

# SharpLeaf

## Flowing automated reports to paper

### SharpLeaf Unleashed

**S**harpLeaf is page-based reporting tool providing basic typesetting abilities. It is an object-oriented state machine that allows you to programmatically flow content (text, tables, charts and images), page by page. The philosophy, as opposed to most modern reporting tools, is to keep it simple and clean, rather than complex and dirty.

The goal of this report is to demonstrate some basic features of the SharpLeaf engine, which is why this foreword starts by showing off a dropped capital, and how the text flow nicely wraps around it! We will look at other inclusions later, namely charts and tables.

You will find contact information in the "Contact" section (p. 4) .

#### Flowing text

Here are extracts of the Wikipedia article on typography. In the first section, it will be flown as text, with fine control over the "print head". The second section will just flow regular paragraphs. The third, and last, will demonstrate bullet lists.

##### Overview

Typography is the art and technique of arranging type to make written language readable and appealing. The arrangement of type involves selecting **typefaces**, **point size**, line length, *[censored content]*. Type design is a closely related craft, sometimes considered part of typography: most typographers do not design typefaces, and some type designers do not consider themselves typographers.<sup>[3]</sup> In modern times, typography has been put in film, television and online broadcasts to add emotion to communication.<sup>[4]<sup>[unreliable source?</sup>]</sup>

Typography is performed by typesetters, compositors, typographers, graphic designers, art directors, manga artists, comic book artists, graffiti artists, clerical workers, and everyone else who arranges type for a product. Until the Digital Age, typography was a specialized occupation. Digitization

opened up typography to new generations of visual designers and lay users, and David Jury, Head of Graphic Design at Colchester Institute in England, states that "typography is now something everybody does."<sup>[1]</sup>

##### Basic principles

Legibility is primarily the concern of the typeface designer, to ensure that each individual character or glyph is unambiguous and distinguishable from all other characters in the font. Legibility is also in part the concern of the typographer to select a typeface with appropriate clarity of design for the intended use at the intended size. An example of a well-known design, Brush Script, contains a number of illegible letters, since many of the characters can be easily misread especially if seen out of textual context.

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chinking" and choice of the text architecture of titles, folios, and reference links.

Some commonly agreed findings of legibility research include.<sup>[citation needed]</sup>

1. Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
2. Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
3. Regular upright type (roman type) is found to be more legible than italic type.
4. Contrast, without dazzling brightness, has also been found to be important, with black on yellow/cream being most effective.
5. Positive images (e.g. black on white) are easier to read than negative or reversed (e.g. white on black). However even this commonly accepted practice has some exceptions, for example in some cases of disability.
6. The upper portions of letters play a stronger part than the lower portions in the recognition process.

# What is SharpLeaf ?

- Page-based reporting tool
  - Typographic and tabulation engine
  - Width-constrained downward flow
  - Fill frames and move on

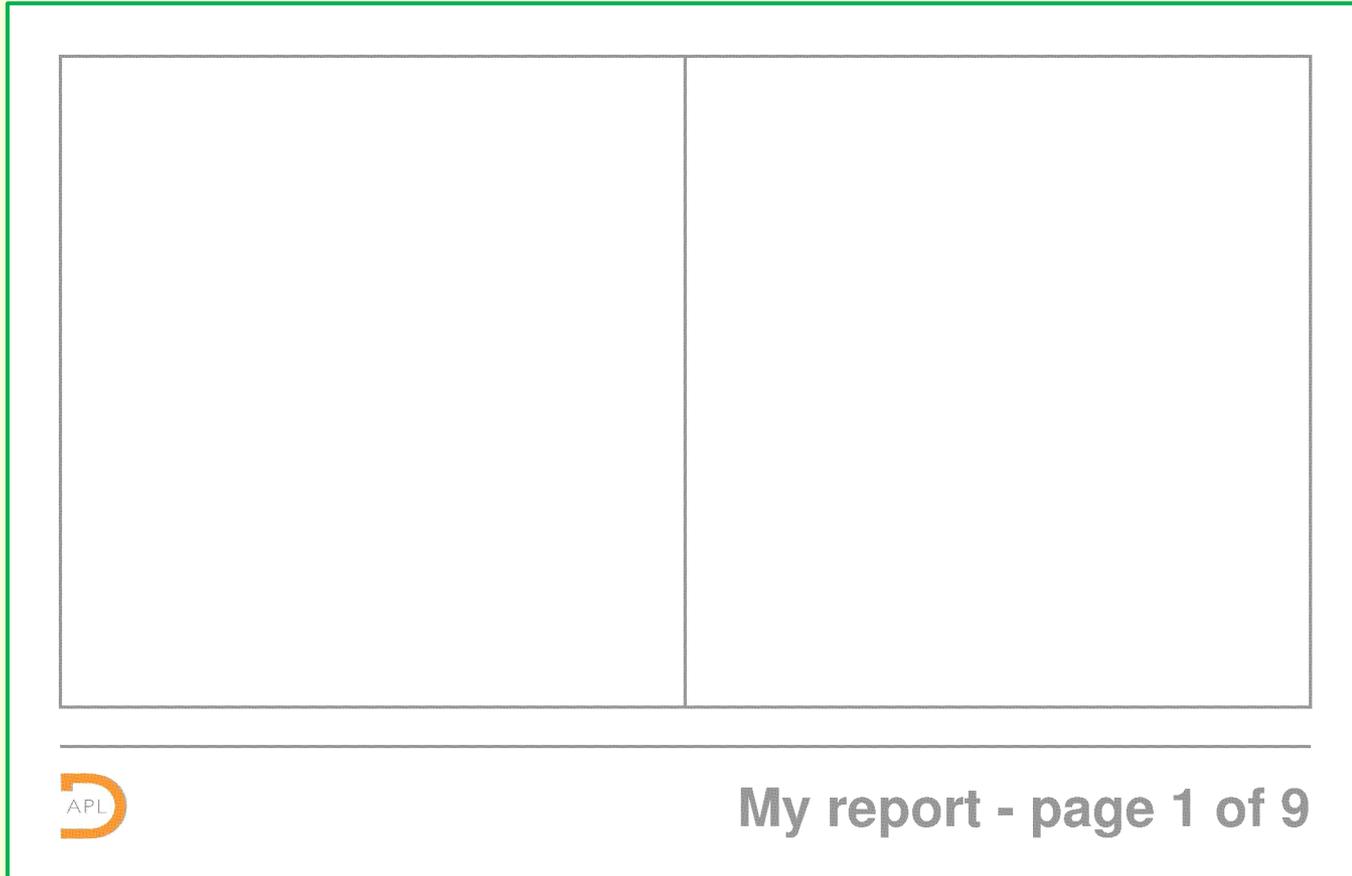


# How to use it ?

- Create Page Layouts (incl. Frames)
- Create Paragraph Styles (incl. Fonts)
- Flow content
  - Move cursor (NextLine, NextPage...)
  - Flow text
  - Include objects (Images, Charts, Tables)
- Output to file (or stream)



