



ReceiptLine

Receipt Description Languages

Version 1.0

2020年7月13日 - Release

OFSC 店舗システム分科会
Printer WG

免責事項

本書および本書に記載されている情報は、現状有姿で提供されており、OFSCは、明示的又は黙示的な保証を一切否認します。特定の目的のために商品性または適合性を保証するものではありません。

OFSCは、本書または本書を参照、引用、またはリンクされている本書または他の文書の誤植または漏れ落ちについて責任を負うものではありません。

本書には、技術的またはその他の不正確さ、又は誤植が含まれている可能性があります。

OFSCは本書の情報を改善及び／又は変更する権利を留保します。

Disclaimer

This document and the information contained herein is provided on an "As Is" basis and OFSC disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

OFSC assume no responsibility for errors or omission sin this publication or other documents which are referenced by, cited by, or linked to this publication.

This publication could include the technical or other inaccuracies or typographical errors.

OFSC reserve the right to make improvements and/or changes to the information herein.

商標

Unicode は、米国 Unicode, Inc.の登録商標です。

QR コード/QR Code は、株式会社デンソーウェーブの登録商標です。

ReceiptLine は、一般社団法人オープン・フードサービス・システム・コンソーシアムの登録商標です。

名簿

議長

中村 英雄	セイコーエプソン株式会社
-------	--------------

メンバー

湯澤 一比古	株式会社 広告社ぶれいん
樋口 悟	シチズン・システムズ株式会社
今福 貴志	株式会社ジャストプランニング
角野 宏司	株式会社ジャストプランニング
藤原 博文	スター精密株式会社
大場 佳男	セイコーインスツル株式会社
岡安 孝昇	セイコーインスツル株式会社
古幡 整	セイコーエプソン株式会社
梅津 紀之	富士通アイソテック株式会社
三浦 愛弓	富士通アイソテック株式会社
村井 芸典	プットメニュー株式会社

事務局

酒美 保夫	一般社団法人オープン・フードサービス・システム・コンソーシアム
-------	---------------------------------

(社名五十音順)

本書に関するお問い合わせは OFSC 事務局で受け付けております。

目次

1.	はじめに.....	7
1.1	本書について.....	7
1.2	文字集合.....	7
1.3	文字符号化方式.....	7
1.4	用紙.....	7
1.5	データ構造.....	8
2.	構文.....	12
2.1	表記法.....	12
2.2	文書.....	12
2.3	行.....	13
2.4	列.....	13
2.5	テキスト.....	14
2.6	文字.....	15
2.7	エスケープ文字.....	15
2.8	空白.....	16
2.9	プロパティ.....	17
2.10	キーと値.....	18
3.	特殊文字とエスケープ.....	19
3.1	特殊文字.....	19
3.2	エスケープシーケンス.....	19
4.	行.....	21
4.1	行と列.....	21
4.2	行の桁数.....	23
4.3	行の位置揃え.....	23
5.	列.....	25
5.1	列の桁数.....	25
5.2	列の境界.....	29
5.3	列の位置揃え.....	31
6.	テキスト.....	32
6.1	文字列の折り返し.....	32
6.2	文字装飾.....	34
6.3	文書の区切り.....	36
7.	プロパティ.....	37
7.1	プロパティの種類.....	37
7.2	バーコード・二次元コードの種類.....	39

1. はじめに

1.1 本書について

本書では、レシート記述言語 ReceiptLine の仕様を説明する。

ReceiptLine は、小型ロール紙の出力イメージを表現するレシート記述言語である。レシートプリンターによる紙レシートの印刷と、POS システムやスマートフォンによる電子レシートの画面表示に対応し、レシートをはじめ、外食伝票や取引伝票等のイメージを、用紙幅に依存しないテキストデータで簡潔に記述することができる。

1.2 文字集合

ReceiptLine の文字集合(character set)は任意である。

本書では Unicode を想定している。文字幅に関する特性は附属書を参照する。

- The Unicode Consortium
<https://unicode.org/>
- Unicode Standard Annex #11 EAST ASIAN WIDTH
<https://www.unicode.org/reports/tr11/>

1.3 文字符号化方式

ReceiptLine の文字符号化方式(character encoding scheme)は任意だが、システム間で情報をやり取りする場合、UTF-8 を使用すべきである。

