text++

Programmable, Typesetting PDF Generation for the C Programmer.

Joi Anderson - jna2123 // Manager + Tester
Klarizsa Padilla - ksp2127 // Language Guru
Maria Javier - mj2729 // System Architect + Tester

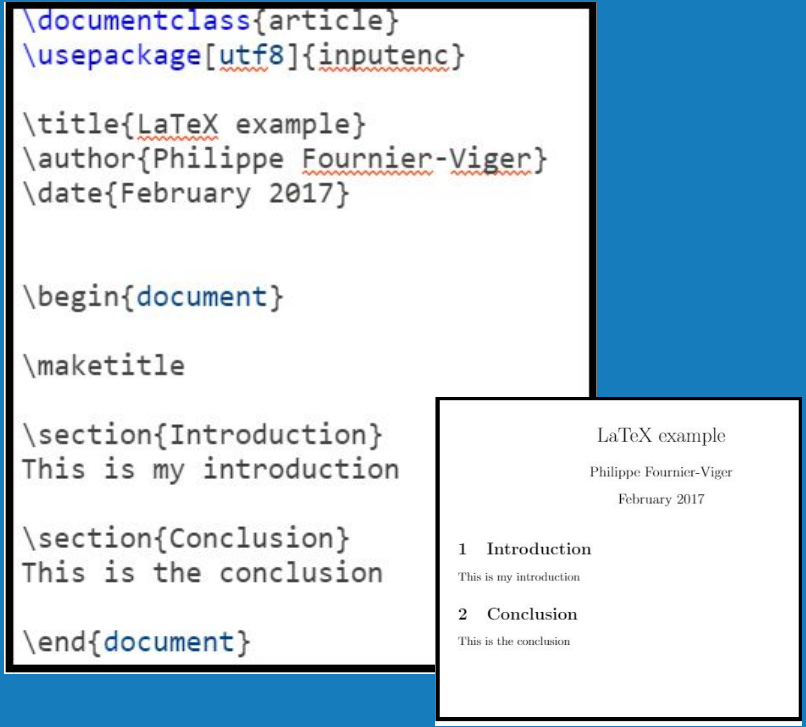
text++

- Allows programmers to create documents as efficiently as they write code.
- Intuitive programming syntax, especially for C programmers.
- Designed for the production of technical documentation.
- Algorithmic computing capabilities

Motivation

{ TEX }

"TeX is intended for the creation of **beautiful books** — and especially for books that contain a lot of mathematics."



```
\documentclass{article}
\usepackage[utf8]{inputenc}

\title{LaTeX example}
\author{Philippe Fournier-Viger}
\date{February 2017}


\begin{document}

\maketitle

\section{Introduction}
This is my introduction

\section{Conclusion}
This is the conclusion

\end{document}
```

