

# The Big Book of Science

ocket88888

Big Book of Science Series



Series information: ./LSP/lsp-seriesinfo/eotms-info.tex not found!

# The Big Book of Science

ocket88888



# Contents

<b>1</b>	<b>Building Blocks: Basic Data Types</b>	<b>1</b>
1.1	Functions . . . . .	1
<b>2</b>	<b>Change title for chapter 2 in chapters/02.tex</b>	<b>3</b>
<b>3</b>	<b>Change title for chapter 3 in chapters/03.tex</b>	<b>5</b>
	<b>List of references</b>	<b>7</b>
	<b>Index</b>	<b>7</b>
	Name index . . . . .	7
	Language index . . . . .	7
	Subject index . . . . .	7



# 1 Building Blocks: Basic Data Types

Add content in chapters/01.tex.

## 1.1 Functions

Functions are incredibly useful constructs throughout mathematics, the natural sciences, and in particular computer science. Functions take any number of inputs, or arguments, usually performing operations on those arguments to produce an output. The definition of a function takes the following form:

```
1 type FunctionName [ type ArgumentName0 , type ArgumentName1 , ...  
    type ArgumentNameN ] {  
2     Operations  
3     Possibly Output  
4 }
```

Listing 1.1: Function Example

As seen above, the first word is "type," which refers to the type of data that the result, or output, of the function will be. In later chapters we will define more of these datatypes, but for now we'll introduce the most simple. "void" is the type of data that doesn't exist. That is to say, a function of type void not produce a result, and is typically used only to modify the arguments. It should be noted that an argument cannot be of this type, as it's kind of nonsensical to pass no information as an argument.

The second word is "FunctionName," which is, perhaps obviously, a placeholder for the name of the function. A reader with any amount of experience with trigonometry will recognize the function name Sine, for example. When using a function, only the name of and arguments to the function need be provided for its use to be complete and valid.

After the function name, a list of comma-separated arguments, with their own types, are specified. A function may take any number of arguments, but the most common is simply one. In cases of multiple arguments the order of the arguments

## *1 Building Blocks: Basic Data Types*

is important. A datum of type A may not be passed to a function that only accepts one argument of type B. Similarly, a function that takes one argument of type A, and then an argument of type B cannot be used on one argument of type B, and then an argument of type A.

The placeholders "Operations" and "Possibly Output" are contained within curly braces ({ and }). This serves to make entirely explicit the operations that the function performs, so that they may not be confused with any work done directly beneath them.

Functions are used in a similar manner to this throughout many computer science languages. Specifically, those familiar with Java or C may find 1.1 to be reminiscent of those languages. Please note that the above is NOT representative of any programming language, it is merely a way of organizing definitions of mathematical operations.



## **2 Change title for chapter 2 in chapters/02.tex**

Add content in chapters/02.tex



# 3 Change title for chapter 3 in chapters/03.tex

Add content in chapters/03.tex<sup>1</sup>

---

<sup>1</sup> Comrie (1981) is still useful for diversity linguistics.



# Bibliography

Comrie, Bernard. 1981. *Language universals and linguistic typology*. Oxford: Basil Blackwell.

# Back Title

ISBN 978-0-000000-00-2



9 780000 000002