異なる文字集合や文字符号化方式を利用する場合、変換に注意する。特に、日本語における円記号は、Shift_JIS では「¥」(0x5C)で、ReceiptLine のエスケープシーケンスにより「¥¥」(0x5C, 0x5C)と記述するが、UTF-8 では Unicode の円記号「¥」(U+00A5 YEN SIGN)を用いる。

1.4 用紙

ReceiptLine のイメージを出力する紙や画面の媒体を用紙(paper)と呼ぶ。また、用紙の横方向に印字できる文字数を印字桁数(characters per line)と呼ぶ。

印字桁数は、文字幅が Narrow または Halfwidth の文字を基準とする。文字幅が Wide または Fullwidth の文字は、1 文字で印字桁数 2 桁分になる。

用紙は出力イメージの終わりでカットする。また、出力イメージの途中で手動カットすることができる。

1.5 データ構造

ReceiptLine 文書はテキストデータである。文書は行の集合で、行の区切り文字は改行である。行は列の集合で、列の区切り文字は縦線「|」である。列はテキストまたはプロパティを持つ。プロパティはキーと値のペアの集合で、中括弧「{」「}」で囲む。

仕様の説明を補助するため、ReceiptLine 文書と出力イメージの例を示す。点線は行と列の境界である。

- ReceiptLine 文書の例

```
{image:iVBORw0KGgoAAAANSUhEUgAAAIAAA
AwAQMAAADj0uD9AAAAB1BMVEUAAAD///+12Z/
dAAAAZk1EQVQoz2P4jwYYRrrABwYG0wYG5gMM
DBUMDPxAgQcMDDJAgQYGHgJcAv//yMj//9/8/
/+HerAZRAsAzUASAJoGMhRF4AC6ANCIhQz8A
kAXQoUOIdiDBQBkG8hAj8gAqPJAA8AAGju1h0
sX97yAAAAAE1FTkSuQmCC}
      市ヶ谷駅前店
      東京都千代田区九段1-Y-X

      2019年 2月19日(火) 19:00
{border:line}
      ^領 収 証
{border:space}
{width:*, 2, 10}
ビール                | 2 |    ¥1,300
千鳥コース            | 2 |   ¥17,280
-----
{width:*, 20}
^合計                  |    ^¥18,580
現 金                  |    ¥20,000
お 釣 り              |    ¥1,420
```


● 出カイメージの例

OFSC		
市ヶ谷駅前店		
東京都千代田区九段 1-Y-X		
2019年 2月 19日(火) 19:00		
領 収 証		
ビール	2	¥1,300
千鳥コース	2	¥17,280
<hr/>		
合計		¥18,580
現金		¥20,000
お釣り		¥1,420

● ReceiptLine 文書の例

~~~新規~~~ {image:iVBORw0KGgoAAAANSUgAAAIAAAA BAAQMAAADoG008AAAAB1BMVEUAAAD///+12Z/ dAAAAaU1EQVQ4y2P4jwYYRgUGh8AHBgY7Bgbm AwwMFQwM/ECBBwwMMkCBBgaGAlwC///IyP//3 /z//4d6sBlECwDNQBIAmgYyFEXgALoA0IgCFD PwCQBdChQ4g0J0FAGQbyECPyACo8lhSgAACg A1SKnht6nAAAAAE1FTkSuQmCC}			
テーブル	A05	伝票	0003-01
時刻	19:00	人数	2
{width:3,*,4,9; border:line}			
-----			
	品名	数量	金額
-----			
□	ビール	2	1,300
□	千鳥コース	2	17,280
-----			
{width:10,18; align:right}			
	小計		18,580
	サービス料		0
-----			
	合計		18,580
-----			
{width:*; border:none; align:center}			
{code:0003; option:code39,48}			

## ● 出カイメージの例

				新規
<b>OFSC</b>				
テーブル	A05	伝 票	0003-01	
時 刻	19:00	人 数	2	
	品名	数量	金額	
<input type="checkbox"/>	ビール	2	1,300	
<input type="checkbox"/>	千鳥コース	2	17,280	
小 計			18,580	
サービス料			0	
<b>合計</b>			<b>18,580</b>	
				

## 2. 構文

---

### 2.1 表記法

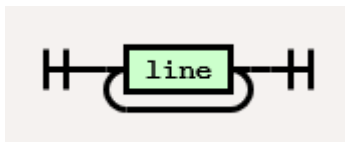
本書では「JSON の紹介」で採用されている簡易バックス・ナウア記法と構文図のスタイルを用いて説明する。

- JSON の紹介  
<https://www.json.org/>
- バックス・ナウア記法(BNF, Backs-Naur Form)  
[https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form](https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form)
- 構文図(Syntax diagram, Railroad diagram)  
[https://en.wikipedia.org/wiki/Syntax_diagram](https://en.wikipedia.org/wiki/Syntax_diagram)

### 2.2 文書

文書(document)は、ひとつの出カイメージを表現する情報のかたまりである。  
文書は、行(line)の集合で構成する。

#### ***document***