LaTeX example

Philippe Fournier-Viger

February 2017

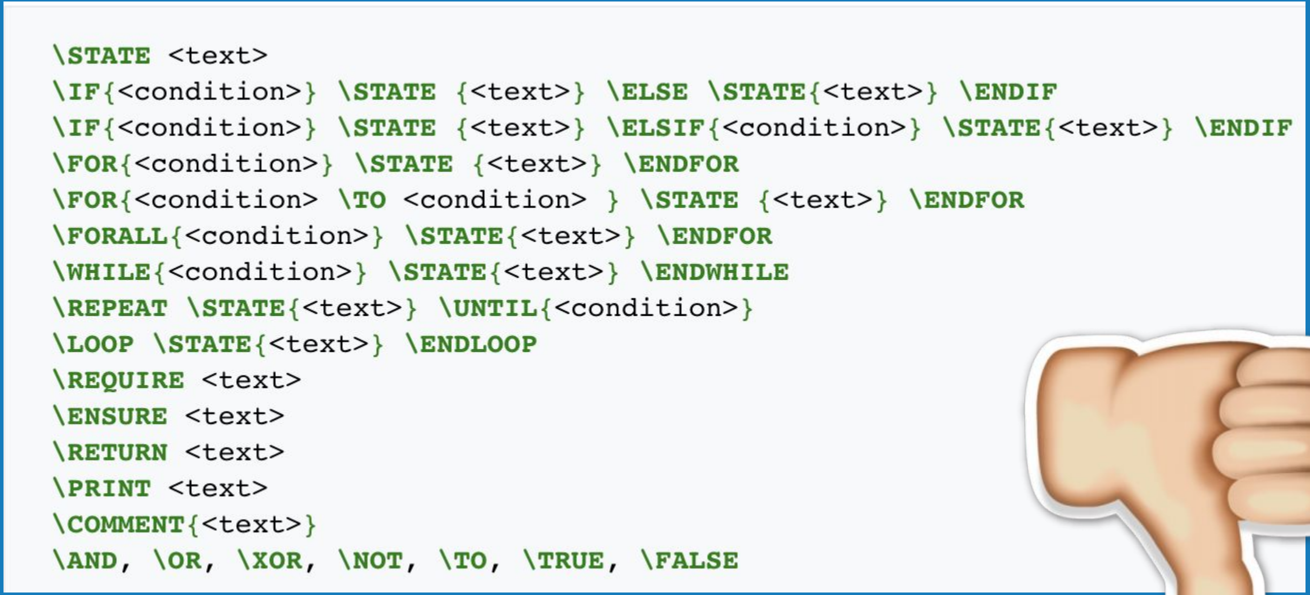
1 Introduction

This is my introduction

2 Conclusion

This is the conclusion

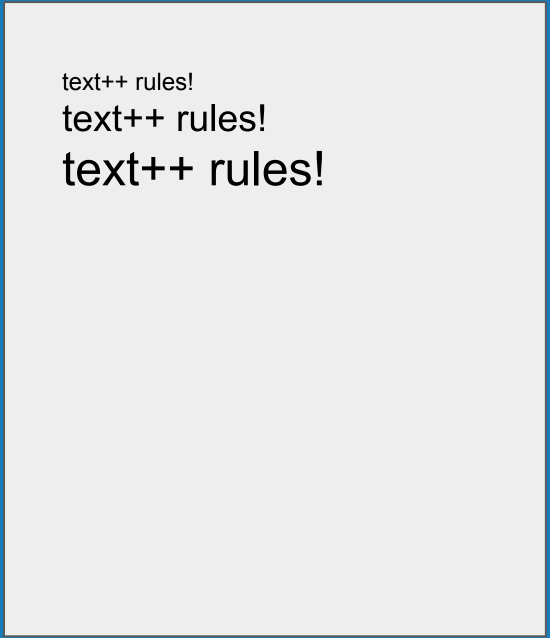
Motivation



```
\STATE <text>
\IF{<condition>} \STATE {<text>} \ELSE \STATE{<text>} \ENDIF
\IF{<condition>} \STATE {<text>} \ELSIF{<condition>} \STATE{<text>} \ENDIF
\FOR{<condition>} \STATE {<text>} \ENDFOR
\FOR{<condition> \TO <condition> } \STATE {<text>} \ENDFOR
\FORALL{<condition>} \STATE{<text>} \ENDFOR
\WHILE{<condition>} \STATE{<text>} \ENDWHILE
\REPEAT \STATE{<text>} \UNTIL{<condition>}
\LOOP \STATE{<text>} \ENDLOOP
\REQUIRE <text>
\ENSURE <text>
\RETURN <text>
\PRINT <text>
\COMMENT{<text>}
\AND, \OR, \XOR, \NOT, \TO, \TRUE, \FALSE
```

Solution: text++

```
def void start(){

   int i;
   int size;
   string text;

   size = 12;
   text = “text++ rules!”;

   for (i =0; i < 3; i = i + 1){

     changeFontSize(“Helvetica”, size);
     write( text );
     size = size + 20;

   }
}
```



text++ rules!

text++ rules!

text++ rules!

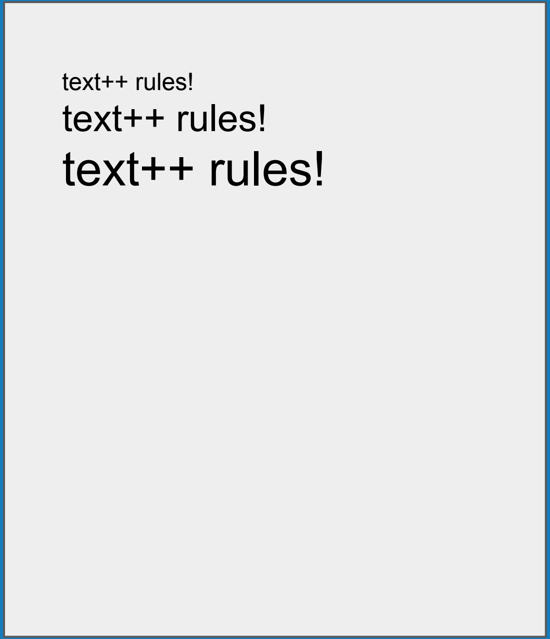
text.pdf

Solution: text++

```
def void start(){

    int i;
    int size;
    string text;

    size = 12;
    text = “text++ rules!”;

    for (i =0; i < 3; i = i + 1){

        changeFontSize(“Helvetica”, size);
        write( text );
        size = size + 20;

    }
}
```

C - like programming syntax

Built-in doc styling tools



text++ rules!

text++ rules!

text++ rules!

text.pdf

Features

- **Standard Language Elements**
  - Strongly and statically typed
  - Operators
    - Additive
    - Multiplicative
  - Program entry point: def void start()