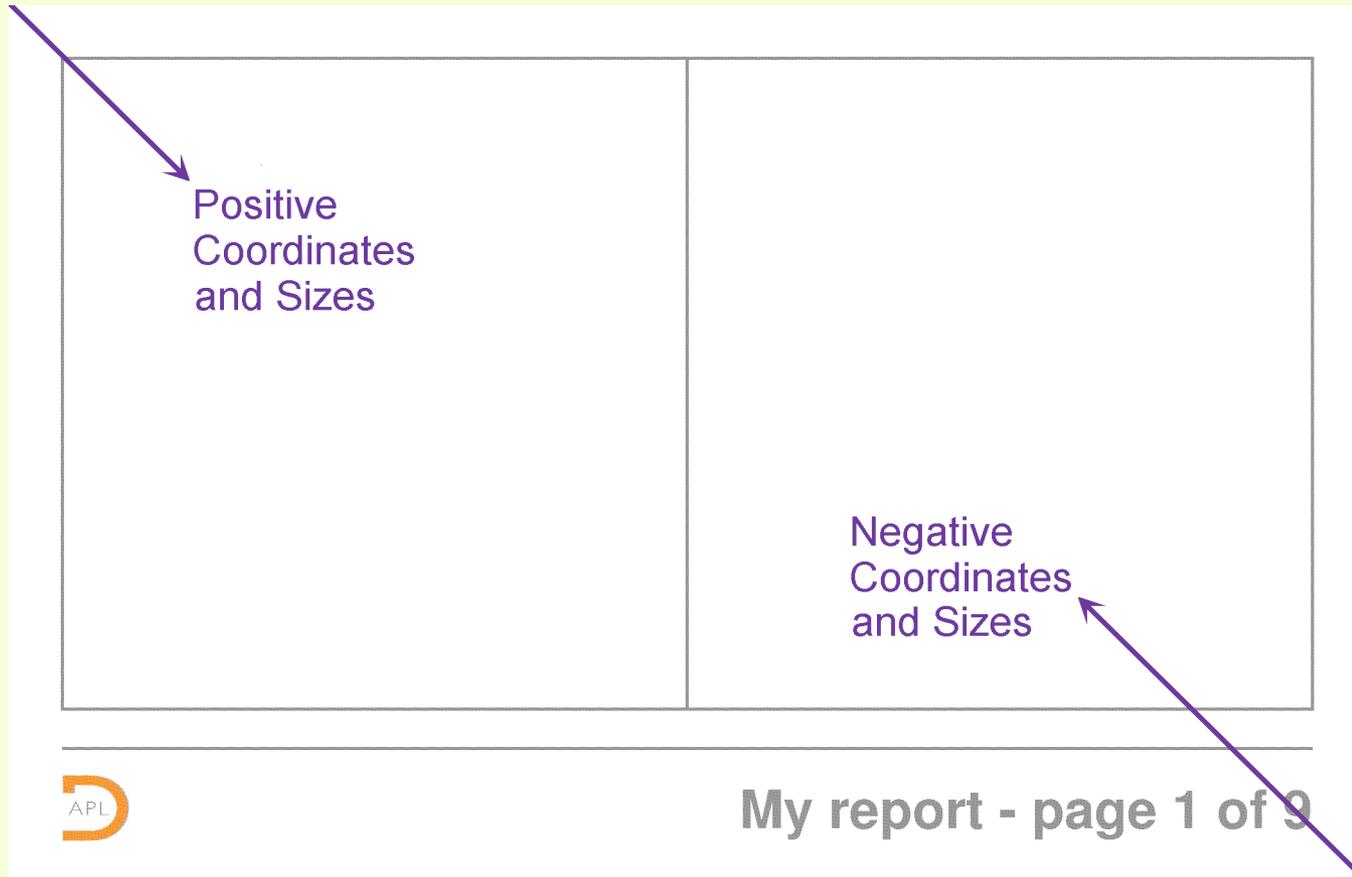
# Document and Page Layout



Paper Size and Fill



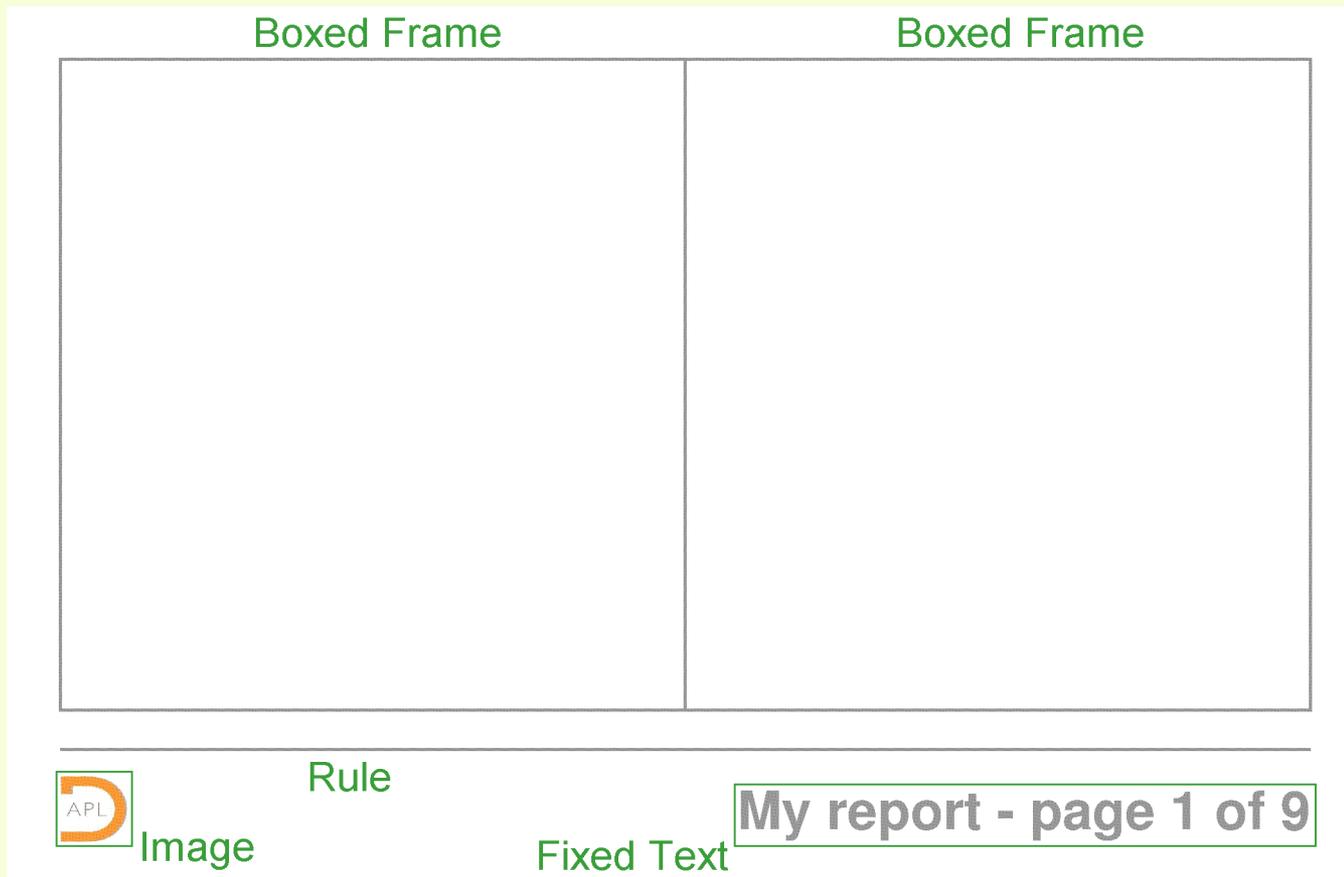
# Document and Page Layout



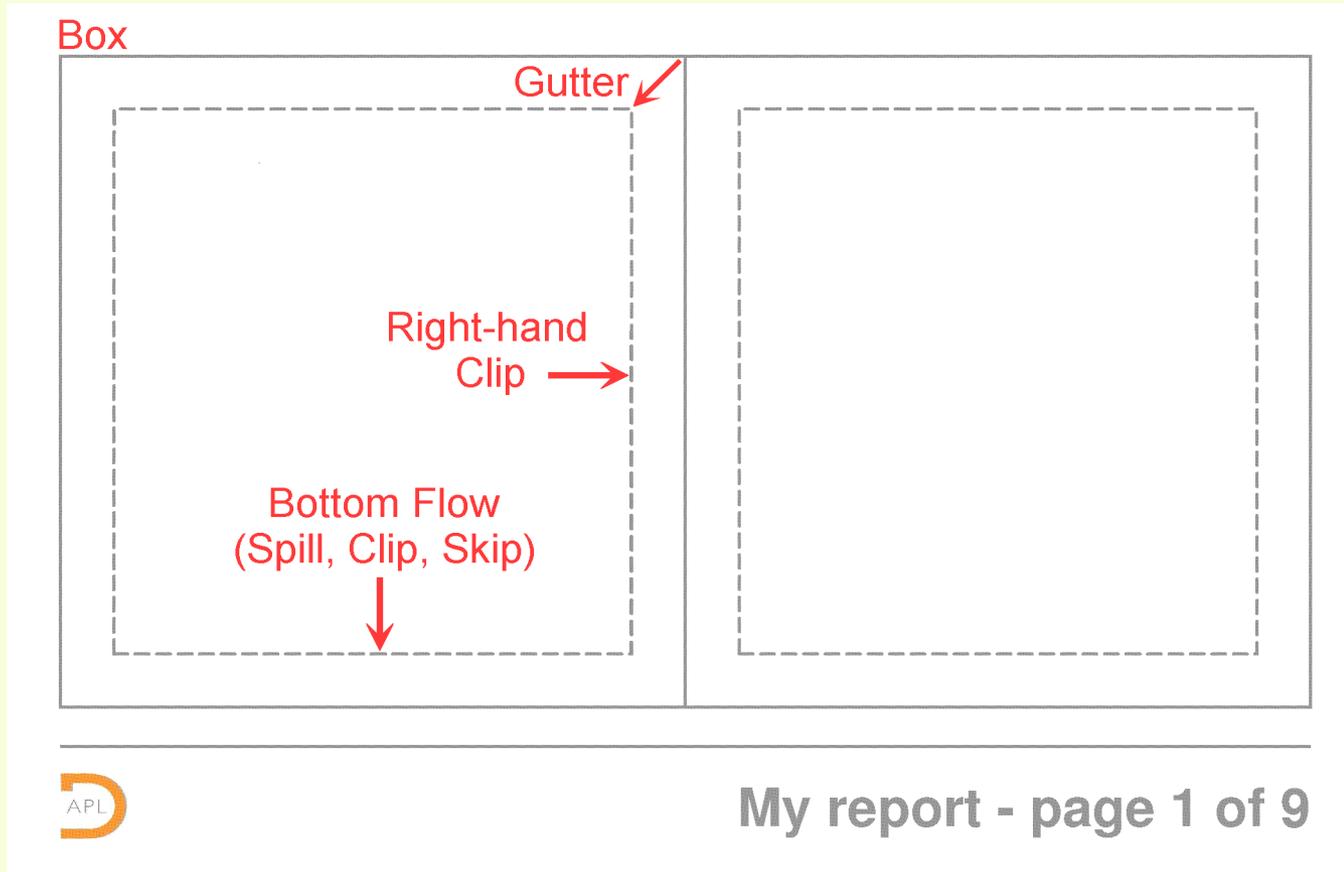
72 points in a inch, 1 point  $\approx$  0.353mm



# Document and Page Layout



# Document and Page Layout



# Document and Page Layout

The mathematical notation for manipulating arrays which developed into the APL programming language was developed by Iverson at Harvard University starting in 1957, and published in his A Programming Language in 1962. [6] The preface states its premise:

Applied mathematics is largely concerned with the design and analysis of explicit procedures for calculating the exact or approximate values of various functions. Such explicit procedures are called algorithms or programs. Because an effective notation for the description of programs exhibits considerable syntactic structure, it is called a programming language.

In 1960, he began work for IBM and, working with Adin Falkoff, created APL based on the notation he had developed. This notation was used inside IBM for short research reports on computer systems, such as the Burroughs B5000 and its stack mechanism when stack machines versus register machines were being evaluated by IBM for upcoming computers.

Also in 1960, Iverson used his notation in a draft of the chapter "A Programming Language", written for a book he was writing with Fred Brooks, Automatic Data Processing, which would be published in 1963. [11] [12]

As early as 1962, the first attempt to use the notation to describe a complete computer system happened after Falkoff discussed with Dr. Wil-

liam C. Carter his work in the standardization of the instruction set for the machines that later became the IBM System/360 family.

In 1963, Herbert Hellerman, working at the IBM Systems Research Institute, implemented a part of the notation on an IBM 1620 computer, and it was used by students in a special high school course on calculating transcendental functions by series summation. Students tested their code in Hellerman's lab. This implementation of a portion of the notation was called PAT (Personalized Array Translator). [13]

In 1963, Falkoff, Iverson, and Edward H. Sussenguth Jr., all working at IBM, used the notation for a formal description of the IBM System/360 series machine architecture and functionality, which resulted in a paper published in IBM Systems Journal in 1964. After this was published, the team turned their attention to an implementation of the notation on a computer system. One of the motivations for this focus of implementation was the interest of John L. Lawrence who had new duties with Science Research Associates, an educational company bought by IBM in 1964. Lawrence asked Iverson and his group to help utilize the language as a tool for the development and use of computers in education. [14]

After Lawrence M. Breed and Philip S. Abrams of Stanford University joined the team at IBM Research, they continued their prior work on an implementation programmed in FORTRAN IV for a portion of the notation was done for the IBM 7090 computer running under the IBSYS operating system. This work was finished in late 1965 and later known