```
document
  lines

lines
  line
  line lines
```

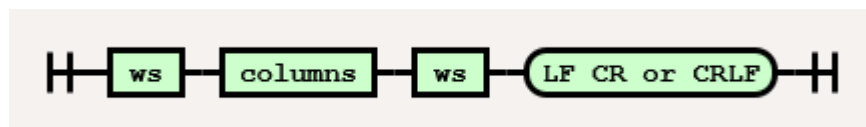
## 2.3 行

行(line)は、文書(document)を構成する横一列の情報である。

行は、列の集合(columns)を持ち、改行(newline)を終端とする。可読性を高めるため、列の集合の前後に空白(ws)を挿入することができる。

改行文字は、LF(U+000A LINE FEED), CR(U+000D CARIDGE RETURN), または、CRLF(U+000D CARIDGE RETURN, U+000A LINE FEED)である。

### *line*



```

line
  ws columns ws newline

newline
  '000a'
  '000d'
  '000d' '000a'

```

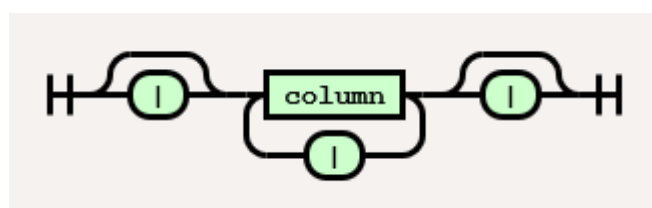
## 2.4 列

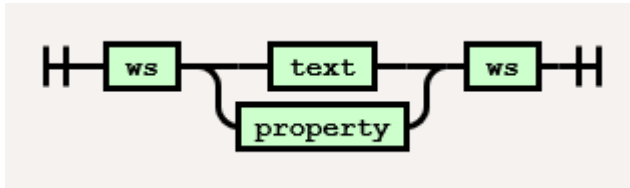
列(column)は、列の集合(columns)を構成する情報である。

列の区切り記号は、縦線「|」(U+007C VERTICAL LINE)である。列の集合の先頭と末尾の縦線は省略することができる。

列は、テキスト(text)またはプロパティ(property)を持つ。位置揃えを指定するため、テキストやプロパティの前後に空白(ws)を挿入することができる。

### *columns*



**column**

```
columns
  content
  content '|'
  '|' content
  '|' content '|'

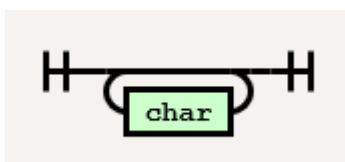
content
  column
  column '|' content

column
  ws text ws
  ws property ws
```

**2.5 テキスト**

テキスト(text)は、列(column)の情報のひとつである。

テキストは、文字(char)の集合で、空文字列を含む文字列である。

**text**