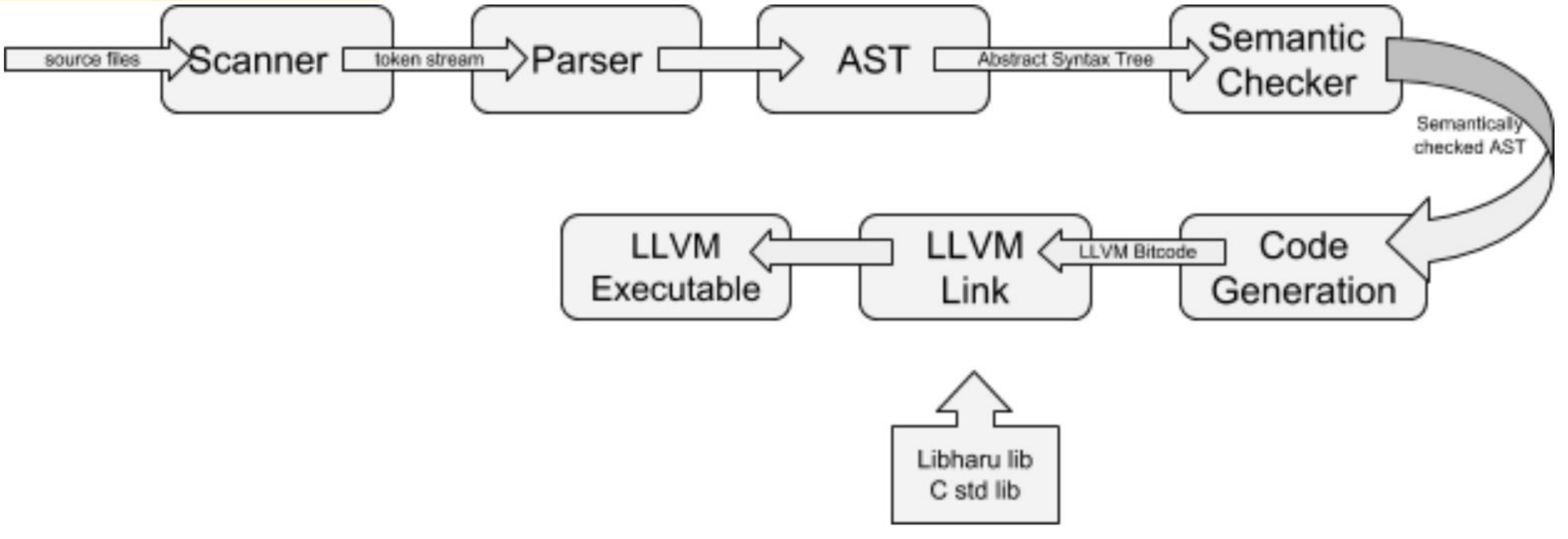
- **text++ Specialties**
  - Generating PDF files with lines
  - Text Color Selection
  - Font and Size Selection

Software Technologies Used

Haru
free PDF library

ORACLE
VirtualBox

Architecture



Introducing text++

“C”-like Syntax

DATA TYPES

- String
- Integer
- Float
- Boolean

VARIABLE DECLARATION + INITIALIZATION

```
int i;
int size;
string text;

size = 12;
text = “text++ rules!”;
```

“C”-like Syntax

FUNCTION DEFINITION

```
def void header (string text) {
    changeFontSize(“Times-Roman”, 24);
    write(text);
}


def void start(){
  header(“Programming Languages and Translators”);
}
```

“C”-like Syntax

FUNCTION CALL

```
def void header (string text) {
    changeFontSize(“Times-Roman”, 24);
    write(text);
}



def void start(){
  header(“Programming Languages and Translators”);
}
```

“C”-like Syntax

OPERATIONS

- Assignment Operator
- Arithmetic Operators
- Boolean Operators
- Bitwise Operators
- Comparison Operators

```
sum = 3 + 2;

true & false;

4 < 7
```

“C”-like Syntax

IF STATEMENT

```
if (<condition>) <stmt>
```

WHILE LOOP

```
while (<condition>) <stmt>
```

FOR LOOP

```
for (<init>; <condition>;
<step>) <statement>
```

CODE BLOCK

```
{ <stmt> }
```

RETURN

```
return x;
```

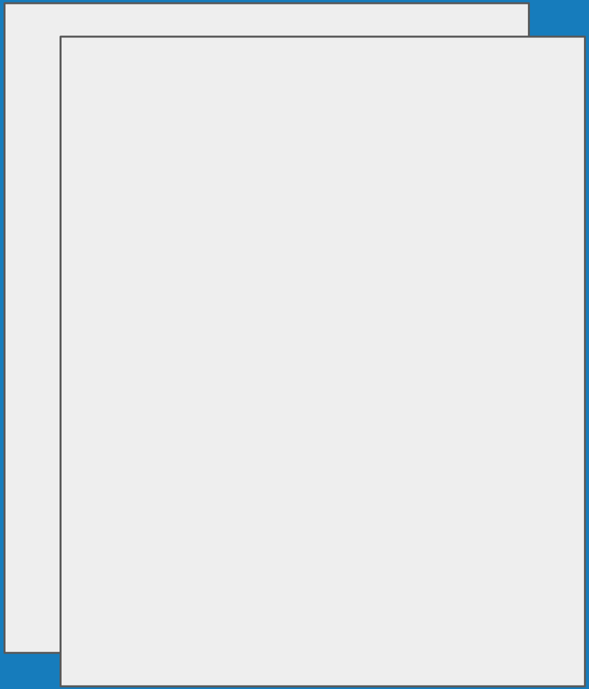
Built in Primitives + Functions

```
def void start(){


   /* Your code lives here! */



 }
```