My report - page 1 of 9



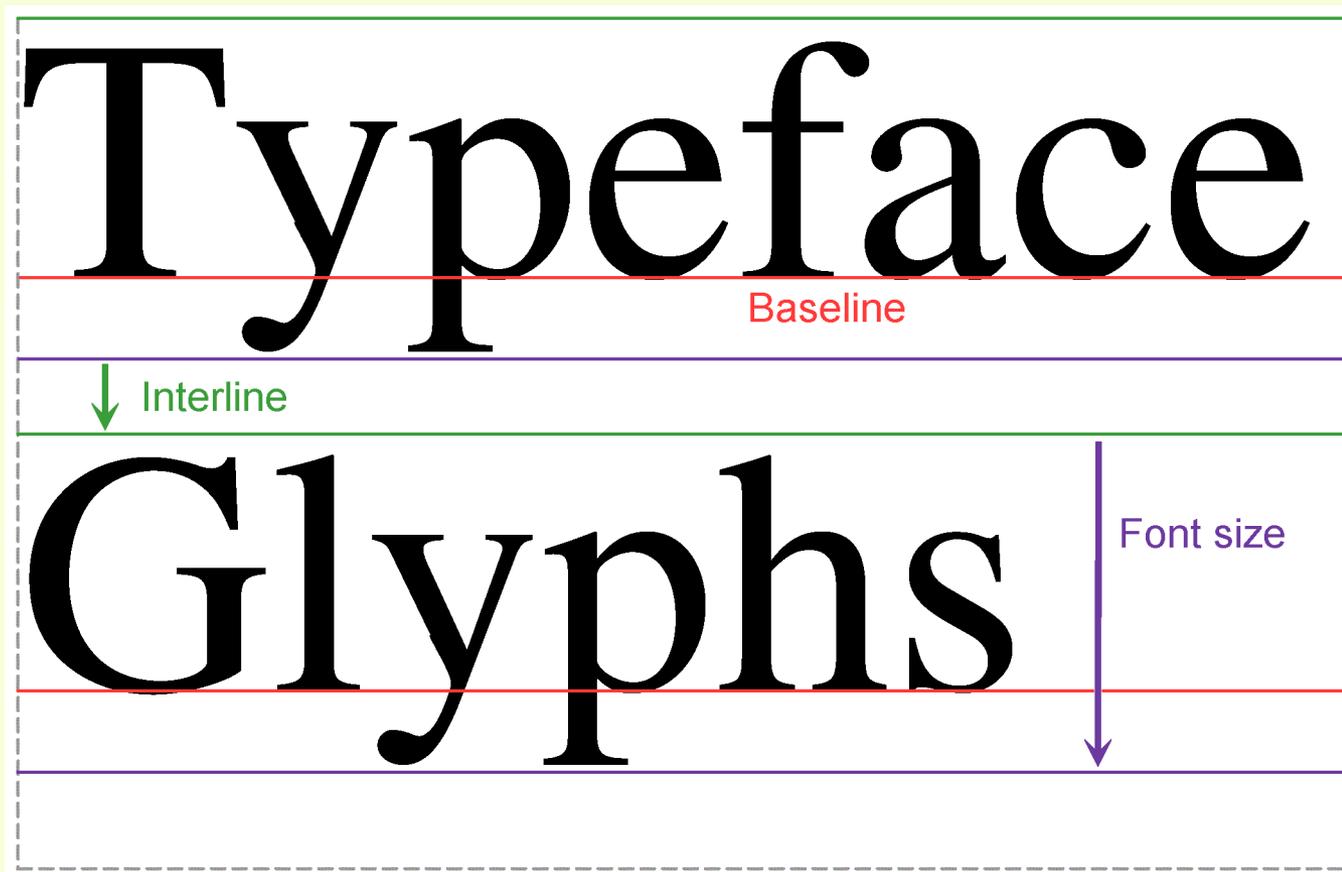
# Fonts

Typeface  
Glyphs

Font Name, Style, Color



# Fonts



# Paragraphs

Typography is the art and technique of arranging type to make written language readable and appealing. Type design is a closely related craft, sometimes considered part of typography; most typographers do not design typefaces, and some type designers do not consider themselves typographers.

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chunking" and choice of the text architecture of titles, folios, and reference links.

1. Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
2. Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
3. Regular upright type (roman type) is found to be more legible than italic type.

Old-style tab stops are supported, but hard to automate:

Name	Salary
John Smith	23,540
Theodore Leopold Abrahamovitch	25,430



# Paragraphs

Typography is the art and technique of arranging type to make written language readable and appealing. Type design is a closely related craft, sometimes considered part of typography; most typographers do not design typefaces, and some type designers do not consider themselves typographers.

## First Indent

## Space Before

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chunking" and choice of the text architecture of titles, folios, and reference links.

## Left Indent

## Space After

## Right Indent

1. Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
2. Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
3. Regular upright type (roman type) is found to be more legible than italic type.

Old-style tab stops are supported, but hard to automate:

Name	Salary
John Smith	23,540
Theodore Leopold Abrahamovitch	25,430



# Paragraphs

Typography is the art and technique of arranging type to make written language readable and appealing. Type design is a closely related craft, sometimes considered part of typography; most typographers do not design typefaces, and some type designers do not consider themselves typographers.

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chunking" and choice of the text architecture of titles, folios, and reference links.

- Bullet** **First Indent**
1.  Text set in lower case is more legible than text set all in upper case (capitals),  
 presumably because lower case letter structures and word shapes are more distinctive.
- Bullet Indent**
2.            Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
3.            Regular upright type (roman type) is found to be more legible than italic type.

Old-style tab stops are supported, but hard to automate:

Name	Salary
John Smith	23,540
Theodore Leopold Abrahamovitch	25,430



# Paragraphs

Typography is the art and technique of arranging type to make written language readable and appealing. Type design is a closely related craft, sometimes considered part of typography; most typographers do not design typefaces, and some type designers do not consider themselves typographers.

Left Alignment

Wrap, Soft Hyphen  
Non-breakable space

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chunking" and choice of the text architecture of titles, folios, and reference links.

1. Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
2. Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
3. Regular upright type (roman type) is found to be more legible than italic type.

Old-style tab stops are supported, but hard to automate:

Name	Salary
John Smith	23,540
Theodore Leopold Abrahamovitch	25,430



# Paragraphs

Typography is the art and technique of arranging type to make written language readable and appealing. Type design is a closely related craft, sometimes considered part of typography; most typographers do not design typefaces, and some type designers do not consider themselves typographers.

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chunking" and choice of the text architecture of titles, folios, and reference links.

1. Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
2. Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
3. Regular upright type (roman type) is found to be more legible than italic type.

Old-style tab stops are supported, but hard to automate:

Name	Salary
John Smith	23,540
Theodore Leopold Abrahamovitch	25,430

Tabs  
Tab Alignments



# Widow and Orphan control

The mathematical notation for manipulating arrays which developed into the APL programming language was developed by Iverson at Harvard University starting in 1957, and published in his A Programming Language in 1962.[6] The preface states its premise:

Applied mathematics is largely concerned with the design and analysis of explicit procedures for calculating the exact or approximate values of various functions. Such explicit procedures are called algorithms or programs. Because an effective notation for the description of programs exhibits considerable syntactic structure, it is called a programming language.

In 1960, he began work for IBM and, working with Adin Falkoff, created APL based on the notation he had developed. This notation was used inside IBM for short research reports on computer systems, such as the Burroughs B5000 and its stack mechanism when stack machines versus register machines were being evaluated by IBM for upcoming computers.

Also in 1960, Iverson used his notation in a draft of the chapter "A Programming Language", written for a book he was writing with Fred Brooks, Automatic Data Processing, which would be published in 1963.[11][12]

As early as 1962, the first attempt to use the notation to describe a complete computer system happened after Falkoff discussed with Dr. William C. Carter his work in the standardization of the instruction set for the machines that later

became the IBM System/360 family.