```
text
  ""
  chars

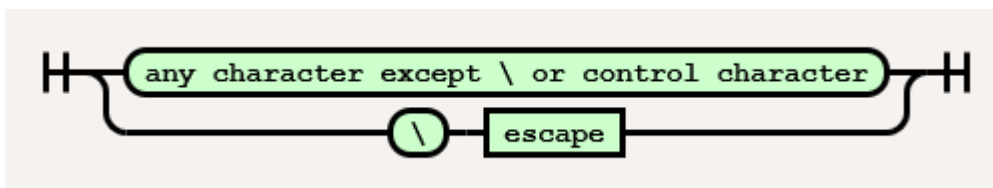
chars
  char
  char chars
```

## 2.6 文字

文字(char)は、テキスト(text)を構成する情報である。

文字は、制御文字を除く任意の1文字か、バックスラッシュ「¥」(U+005C REVERSE SOLIDUS)とエスケープ文字(escape)で構成するエスケープシーケンスである。

### **char**



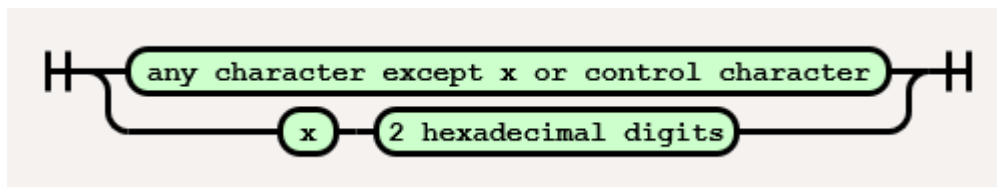
```
char
  '0020' . '10ffff' - '¥'
  '¥' escape
```

## 2.7 エスケープ文字

エスケープ文字(escape)は、エスケープシーケンスを構成する文字である。

エスケープ文字は、特別な意味を割り当てた文字を出力する。エスケープ文字が「x」で始まる場合は2桁の16進数を続けて指定する。

### **escape**



```
escape
    '0020' . '10ffff' - 'x'
    'x' hex hex

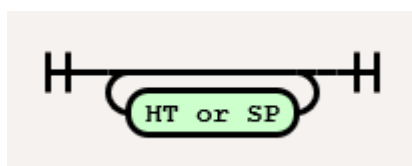
hex
    '0' . '9'
    'A' . 'F'
    'a' . 'f'
```

## 2.8 空白

空白(ws, whitespace)は、可読性を高め、列の位置揃えを指定するために挿入する、出力しない文字である。

空白は、1文字以上の水平タブ(U+0009 HORIZONTAL TABULATION)またはスペース(U+0020 SPACE)の集合で、空文字列を含む文字列である。

### **WS**



```
WS
    ""
    '0009' ws
    '0020' ws
```



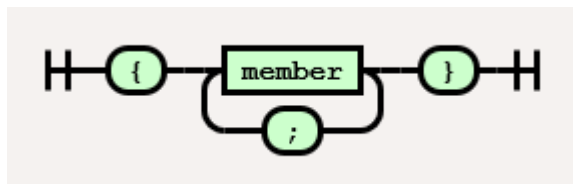
## 2.9 プロパティ

プロパティ(property)は、列(column)の情報のひとつである。

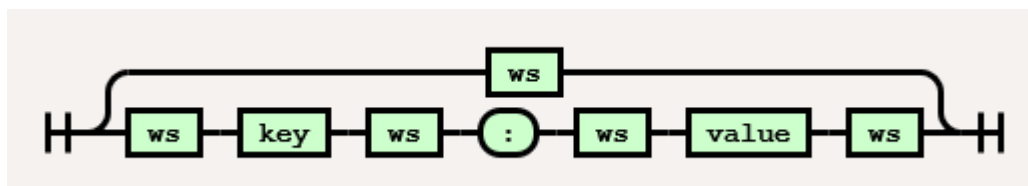
プロパティは、左中括弧「{」(U+007B LEFT CURLY BRACKET)で始まり、メンバー(member)を持ち、右中括弧「}」(U+007D RIGHT CURLY BRACKET)で終わる。複数のメンバーを持つ場合、メンバーをセミコロン「;」(U+003B SEMICOLON)で区切る。

メンバー(member)は、キー(key)と値(value)のペアを持つことができる。キーと値はコロン「:」(U+003A COLON)で区切る。可読性を高めるため、空白(ws)を挿入することができる。

### **property**



### **member**