text.pdf

Built in Primitives + Functions

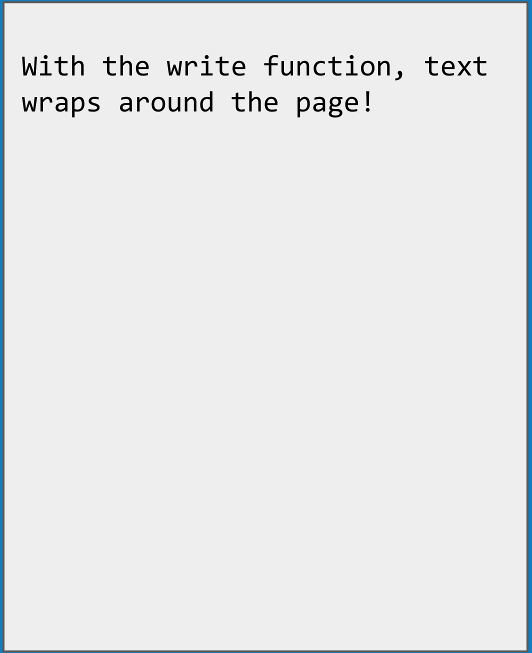
```
def void start(){

    addPage();


 }
```



text.pdf

Built in Primitives + Functions

```
def void start(){

    write(“With the write
function, text wraps around
the page!” );


 }
```

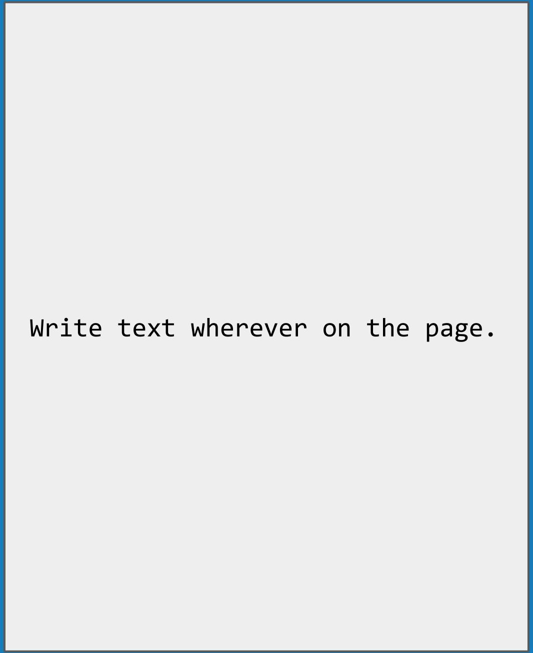


With the write function, text
wraps around the page!

text.pdf

Built in Primitives + Functions

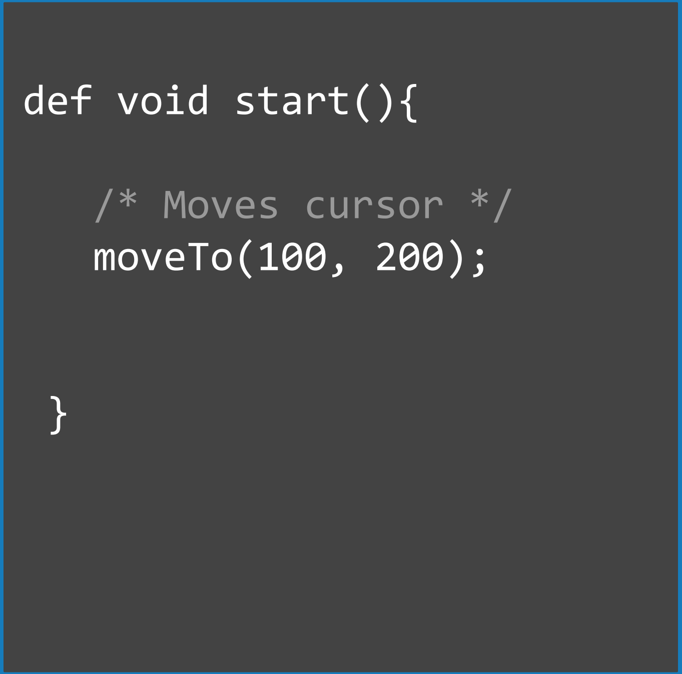
```
def void start(){

    textOut(10, 200, “Write
text wherever on the page.”
);


  }
```



