



The Pragmatic Language

Primer

This document is designed to help programmers learn the basics of Bill.

Copyright (c) 2022 J. Smith. Permission is granted to copy, distribute and/or modify this document, provided this notice is kept.

	1
H ello World	3
sectionB eginning3	
T ypes	5
chapterE xpressions7	
S trings	9
chapterC onditionals11	
C ondtainers	13
chapterL oops15	
F unctions	D eclaring
E xceptions	S tyle

Chapter 1

Bill Primer

1.1 Beginning

Hello

World

"Hello World" is traditionally the first program one writes in a new language.

```
#!/usr/bin/env bill

# hello.bill
# aka hello world

fun main():no_value
{
    println("Hello World")
    exit                // defaults to 0
}
```

hello.bill

Of course, the first few lines are unnecessary. However, declaring a main function is required. Details, on that, in the next chapter ([Functions](#)).

Next: [Types](#)

Chapter 2

Conditionals

Conditionals

If...

Chapter 3

Containers

Containers

Containers can hold multiple values.

Chapter 4

Exceptions

Exceptions

oops!

Chapter 5

Expressions

[Expressions](#)

Express!

Chapter 6

Functions

Functions

Encapsulating repeatable steps, is how we make programming easier.

```
#!/usr/bin/env bill

# hello.bill
# aka hello world

fun main():no_value
{
    println("Hello World")
    exit // defaults to 0
}
```

hello.bill

The function definition line should look familiar. The "fun" reserved word declares a function.

After the ":" is the function return type. However, in this case "no_value" indicates there is no return of any kind.

A common practice of statically typed languages is to declare the type "void," which is similar.

6.1 Declaring a Function

etc...

Chapter 7

Strings

Strings

\$\$\$

Chapter 8

Style

Style

STYLE!

Chapter 9

Types

Types

Every form of data is a type. By extension, the same is true for functions.