In 1963, Herbert Hellerman, working at the IBM Systems Research Institute, implemented a part of the notation on an IBM 1620 computer, and it was used by students in a special high school course on calculating transcendental functions by series summation. Students tested their code in Hellerman's lab. This implementation of a portion of the notation was called PAT (Personalized Array Translator).[13]

In 1963, Falkoff, Iverson, and Edward H. Sussenguth Jr., all working at IBM, used the notation for a formal description of the IBM System/360 series machine architecture and functionality, which resulted in a paper published in IBM Systems Journal in 1964. After this was published, the team turned their attention to an implementation of the notation on a computer system. One of the motivations for this focus of implementation was the interest of John L. Lawrence who had new duties with Science Research Associates, an educational company bought by IBM in 1964. Lawrence asked Iverson and his group to help utilize the language as a tool for the development and use of computers in education.[14]

After Lawrence M. Breed and Philip S. Abrams of Stanford University joined the team at IBM Research, they continued their prior work on an implementation programmed in FORTRAN IV for a portion of the notation was done for the IBM 7090 computer running under the IBSYS operating system. This work was finished in late 1965 and later known as IVSYS (Iverson



# Widow and Orphan control

The mathematical notation for manipulating arrays which developed into the APL programming language was developed by Iverson at Harvard University starting in 1957, and published in his A Programming Language in 1962.[6] The preface states its premise:

Applied mathematics is largely concerned with the design and analysis of explicit procedures for calculating the exact or approximate values of various functions. Such explicit procedures are called algorithms or programs. Because an effective notation for the description of programs exhibits considerable syntactic structure, it is called a programming language.

In 1960, he began work for IBM and, working with Adin Falkoff, created APL based on the notation he had developed. This notation was used inside IBM for short research reports on computer systems, such as the Burroughs B5000 and its stack mechanism when stack machines versus register machines were being evaluated by IBM for upcoming computers.

Also in 1960, Iverson used his notation in a draft of the chapter "A Programming Language", written for a book he was writing with Fred Brooks, Automatic Data Processing, which would be published in 1963.[11][12]

As early as 1962, the first attempt to use the notation to describe a complete computer system happened after Falkoff discussed with Dr. William C. Carter his work in the

Keep | standardization of the instruction set for the machines that later became the IBM System/360 family.

In 1963, Herbert Hellerman, working at the IBM Systems Research Institute, implemented a part of the notation on an IBM 1620 computer, and it was used by students in a special high school course on calculating transcendental functions by series summation. Students tested their code in Hellerman's lab. This implementation of a portion of the notation was called PAT (Personalized Array Translator).[13]

In 1963, Falkoff, Iverson, and Edward H. Sussenguth Jr., all working at IBM, used the notation for a formal description of the IBM System/360 series machine architecture and functionality, which resulted in a paper published in IBM Systems Journal in 1964. After this was published, the team turned their attention to an implementation of the notation on a computer system. One of the motivations for this focus of implementation was the interest of John L. Lawrence who had new duties with Science Research Associates, an educational company bought by IBM in 1964. Lawrence asked Iverson and his group to help utilize the language as a tool for the development and use of computers in education.[14]

After Lawrence M. Breed and Philip S. Abrams of Stanford University joined the team at IBM Research, they continued their prior work on an implementation programmed in FORTRAN IV for a portion of the notation was done for the IBM 7090 computer running under the IBSYS operating system. This work was



# Widow and Orphan control

## Definitions

Typography is the art and technique of arranging type to make written language readable and appealing. Type design is a closely related craft, sometimes considered part of typography; most typographers do not design typefaces, and some type designers do not consider themselves typographers.

Legibility is primarily the concern of the typeface designer, to ensure that each individual character or glyph is unambiguous and distinguishable from all other characters in the font. Legibility is also in part the concern of the typographer to select a typeface with appropriate clarity of design for the intended use at the intended size. An example of a well-known design, Brush Script, contains a number of illegible letters, since many of the characters can be easily misread especially if seen out of textual context.

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chunking" and choice of the text architecture of titles, folios, and reference links.

## Tips on legibility

- I - Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
- II - Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
- III - Regular upright type (roman type) is found to be more legible than italic type.
- IV - Contrast, without dazzling brightness, has also been found to be important, with black on yellow/cream being most effective.
- V - Positive images (e.g. black on white) are easier to read than negative or reversed (e.g. white on black). However even this commonly accepted practice has some exceptions, for example in some cases of disability.
- VI - The upper portions of letters play a stronger part than the lower portions in the recognition process.



# Widow and Orphan control

## Definitions

Typography is the art and technique of arranging type to make written language readable and appealing. Type design is a closely related craft, sometimes considered part of typography; most typographers do not design typefaces, and some type designers do not consider themselves typographers.

Legibility is primarily the concern of the typeface designer, to ensure that each individual character or glyph is unambiguous and distinguishable from all other characters in the font. Legibility is also in part the concern of the typographer to select a typeface with appropriate clarity of design for the intended use at the intended size. An example of a well-known design, Brush Script, contains a number of illegible letters, since many of the characters can be easily misread especially if seen out of textual context.

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chunking" and choice of the text architecture of titles, folios, and reference links.

## Tips on legibility

### Space Following

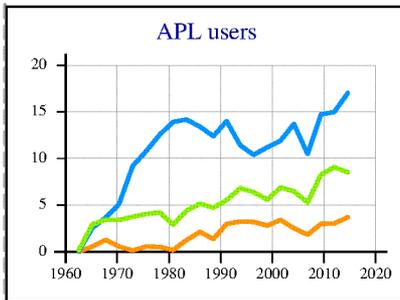
- I - Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
- II - Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
- III - Regular upright type (roman type) is found to be more legible than italic type.
- IV - Contrast, without dazzling brightness, has also been found to be important, with black on yellow/cream being most effective.
- V - Positive images (e.g. black on white) are easier to read than negative or reversed (e.g. white on black). However even this commonly accepted practice has some exceptions, for example in some cases of disability.
- VI - The upper portions of letters play a stronger part than the lower portions in the recognition process.



# Inclusions



The mathematical notation for manipulating arrays which developed into the APL programming language was developed by Iverson at Harvard University starting in 1957, and published in his A Programming Language in 1962. [6] The preface states its premise:



Applied mathematics is largely concerned with the design and analysis of explicit procedures for calculating the exact or approximate values of various functions. Such explicit procedures are called algorithms or programs. Because an effective notation for the description of programs exhibits considerable syntactic structure, it is called a programming language.

In 1960, he began work for IBM and, working with Adin Falkoff, created APL based on the notation he had developed. This notation

was used inside IBM for short research reports on computer systems, such as the Burroughs B5000 and its stack mechanism when stack machines versus register machines were being evaluated by IBM for upcoming computers.