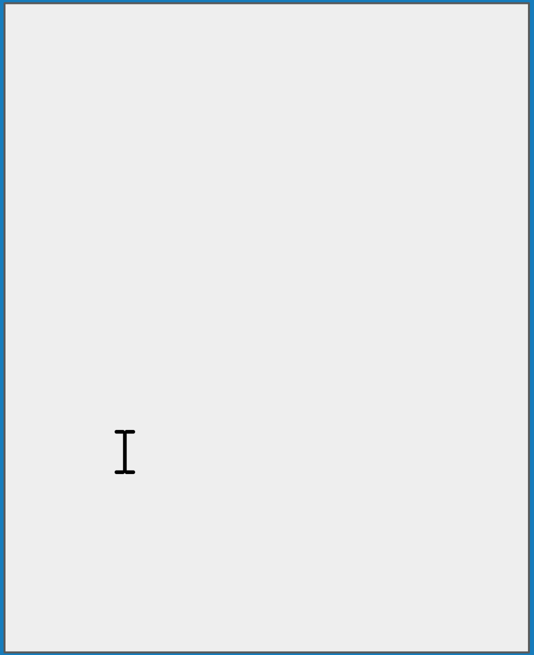
Write text wherever on the page.

text.pdf

Built in Primitives + Functions



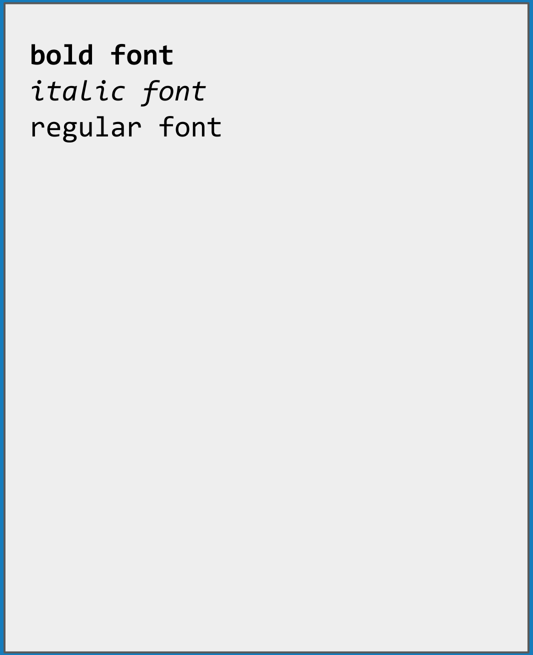
```
def void start(){

   /* Moves cursor */
   moveTo(100, 200);


 }
```



text.pdf

Built in Primitives + Functions

```
def void start(){

   bold();
   write(“bold font”);

   italic();
   write(“italic font”);

   regular();
   write(“regular font”);
 }
```

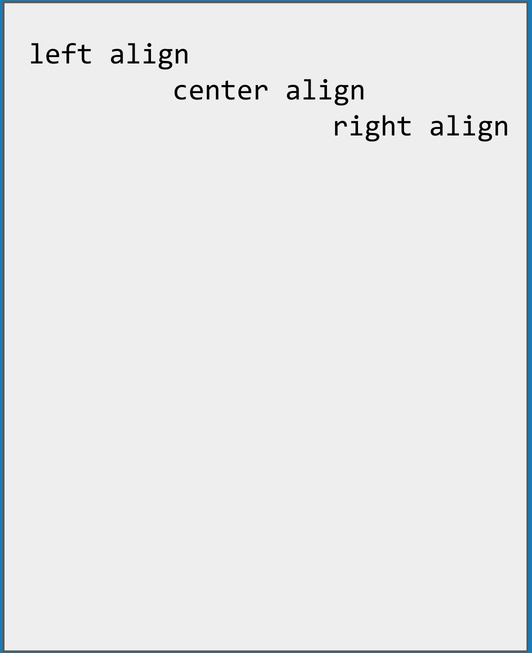


bold font
italic font
regular font

text.pdf

Built in Primitives + Functions

```
def void start(){

   left();
   write(“left align”);

   center();
   write(“center align”);

   right();
   write(“right align”);
 }
```

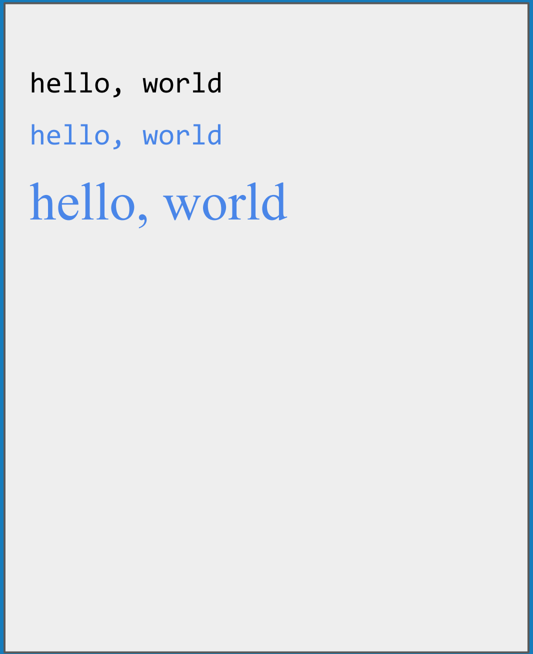


```
left align
         center align
                   right align
```

text.pdf

Built in Primitives + Functions

```
def void start(){

   write(“hello, world”);

   changeColor(0, 0, 1);
   write(“hello, world”);

   changeFontSize(“Times”, 20);
   write(“hello, world”);
 }
```