```

property
  '{' members '}'

members
  member
  member ';' members

member
  ws
  ws key ws ':' ws value ws

```

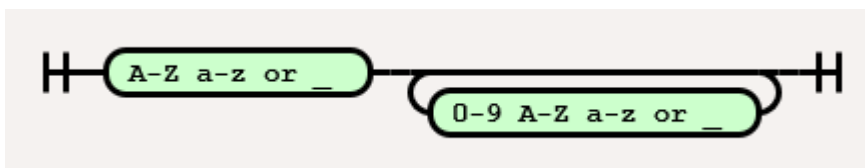
## 2.10 キーと値

キー(key)は、プロパティのメンバーを識別する情報である。  
キーとして使える文字は、数字、英字、アンダースコア「_」(U+005F LOW LINE)である。ただし、先頭の文字は英字かアンダースコアである。

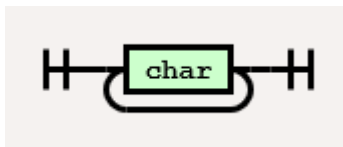
プロパティのキーが重複した場合、後に記述した値を優先する。未知のプロパティは無視する。

値(value)は、空ではない文字列である。

### **key**



### **value**



```
key
  word
  word alphanumeric

alphanumeric
  '0' . '9'
  word

word
  'A' . 'Z'
  'a' . 'z'
  '_'

value
  chars
```

### 3. 特殊文字とエスケープ

#### 3.1 特殊文字

特殊文字は文書で特別な意味を持つ記号である。レシートにおいて使用頻度が低い文字を特殊文字に割り当てている。

共通	説明
¥	文字エスケープ
	列の区切り
{	プロパティの開始
}	プロパティの終了

固有(テキスト)	説明
- (1 個以上, 他の文字なし)	水平線 ・列が 1 個の行で有効
= (1 個以上, 他の文字なし)	用紙カット ・列が 1 個の行で有効
~	空白 ・テキストの先頭や末尾で空白を出力する場合
_	下線
"	強調
`	白黒反転
^	倍角

固有(プロパティの値)	説明
:	キーと値の区切り
;	キー・値ペアの区切り

#### 3.2 エスケープシーケンス

文書に特殊文字や制御文字を埋め込む場合、文字エスケープ「¥」で始まるエスケープシーケンスを使用する。

「¥」の後に正しく文字が続かない不正なエスケープシーケンスは無視する。例え

ば、16進数ではない「¥xnn」や、改行直前の「¥」「¥x」「¥xn」は無効である。

共通	説明
¥¥	「¥」 (U+005C REVERSE SOLIDUS)
¥	「 」
¥{	「{」
¥}	「}」
¥xnn	16進数文字コード指定 ・「nn」は大文字小文字を区別しない

固有(テキスト)	説明
¥-	「-」 (水平線の解除)
¥=	「=」 (用紙カットの解除)
¥~	「~」
¥_	「_」
¥"	「"」
¥`	「`」
¥^	「^」
¥n	文字列の折り返し
¥char (上記以外の文字)	無視する

固有(プロパティの値)	説明
¥;	「;」
¥n	「LF」 (U+000A LINE FEED)
¥char (上記以外の文字)	無視する

## 4. 行

### 4.1 行と列

行は、列の区切り文字「|」を使用して1個から複数個の列を持つことができる。

行	説明
<i>column</i>   <i>column</i>     <i>column</i> <i>column</i>	列が1個の行 ・列の内容はテキストかプロパティ
<i>column</i>   <i>column</i>   <i>column</i>   <i>column</i>     <i>column</i>   <i>column</i> <i>column</i>   <i>column</i>	列が2個の行 ・列の内容はテキスト ・プロパティを含む行を無視する
<i>column</i>   ...   <i>column</i>   <i>column</i>   ...   <i>column</i>     <i>column</i>   ...   <i>column</i> <i>column</i>   ...   <i>column</i>	列がn個の行 ・列の内容はテキスト ・プロパティを含む行を無視する

- ReceiptLine 文書の例

```
|クレジットカード売上票
{width:10 *}
金額 | ^¥12,345
会員番号 |XXXXXXXXXXXX7890
お取扱日 |2019年02月22日
承認番号 |0123456
{width:auto}
お客様控え|
```

- 出カイメージの例

クレジットカード売上票	
金額	¥12,345
会員番号	XXXXXXXXXXXX7890
お取扱日	2019年02月22日
承認番号	0123456
お客様控え	

- ReceiptLine 文書の例