Jan	65.64	27.87	10.18
Feb	33.33	46.58	70.66
Mar	20.58	59.38	65.17
Apr	80	67.77	33.04
May	31.62	23.72	72.66
June	16.02	27.94	67.3
July	10.25	12.74	1.21
Aug	98.34	28.1	56.8
Sept	79.45	81.54	71.21

Also in 1960, Iverson used his notation in a draft of the chapter "A Programming Language", written for a book he was writing with Fred Brooks, Automatic Data Processing, which would be published in 1963. [11] [12]



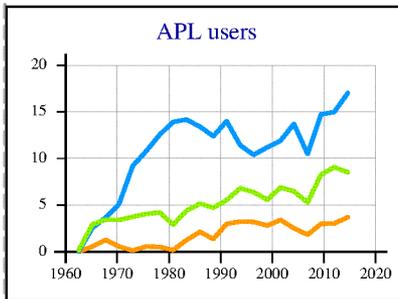
# Inclusions



## Image as Paragraph

The mathematical notation for manipulating arrays which developed into the APL programming language was developed by Iverson at Harvard University starting in 1957, and published in his A Programming Language in 1962. [6] The preface states its premise:

### Chart on the Left



tion was used inside IBM for short research reports on computer systems, such as the Burroughs B5000 and its stack mechanism when stack machines versus register machines were being evaluated by IBM for upcoming computers.

Also in 1960, Iverson used his notation in a draft of the chapter "A Programming Language", written for a book he was writing with Fred Brooks, Automatic Data Processing, which would be published in 1963. [11] [12]

Applied mathematics is largely concerned with the design and analysis of explicit procedures for calculating the exact or approximate values of various functions. Such explicit procedures are called algorithms or programs. Because an effective notation for the description of programs exhibits considerable syntactic structure, it is called a programming language.

### Table on the Right

Jan	65.64	27.87	10.18
Feb	33.33	46.58	70.66
Mar	20.58	59.38	65.17
Apr	80	67.77	33.04
May	31.62	23.72	72.66
June	16.02	27.94	67.3
July	10.25	12.74	1.21
Aug	98.34	28.1	56.8
Sept	79.45	81.54	71.21



# Decorations

~~And~~ and now let's ~~ruin~~ enhance the readability of our document, and show off with all sorts of typographic eye-candy, such as **C***r***a***z***Y** inlined font changes.

---

More reasonably, we can also include <sup>superscripts</sup> and <sub>subscripts</sub> and also combinations of the two such as M<sup>i</sup><sub>j</sub> that align as nicely as you could expect. *[citation needed]*

**WE CAN ALSO BOX AND CENTER  
PARAGRAPHS TO MAKE THEM APPEAR  
MORE IMPORTANT.**



# Decorations

Dropped  
Capital  
on 2 lines

And now let's ~~ruin~~ enhance the readability of our document, and show off with all sorts of typographic eye-candy, such as **C**razy inlined font changes.

---

Rule

More reasonably, we can also include <sup>superscripts</sup> and <sub>subscripts</sub> and also combinations of the two such as M<sup>i</sup><sub>j</sub> that align as nicely as you could expect. *[citation needed]*

**WE CAN ALSO BOX AND CENTER  
PARAGRAPHS TO MAKE THEM APPEAR  
MORE IMPORTANT.**

Box



# Tables

**Table Heading**  
*with a subheading*

Table may also have a caption

	Left	Center	Right	Decimal
<b>Top</b>	Cell with explicit newlines	Cell that doesn't wrap nor clip, spilling all over the place	123.45	N/A
<b>Middle</b>	Cell that forces a large column width	Cell that wraps nicely on multiple lines	Boxed cell	Cell with explicit newlines : A = 3.1416 B = 27 C = 99.9
<b>Bottom</b>	Fill character.....	<i>Fancy Font</i>	Cell that doesn't wrap but	Cell that wraps and ends with a decimal value = 123.45  (and a footnote too)



# Table-wide settings

**Table Heading**  
*with a subheading*

**Titles, Offset, Retain**  
Table may also have a caption

**Table box**

	<b>Left</b>	<b>Center</b>	<b>Right</b>	<b>Decimal</b>
<b>Top</b>	Cell with explicit newlines	Cell that doesn't wrap nor clip, spilling all over the place	123.45	N/A
<b>Middle</b>	Cell that forces a large column width	Cell that wraps nicely on multiple lines	Boxed cell	Cell with explicit newlines : A = 3.1416 B = 27 C = 99.9
<b>Bottom</b>	Fill character.....	<i>Fancy Font</i>	Cell that doesn't wrap but	Cell that wraps and ends with a decimal value = 123.45  (and a footnote too)

**Cell Gutter**

↔ **Column Min and Max Width** ↔



# Column- and Row-wise settings

**Table Heading**  
*with a subheading*

Table may also have a caption

	Left	Center	Right	Decimal
Top	Cell with explicit newlines	Cell that doesn't wrap nor clip, spilling all over the place	123.45	N/A
Middle	Cell that forces a large column width	Cell that wraps nicely on multiple lines	Boxed cell	Cell with explicit newlines : A = 3.1416 B = 27 C = 99.9
Bottom	Fill character.....	<i>Fancy Font</i>	Cell that doesn't wrap but	Cell that wraps and ends with a decimal value = 123.45

(and a footnote too)

Column Widths (Fixed, Relative, Autofit)

Minimum Row Heights

Column and Row Grids (Width, Style, Color)



# Cell-wise settings

Table Heading  
*with a subheading*

Table may also have a caption

	Left	Center	Right	Decimal
Top	Cell with explicit newlines	Cell that doesn't wrap nor clip, spilling all over the place	123.45	N/A
Middle	Cell that forces a large column width	Cell that wraps nicely on multiple lines	Boxed cell	Cell with explicit newlines : A = 3.1416 B = 27 C = 99.9
Bottom	Fill character.....	<i>Fancy Font</i>	Cell that doesn't wrap but	Cell that wraps and ends with a decimal value = 123.45  (and a footnote too)

Font, Box  
 Alignment, Wrap, Clip  
 Fill Character



# Filling tables

**Table Heading**

Caption

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	Totals
Denmark	0.22	9.75	7	6.39	7.49	7.1	3.77	7.38	5.18	1.94	8.78	4.94	1.26	7.15	7.11	85.46
France	3.56	6.62	0.1	4.51	5.89	1.18	0.55	8.79	0.42	8.97	5.9	6.96	1.66	4.27	5.48	64.86
Germany	7.37	1.47	2.55	6.45	2.96	4.15	4.92	1.15	7.63	1.26	1.89	5.76	5.7	4.16	4.5	61.92
Italy	1.13	6.68	0.33	7.51	2.64	2.59	1.01	2.63	6.48	1.71	5.24	3.39	5.99	9.84	5.46	62.63
United Kingdom	2.36	5.29	2.75	2.24	4.4	9.92	2.7	2.44	5.47	8.69	0.46	4.69	5.68	8.63	7.57	73.29
Canada	3.42	1.84	2.12	1.47	4.55	4.7	5.14	0.8	3.36	5.62	4.44	0.64	7.75	4.35	4.34	54.54
USA	7.71	1.08	5.27	8.11	0.93	1.05	5.72	4.9	9.92	0.29	3.27	3.14	9.2	9.9	2.56	73.05
China	6.1	5.29	9.12	7.22	8.55	6.74	3.61	0.46	1.79	9.31	7.76	4.12	8.58	1.95	5.61	86.21
India	4.67	9.24	0.37	4.8	8.25	2.17	1.92	7.76	7.42	5.25	7.17	3.05	8.94	1.4	9.83	82.24
Japan	3.58	2.13	7.49	5.59	0.34	7.56	3.95	3.36	9.32	1.63	0.04	8.4	2.75	9.42	1.47	67.03