hello, world

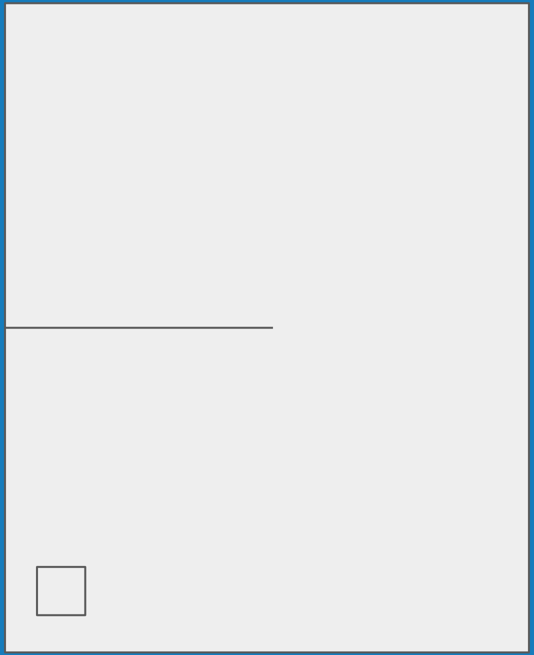
hello, world

hello, world

text.pdf

Built in Primitives + Functions

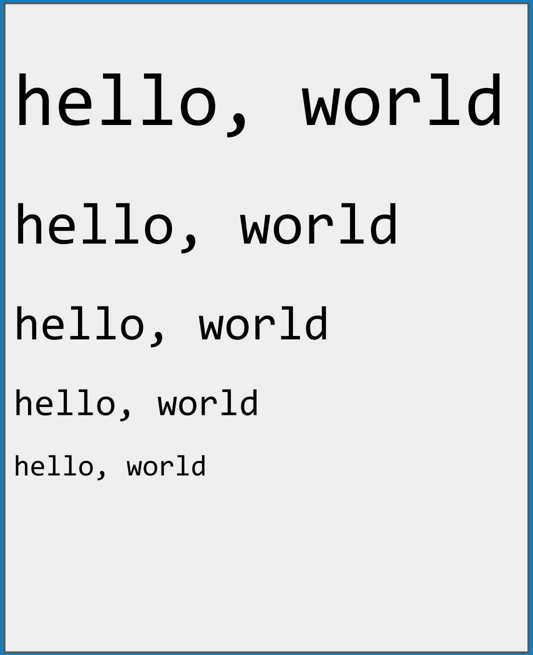
```
def void start(){

    /* Draw lines!*/
    drawLine(0, 200, 200, 0);

    /* Draw rectangles!*/
    drawRectangle(10, 10, 50, 50);

 }
```



text.pdf

Standard Library

```
def void start(){

   heading1();
   write(“hello, world”);
   heading2();
   write(“hello, world”);
   heading3();
   write(“hello, world”);
   heading4();
   write(“hello, world”);
   heading5();
   write(“hello, world”);
   heading6();
   write(“hello, world”);

 }
```



hello, world

hello, world

hello, world

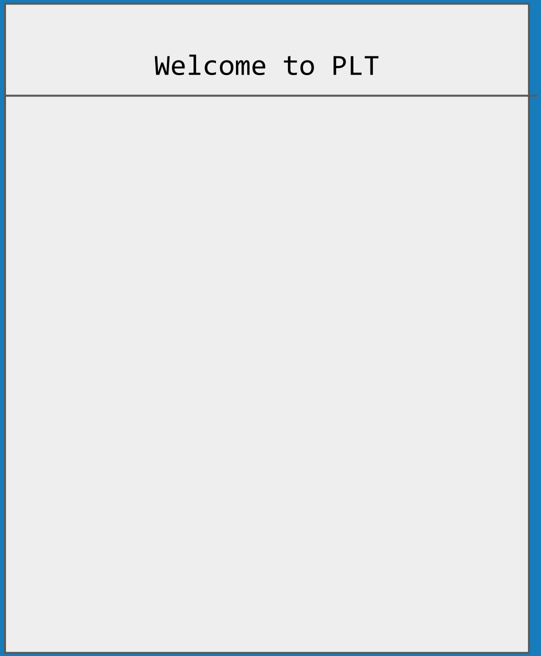
hello, world

hello, world

text.pdf

Standard Library

```
def void start(){

   pageTitle(“Welcome to PLT”);
   horizontalLine();

 }
```