```

{width:21,10}
|お取扱日           |伝票番号           |
|2019年 2月19日    |           12345   |
{width:10,10,10}
|商品区分  |取引内容|取扱区分  |
|   0123456| お買上 |           123|
|^処理通番 |        | ^[一括払い]|
|       7890|金額   |           ¥1,234|
|承認番号  |税送料 |           ¥0   |
|^0001111|^合計  |           ^¥1,234|

```

- 出力イメージの例

お取扱日		伝票番号
2019年 2月 19日		12345
商品区分	取引内容	取扱区分
0123456	お買上	123
処理通番		[一括払い]
7890	金額	¥1,234
承認番号	税送料	¥0
0001111	合計	¥1,234

## 4.2 行の桁数

行の桁数は、`width` プロパティで設定する各列の桁数と、`border` プロパティで設定する列の境界の桁数の合計である。この合計が印字桁数を超える場合や、桁数を自動設定する列を含む場合は、各列の桁数を調整して印字桁数に合わせる。

行の桁数は、画像、バーコード・二次元コード、水平線、コマンドに作用する。テキストは列の桁数を適用する。

- ReceiptLine 文書の例

```
{width: 12 12; border: none}
{image: iVBORw0KGgoAAAANSUgAAAIAAAA
AwAQMAAADj0uD9AAAAB1BMVEUAAAD///+12Z/
dAAAAZk1EQVQoz2P4jwYYRrrABwYG0wYG5gMM
DBUMDPxAgQcMDDJAgQYGHgJcAv//yMj//9/8/
/+HerAZRAsAzUASAJoGMhRF4AC6ANCIahQz8A
kAXQoUOIdiBQBkG8hAj8gAqPJAA8AAGjulh0
sX97yAAAAAE1FTkSuQmCC} |
|合 計 | 1,234|
-
| {code: 0003; option: 48}
```

- 出カイメージの例



## 4.3 行の位置揃え

行の位置揃えは、`align` プロパティで設定することができる。

行の桁数が印字桁数に満たない場合、行を配置する位置を指定する。

行の位置揃え	説明
left	左揃え
center (既定値)	中央揃え
right	右揃え

- ReceiptLine 文書の例

```
{width: 25}
{align: left}
|お客様
{border: line}
^^おすすめ料理は何ですか?
{border: space}
11:59|
{align: right}
ご案内係|
{border: line}
^^中おち丼とあさりのみそ汁です
{border: space}
|12:00
```

- 出力イメージの例

お客様	
おすすめ料理は何ですか?	
11:59	
ご案内係	
中おち丼とあさりのみそ汁 です	
12:00	



## 5. 列

### 5.1 列の桁数

列の桁数は、`width` プロパティで設定することができる。

列の桁数はテキストに作用する。画像、バーコード・二次元コード、水平線、コマンドは、行の桁数を適用する。

列の桁数	説明
桁数, 桁数, ...	出力する列と桁数を指定する
auto (既定値)	すべての列で「*」を指定する

桁数を指定する場合、各列の桁数は「,」または 1 個以上の空白で区切る。桁数の値には以下の種類がある。桁数の個数とその値から出力する列を決定する。

桁数の値	説明
*	桁数を自動設定して出力する
1 以上の整数	桁数を指定して出力する
0 または省略	出力しない

桁数を自動設定する場合、印字可能桁数を各列に均等配分する。

印字可能桁数とは、印字桁数から `border` プロパティで設定する列の境界の桁数を差し引いた桁数である。

- ReceiptLine 文書の例

```
{width:auto}
塩ゆで落花生 | 1 | 500
田無キャベツ塩だれ | 1 | 500
信州レタスのサラダ | 1 | 500
石垣苺ロール | 1 | 500
伊達の桃ジュース | 1 | 500
```

- 出力イメージの例

塩ゆで落花生	1	500
田無キャベツ 塩だれ	1	500
信州レタスの サラダ	1	500
石垣苺ロール	1	500
伊達の桃ジュ ース	1	500

桁数を自動設定する列と、桁数を指定する列が混在する場合、桁数を自動設定する列の桁数は、印字可能桁数から桁数を指定する列の桁数を差し引いて均等配分する。

- ReceiptLine 文書の例