Footnote



# Filling tables

**Table Heading**

Caption

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	Totals
Denmark	0.22	9.75	7	6.39	7.49	7.1	3.77	7.38	5.18	1.94	8.78	4.94	1.26	7.15	7.11	85.46
France	3.56	6.62	0.1	4.51	5.89	1.18	0.55	8.79	0.42	8.97	5.9	6.96	1.66	4.27	5.48	64.86
Germany	7.37	1.47	2.55	6.45	2.96	4.15	4.92	1.15	7.63	1.26	1.89	5.76	5.7	4.16	4.5	61.92
Italy	1.13	6.68	0.33	7.51	2.64	2.59	1.01	2.63	6.48	1.71	5.24	3.39	5.99	9.84	5.46	62.63
United Kingdom	2.36	5.29	2.75	2.24	4.4	9.92	2.7	2.44	5.47	8.69	0.46	4.69	5.68	8.63	7.57	73.29
Canada	3.42	1.84	2.12	1.47	4.55	4.7	5.14	0.8	3.36	5.62	4.44	0.64	7.75	4.35	4.34	54.54
USA	7.71	1.08	5.27	8.11	0.93	1.05	5.72	4.9	9.92	0.29	3.27	3.14	9.2	9.9	2.56	73.05
China	6.1	5.29	9.12	7.22	8.55	6.74	3.61	0.46	1.79	9.31	7.76	4.12	8.58	1.95	5.61	86.21
India	4.67	9.24	0.37	4.8	8.25	2.17	1.92	7.76	7.42	5.25	7.17	3.05	8.94	1.4	9.83	82.24
Japan	3.58	2.13	7.49	5.59	0.34	7.56	3.95	3.36	9.32	1.63	0.04	8.4	2.75	9.42	1.47	67.03

Footnote

Set Cell Alignment to Center

Add Rows



# Filling tables

**Table Heading**

Caption

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	Totals
Denmark	0.22	9.75	7	6.39	7.49	7.1	3.77	7.38	5.18	1.94	8.78	4.94	1.26	7.15	7.11	85.46
France	3.56	6.62	0.1	4.51	5.89	1.18	0.55	8.79	0.42	8.97	5.9	6.96	1.66	4.27	5.48	64.86
Germany	7.37	1.47	2.55	6.45	2.96	4.15	4.92	1.15	7.63	1.26	1.89	5.76	5.7	4.16	4.5	61.92
Italy	1.13	6.68	0.33	7.51	2.64	2.59	1.01	2.63	6.48	1.71	5.24	3.39	5.99	9.84	5.46	62.63
United Kingdom	2.36	5.29	2.75	2.24	4.4	9.92	2.7	2.44	5.47	8.69	0.46	4.69	5.68	8.63	7.57	73.29
Canada	3.42	1.84	2.12	1.47	4.55	4.7	5.14	0.8	3.36	5.62	4.44	0.64	7.75	4.35	4.34	54.54
USA	7.71	1.08	5.27	8.11	0.93	1.05	5.72	4.9	9.92	0.29	3.27	3.14	9.2	9.9	2.56	73.05
China	6.1	5.29	9.12	7.22	8.55	6.74	3.61	0.46	1.79	9.31	7.76	4.12	8.58	1.95	5.61	86.21
India	4.67	9.24	0.37	4.8	8.25	2.17	1.92	7.76	7.42	5.25	7.17	3.05	8.94	1.4	9.83	82.24
Japan	3.58	2.13	7.49	5.59	0.34	7.56	3.95	3.36	9.32	1.63	0.04	8.4	2.75	9.42	1.47	67.03

Footnote

Set Cell Alignment to Left

Add Columns



# Filling tables

**Table Heading**

Caption

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	Totals
Denmark	0.00	0.75	7	0.00	7.10	7.1	0.77	7.00	5.10	1.04	0.70	1.04	1.00	7.15	7	85.46
France	0.50	0.00	0.1	1.51	5.00	1.10	0.55	0.70	0.10	0.07	5.0	0.00	1.00	1.07	0	64.86
Germany	7.07	1.17	0.55	0.15	0.00	1.15	1.00	1.15	7.00	1.00	1.00	5.70	5.7	1.10	1	61.92
Italy	1.10	0.00	0.00	7.51	0.01	0.50	1.01	0.00	0.10	1.71	5.01	0.00	5.00	0.01	0	62.63
United Kingdom	0.00	5.00	0.75	0.01	1.1	0.00	0.7	0.11	5.17	0.00	0.10	1.00	5.00	0.00	7	73.29
Canada	0.10	1.01	0.10	1.17	1.55	1.7	5.11	0.0	0.00	1.11	0.01	7.75	1.05	1	1	54.54
USA	7.71	1.00	5.07	0.11	0.00	1.05	5.70	1.0	0.00	0.00	0.07	0.11	0.0	0.0	0	73.05
China	0.1	5.00	0.10	7.00	0.55	0.71	0.01	0.10	1.70	0.01	7.70	1.10	0.50	1.05	0	86.21
India	1.07	0.01	0.07	1.0	0.05	0.17	1.00	7.70	7.10	5.05	7.17	0.05	0.01	1.1	0	82.24
Japan	0.50	0.10	7.10	5.50	0.01	7.50	0.05	0.00	0.00	1.00	0.01	0.1	0.75	0.10	1	67.03

Footnote

Set Cell Alignment to Decimal

Set Row-wise Cell Box to (5 3 2/Green Blue Yellow)

Add Rows



# Filling tables

**Table Heading**

Caption	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	Totals
Denmark	0.22	9.75	7	6.39	7.49	7.1	3.77	7.38	5.18	1.94	8.78	4.94	1.26	7.15	7.11	85.46
France	3.56	6.62	0.1	4.51	5.89	1.18	0.55	8.79	0.42	8.97	5.9	6.96	1.66	4.27	5.48	64.86
Germany	7.37	1.47	2.55	6.45	2.96	4.15	4.92	1.15	7.63	1.26	1.89	5.76	5.7	4.16	4.5	61.92
Italy	1.13	6.68	0.33	7.51	2.64	2.59	1.01	2.63	6.48	1.71	5.24	3.39	5.99	9.84	5.46	62.63
United Kingdom	2.36	5.29	2.75	2.24	4.4	9.92	2.7	2.44	5.47	8.69	0.46	4.69	5.68	8.63	7.57	73.29
Canada	3.42	1.84	2.12	1.47	4.55	4.7	5.14	0.8	3.36	5.62	4.44	0.64	7.75	4.35	4.34	54.54
USA	7.71	1.08	5.27	8.11	0.93	1.05	5.72	4.9	9.92	0.29	3.27	3.14	9.2	9.9	2.56	73.05
China	6.1	5.29	9.12	7.22	8.55	6.74	3.61	0.46	1.79	9.31	7.76	4.12	8.58	1.95	5.61	86.21
India	4.67	9.24	0.37	4.8	8.25	2.17	1.92	7.76	7.42	5.25	7.17	3.05	8.94	1.4	9.83	82.24
Japan	3.58	2.13	7.49	5.59	0.34	7.56	3.95	3.36	9.32	1.63	0.04	8.4	2.75	9.42	1.47	67.03