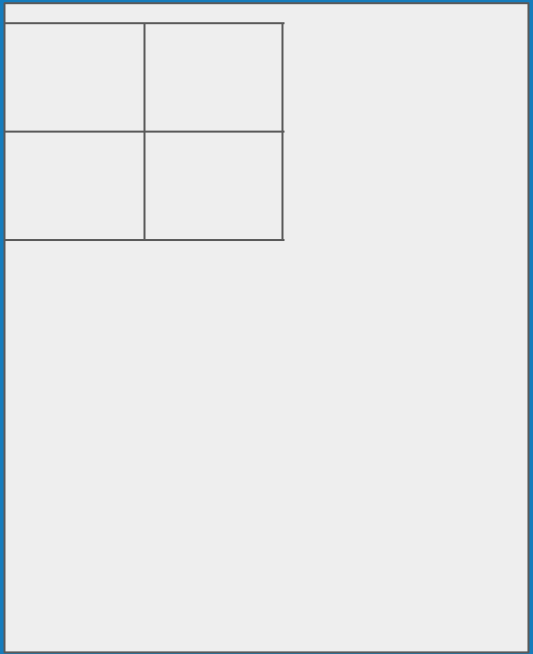


Welcome to PLT

text.pdf

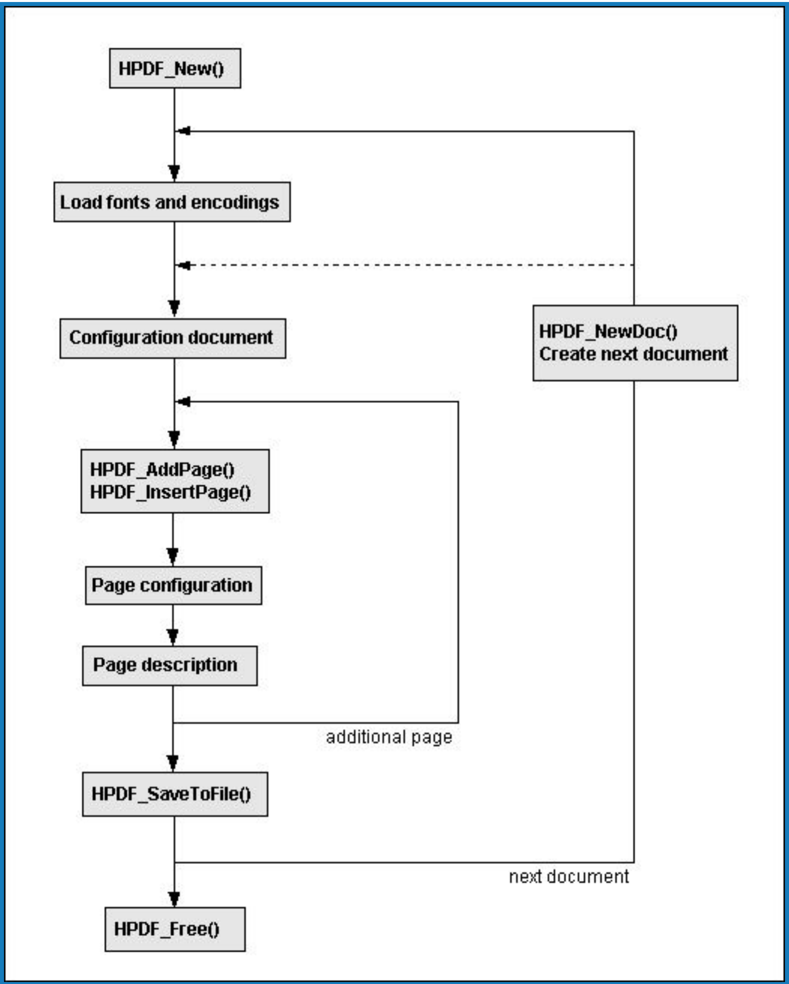
Standard Library

```
def void start(){

    table(2, 2, 200, 400);

 }
```

