IF: simple case

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

```
if bool-expr then  
    statements1  
fi;
```


IF: two-branch case

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

```
if bool-expr then  
    statements1  
else  
    statements2  
fi;
```

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

IF: more complicated case

```
if bool-expr1 then
    statements1
elif bool-expr2 then
    statements2
...
elif bool-exprK then
    statementsK
else
    statement
fi;
```

FOR loop over list

Introduction

- What is GAP?
- What is it not?
- Things available in GAP
- Demo 1

Symbols and Characters

- Input Characters
- Symbols

Keywords, Identifiers and Expressions

- Keywords
- Identifiers
- Expressions

Statements

- Types of Statements
- IF Statement
- Loops**
- Functions and Procedures

Summary

- Challenge
- Further

```
for variable in list-expr do  
    statements  
od;
```

FOR loop over object

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

```
for variable in object do  
    statements  
od;
```

FOR loop over iterator

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

```
for variable in iterator do  
    statements  
od;
```

WHILE loop

Introduction

- What is GAP?
- What is it not?
- Things available in GAP
- Demo 1

Symbols and Characters

- Input Characters
- Symbols

Keywords, Identifiers and Expressions

- Keywords
- Identifiers
- Expressions

Statements

- Types of Statements
- IF Statement
- Loops**
- Functions and Procedures

Summary

- Challenge
- Further

```
while bool-expr do  
    statements  
od;
```

UNTIL loop

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

```
repeat  
    statements  
until bool-expr;
```

Other statements for loops

Introduction

- What is GAP?
- What is it not?
- Things available in GAP
- Demo 1

Symbols and Characters

- Input Characters
- Symbols

Keywords, Identifiers and Expressions

- Keywords
- Identifiers
- Expressions

Statements

- Types of Statements
- IF Statement
- Loops**
- Functions and Procedures

Summary

- Challenge
- Further

```
break;
```

```
continue;
```


Defining a function

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

```
function( comma-separated arguments names )  
  local comma-separated local variables names;  
  statements  
end;
```

Note: `local` can be omitted if there isn't any local variable.

RETURN statement

Introduction

- What is GAP?
- What is it not?
- Things available in GAP
- Demo 1

Symbols and Characters

- Input Characters
- Symbols

Keywords, Identifiers and Expressions

- Keywords
- Identifiers
- Expressions

Statements

- Types of Statements
- IF Statement
- Loops
- Functions and Procedures**

Summary

- Challenge
- Further

```
return;
```

```
return expression;
```

Function call

Introduction

What is GAP?
What is it not?
Things available in GAP
Demo 1

Symbols and Characters

Input Characters
Symbols

Keywords, Identifiers and Expressions

Keywords
Identifiers
Expressions

Statements

Types of Statements
IF Statement
Loops
Functions and Procedures

Summary

Challenge
Further

```
FunctionName ( ) ;
```

```
FunctionName ( comma-separated arguments names  
)
```

```
FunctionName ( comma-separated arguments names  
: comma-separated options )
```

Demo 2

Introduction

- What is GAP?
- What is it not?
- Things available in GAP
- Demo 1

Symbols and Characters

- Input Characters
- Symbols

Keywords, Identifiers and Expressions

- Keywords
- Identifiers
- Expressions

Statements

- Types of Statements
- IF Statement
- Loops
- Functions and Procedures

Summary

- Challenge**
- Further

► Let's see some GAP programming examples

Summary

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and Characters

Input Characters

Symbols

Keywords, Identifiers and Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

- ▶ With the knowledge of GAP programming language you can:
 - ▶ automate routine computations
 - ▶ implement new algorithms
 - ▶ work with GAP in a more efficient way
- ▶ Knowledge of GAP programming language allows you in principle **to predict the result of each and every input**
- ▶ GAP programming expertise levels:
 - ▶ input from command line
 - ▶ writing functions; organising code into files
 - ▶ extending existing operations with new methods
 - ▶ adding new operations for existing kinds of objects
 - ▶ extending the system with new kinds of objects

Introduction

What is GAP?

What is it not?

Things available in GAP

Demo 1

Symbols and
Characters

Input Characters

Symbols

Keywords,
Identifiers and
Expressions

Keywords

Identifiers

Expressions

Statements

Types of Statements

IF Statement

Loops

Functions and Procedures

Summary

Challenge

Further

Further information

The screenshot shows a web browser window with the title "GAP System for Computational Discrete Algebra". The address bar shows the URL "http://www.gap-system.org/". The browser's navigation bar includes buttons for back, forward, home, and search, along with a Google search bar. Below the navigation bar, there is a menu with links: Often, News, GAP, GAP5, Wiki, OpenMath, MathRev, arXiv, NGS, ACM, VUB, Dell, SMI, Golf, Tech, TV, and Maps. The main content area is divided into two columns. The left column has a blue header with the text "GAP" and a "Main Branches" section. Below this, there is a "Sitemap" section with links to "Download", "Overview", "Data Libraries", "Packages", "Documentation", "Contacts", "FAQ", and "GAP3". The right column has a "Main Branches" section with links to "Download", "Overview", "Data Libraries", "Packages", "Documentation", "Contacts", "FAQ", and "GAP3". Below this, there is a "Welcome to" section with the title "GAP - Groups, Algorithms, Programming - a System for Computational Discrete Algebra". This is followed by a "What is GAP?" section, which contains a paragraph describing the system. The paragraph states: "GAP is a system for computational discrete algebra, with particular emphasis on [Computational Group Theory](#). GAP provides a [programming language](#), a library of thousands of functions implementing algebraic algorithms written in the GAP language as well as large [data libraries](#) of algebraic objects. See also the [overview](#) and the description of the [mathematical capabilities](#). GAP is used in research and teaching for studying groups and their representations, rings, vector spaces, algebras, combinatorial structures, and more. The system, including source, is distributed [freely](#). You can study and easily modify or extend it for your special use." Below this paragraph, there is a "Quicklinks" section with links to "Site Structure", "Search Web Site", "Capabilities", "Manuals", and "Examples". The bottom of the browser window shows a "Done" button.

GAP

Main Branches

[Download](#) [Overview](#) [Data Libraries](#) [Packages](#) [Documentation](#) [Contacts](#) [FAQ](#) [GAP3](#)

Sitemap

Navigation Tree

[Start](#)

[Download](#)

[Overview](#)

[Data Libraries](#)

[Packages](#)

[Documentation](#)

[Contacts](#)

[FAQ](#)

[GAP3](#)

Quicklinks

[Site Structure](#)

[Search Web Site](#)

[Capabilities](#)

[Manuals](#)

[Examples](#)

Welcome to

GAP - Groups, Algorithms, Programming - a System for Computational Discrete Algebra

What is GAP?

GAP is a system for computational discrete algebra, with particular emphasis on [Computational Group Theory](#). GAP provides a [programming language](#), a library of thousands of functions implementing algebraic algorithms written in the GAP language as well as large [data libraries](#) of algebraic objects. See also the [overview](#) and the description of the [mathematical capabilities](#). GAP is used in research and teaching for studying groups and their representations, rings, vector spaces, algebras, combinatorial structures, and more. The system, including source, is distributed [freely](#). You can study and easily modify or extend it for your special use.

The current release is GAP 4.4.12. The pages of this web site describe this release if not stated otherwise. The webpage [updates](#) explains the history of changes. The older version [GAP3](#) is still available.