Footnote

Go to Next Empty Column starting at Row 2

Set Cell Box to Grey

Add Columns



# Flowing tables

**Table Heading**

Caption

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	Totals
Denmark	0.22	9.75	7	6.39	7.49	7.1	3.77	7.38	5.18	1.94	8.78	4.94	1.26	7.15	7.11	85.46
France	3.56	6.62	0.1	4.51	5.89	1.18	0.55	8.79	0.42	8.97	5.9	6.96	1.66	4.27	5.48	64.86
Germany	7.37	1.47	2.55	6.45	2.96	4.15	4.92	1.15	7.63	1.26	1.89	5.76	5.7	4.16	4.5	61.92
Italy	1.13	6.68	0.33	7.51	2.64	2.59	1.01	2.63	6.48	1.71	5.24	3.39	5.99	9.84	5.46	62.63
United Kingdom	2.36	5.29	2.75	2.24	4.4	9.92	2.7	2.44	5.47	8.69	0.46	4.69	5.68	8.63	7.57	73.29
Canada	3.42	1.84	2.12	1.47	4.55	4.7	5.14	0.8	3.36	5.62	4.44	0.64	7.75	4.35	4.34	54.54
USA	7.71	1.08	5.27	8.11	0.93	1.05	5.72	4.9	9.92	0.29	3.27	3.14	9.2	9.9	2.56	73.05
China	6.1	5.29	9.12	7.22	8.55	6.74	3.61	0.46	1.79	9.31	7.76	4.12	8.58	1.95	5.61	86.21
India	4.67	9.24	0.37	4.8	8.25	2.17	1.92	7.76	7.42	5.25	7.17	3.05	8.94	1.4	9.83	82.24
Japan	3.58	2.13	7.49	5.59	0.34	7.56	3.95	3.36	9.32	1.63	0.04	8.4	2.75	9.42	1.47	67.03

Footnote

1 Frame Accross



# Flowing tables

**Table Heading**

Caption

	2001	2002	2003	2004	Totals
Denmark	0.22	9.75	7	6.39	85.46
France	3.56	6.62	0.1	4.51	64.86
Germany	7.37	1.47	2.55	6.45	61.92
Italy	1.13	6.68	0.33	7.51	62.63
United Kingdom	2.36	5.29	2.75	2.24	73.29
Canada	3.42	1.84	2.12	1.47	54.54
USA	7.71	1.08	5.27	8.11	73.05
China	6.1	5.29	9.12	7.22	86.21
India	4.67	9.24	0.37	4.8	82.24
Japan	3.58	2.13	7.49	5.59	67.03

	2005	2006	2007	2008	Totals
Denmark	7.49	7.1	3.77	7.38	85.46
France	5.89	1.18	0.55	8.79	64.86
Germany	2.96	4.15	4.92	1.15	61.92
Italy	2.64	2.59	1.01	2.63	62.63
United Kingdom	4.4	9.92	2.7	2.44	73.29
Canada	4.55	4.7	5.14	0.8	54.54
USA	0.93	1.05	5.72	4.9	73.05
China	8.55	6.74	3.61	0.46	86.21
India	8.25	2.17	1.92	7.76	82.24
Japan	0.34	7.56	3.95	3.36	67.03

4 Frames Across

Caption

	2009	2010	2011	2012	Totals
Denmark	5.18	1.94	8.78	4.94	85.46
France	0.42	8.97	5.9	6.96	64.86
Germany	7.63	1.26	1.89	5.76	61.92
Italy	6.48	1.71	5.24	3.39	62.63
United Kingdom	5.47	8.69	0.46	4.69	73.29
Canada	3.36	5.62	4.44	0.64	54.54
USA	9.92	0.29	3.27	3.14	73.05
China	1.79	9.31	7.76	4.12	86.21
India	7.42	5.25	7.17	3.05	82.24
Japan	9.32	1.63	0.04	8.4	67.03

	2013	2014	2015	Totals
Denmark	1.26	7.15	7.11	85.46
France	1.66	4.27	5.48	64.86
Germany	5.7	4.16	4.5	61.92
Italy	5.99	9.84	5.46	62.63
United Kingdom	5.68	8.63	7.57	73.29
Canada	7.75	4.35	4.34	54.54
USA	9.2	9.9	2.56	73.05
China	8.58	1.95	5.61	86.21
India	8.94	1.4	9.83	82.24
Japan	2.75	9.42	1.47	67.03

Footnote



# SharpLeaf

## Flowing automated reports to paper

### SharpLeaf Unleashed

**S**harpLeaf is page-based reporting tool providing basic typesetting abilities. It is an object-oriented state machine that allows you to programmatically flow content (text, tables, charts and images), page by page. The philosophy, as opposed to most modern reporting tools, is to keep it simple and clean, rather than complex and dirty.

The goal of this report is to demonstrate some basic features of the SharpLeaf engine, which is why this foreword starts by showing off a dropped capital, and how the text flow nicely wraps around it! We will look at other inclusions later, namely charts and tables.

You will find contact information in the "Contact" section (p. 4) .

#### Flowing text

Here are extracts of the Wikipedia article on typography. In the first section, it will be flown as text, with fine control over the "print head". The second section will just flow regular paragraphs. The third, and last, will demonstrate bullet lists.

##### Overview

Typography is the art and technique of arranging type to make written language readable and appealing. The arrangement of type involves selecting **typefaces**, **point size**, line length, *[censored content]*. Type design is a closely related craft, sometimes considered part of typography: most typographers do not design typefaces, and some type designers do not consider themselves typographers.<sup>[3]</sup> In modern times, typography has been put in film, television and online broadcasts to add emotion to communication.<sup>[4]<sup>[unreliable source?]</sup></sup>

Typography is performed by typesetters, compositors, typographers, graphic designers, art directors, manga artists, comic book artists, graffiti artists, clerical workers, and everyone else who arranges type for a product. Until the Digital Age, typography was a specialized occupation. Digitization

opened up typography to new generations of visual designers and lay users, and David Jury, Head of Graphic Design at Colchester Institute in England, states that "typography is now something everybody does."<sup>[1]</sup>

##### Basic principles

Legibility is primarily the concern of the typeface designer, to ensure that each individual character or glyph is unambiguous and distinguishable from all other characters in the font. Legibility is also in part the concern of the typographer to select a typeface with appropriate clarity of design for the intended use at the intended size. An example of a well-known design, Brush Script, contains a number of illegible letters, since many of the characters can be easily misread especially if seen out of textual context.

Readability is primarily the concern of the typographer or information designer. It is the intended result of the complete process of presentation of textual material in order to communicate meaning as unambiguously as possible. A reader should be assisted in navigating around the information with ease, by optimal inter-letter, inter-word and particularly inter-line spacing, coupled with appropriate line length and position on the page, careful editorial "chinking" and choice of the text architecture of titles, folios, and reference links.

Some commonly agreed findings of legibility research include.<sup>[citation needed]</sup>

1. Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
2. Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).
3. Regular upright type (roman type) is found to be more legible than italic type.
4. Contrast, without dazzling brightness, has also been found to be important, with black on yellow/cream being most effective.
5. Positive images (e.g. black on white) are easier to read than negative or reversed (e.g. white on black). However even this commonly accepted practice has some exceptions, for example in some cases of disability.
6. The upper portions of letters play a stronger part than the lower portions in the recognition process.