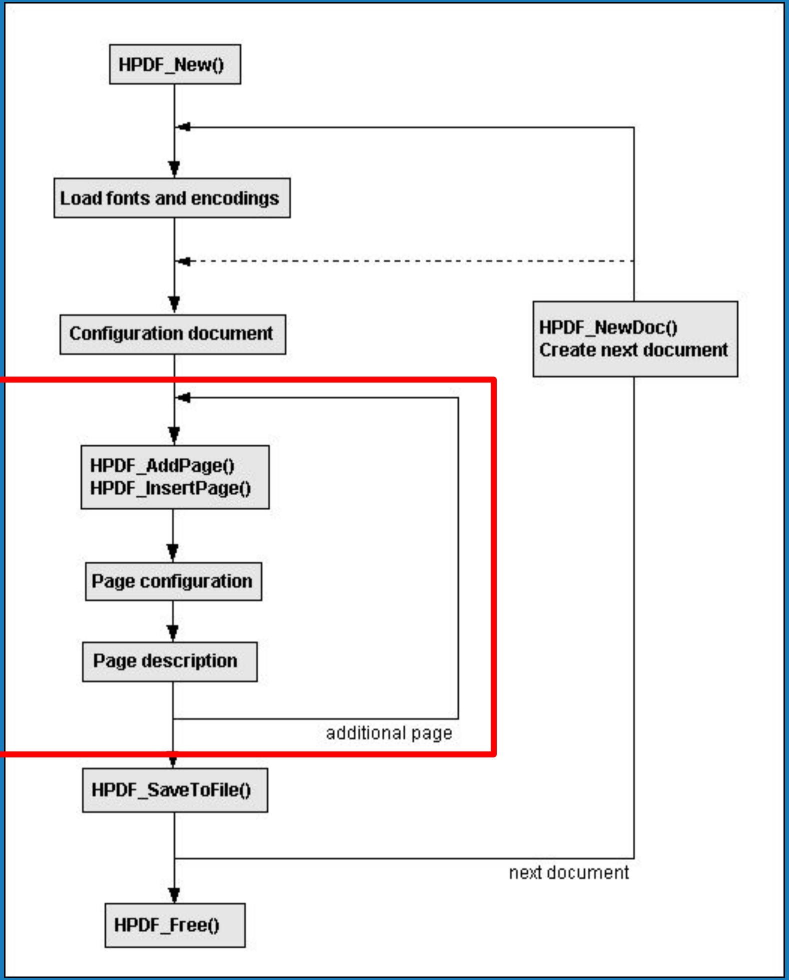


text.pdf

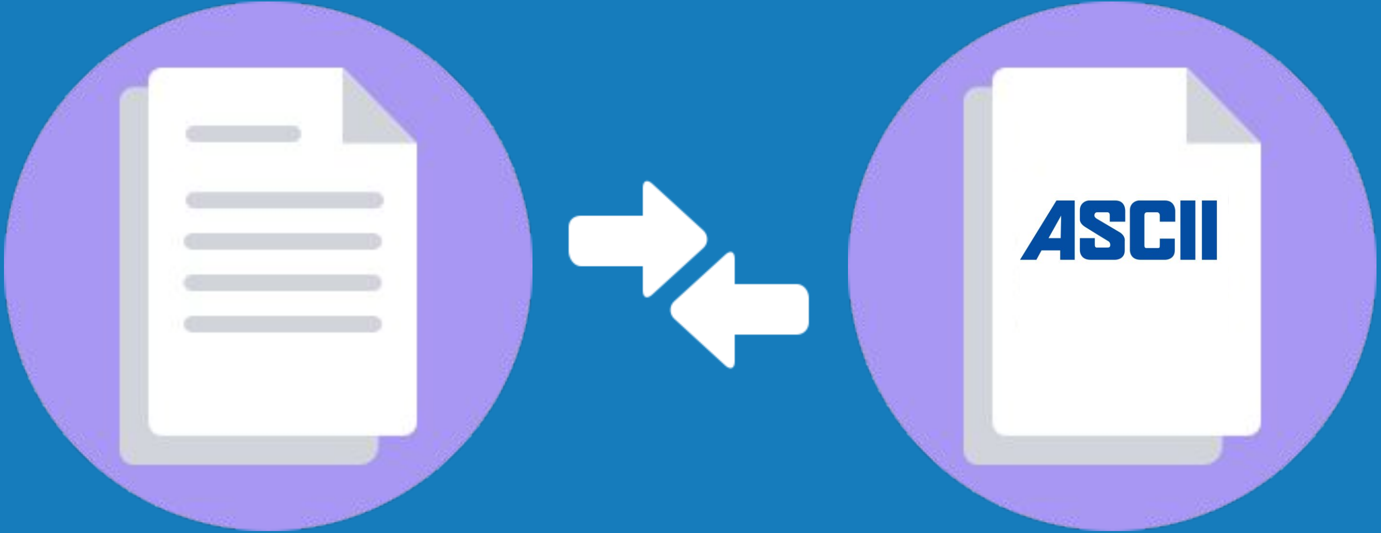
Document Generation Flow



Document Generation Flow

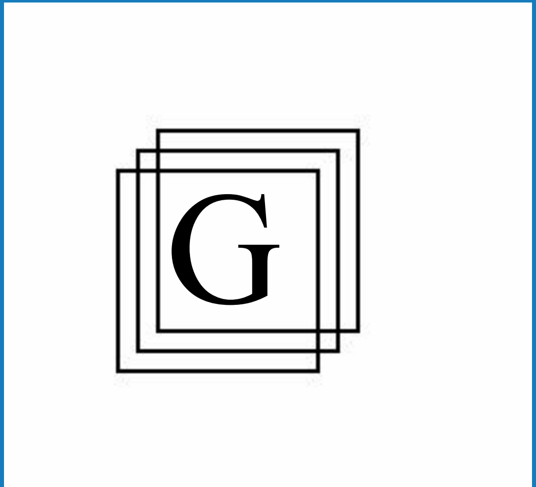


Tests



Sample Program

```
def void logo(){

   int i;
   int ph;
   int pw;
   int offsetX;
   int offsetY;

   pw = getPageWidth();
   ph = getPageHeight();
   offsetX = 10;
   offsetY = 100;

   for (i = 0; i < 3; i = i + 1){
      drawRectangle(pw/2 - offsetX, ph - offsetY, 50, 50);
      offsetX = offsetX + 5;
      offsetY = offsetY + 5;
    }
    heading1();
    textOut( "G" , pw/2 - 5, ph - 90, 0);
}

def void start(){
  logo();
}
```



++ Demo