```
{width:*, 2, *}
塩ゆで落花生 | 1 | 500
田無キャベツ塩だれ | 1 | 500
信州レタスのサラダ | 1 | 500
石垣苺ロール | 1 | 500
伊達の桃ジュース | 1 | 500
```

- 出力イメージの例

塩ゆで落花生	1	500
田無キャベツ塩 だれ	1	500
信州レタスのサ ラダ	1	500
石垣苺ロール	1	500
伊達の桃ジュ ース	1	500

桁数を指定する列の桁数の合計が印字可能桁数を超える場合、桁数を削減して印字

可能桁数に合わせる。削減後の最小桁数は1桁とする。

- ReceiptLine 文書の例

```
{width:100, 2, 10}
塩ゆで落花生 | 1 | 500
田無キャベツ塩だれ | 1 | 500
信州レタスのサラダ | 1 | 500
石垣苺ロール | 1 | 500
伊達の桃ジュース | 1 | 500
```

- 出力イメージの例

塩ゆで落花生	1	500
田無キャベツ塩だれ	1	500
信州レタスのサラダ	1	500
石垣苺ロール	1	500
伊達の桃ジュース	1	500

桁数が0桁の列は出力しない。すべての列の桁数が0桁の場合、空行になる。

- ReceiptLine 文書の例

```
{width:*, 0, 10}
塩ゆで落花生 | 1 | 500
田無キャベツ塩だれ | 1 | 500
信州レタスのサラダ | 1 | 500
石垣苺ロール | 1 | 500
伊達の桃ジュース | 1 | 500
```

- 出力イメージの例

塩ゆで落花生	500
田無キャベツ塩だれ	500
信州レタスのサラダ	500
石垣苺ロール	500
伊達の桃ジュース	500

桁数を指定した列が存在しない場合、空の列を補充する。

- ReceiptLine 文書の例

```
{width:*, , 2, 10}
塩ゆで落花生      | 500| 2| 1,000|外
田無キャベツ塩だれ | 500| 1|   500
信州レタスのサラダ | 500| 2
石垣苺ロール      | 500
伊達の桃ジュース  |
```

- 出力イメージの例

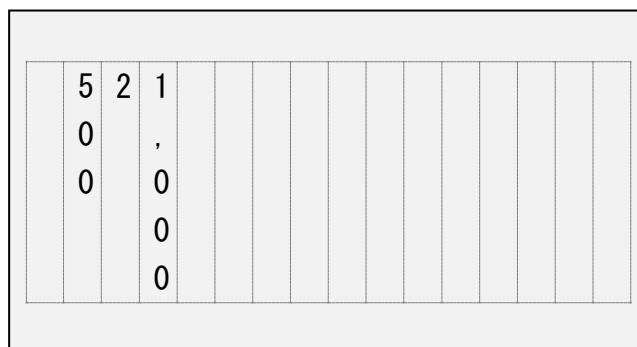
塩ゆで落花生	2	1,000
田無キャベツ塩だれ	1	500
信州レタスのサラダ	2	
石垣苺ロール		
伊達の桃ジュース		

列の個数が印字可能桁数を超える場合、超過する列の桁数を 0 桁にする。  
列の桁数を超える文字幅を持つ倍角文字や漢字は出力しない。

- ReceiptLine 文書の例

```
{width:*, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, 1, 2, 3, 4,
5, 6, 7, 8, 9, *, 1, 2, 3, 4, 5, 6, 7, 8, 9}
塩ゆで落花生      | 500| 2| 1,000|外
```

● 出力イメージの例



## 5.2 列の境界

列の境界は、border プロパティで設定することができる。

列の境界はテキストに作用する。画像、バーコード・二次元コード、水平線には適用しない。

列の境界	説明
0 none	境界なし ・列と列は接する
1 space (既定値)	空白 ・列と列の間隔は1桁
2	幅広の空白 ・列と列の間隔は2桁
line	罫線 ・列と列の間に縦罫線を引く ・列と列の間隔は1桁 ・水平線を横罫線に変換する ・テキストを挟まない水平線は無視する ・有効範囲の外周に枠線を引く ・罫線用に1桁の間隔を空ける ・用紙カット前後と文書の終わりに罫線を引く

## ● ReceiptLine 文書の例

```
{border: space; width: * 4 8}
-
|品名|数量|金額|
-
塩ゆで落花生 | 1| 500
田無キャベツ塩だれ | 1| 500
信州レタスのサラダ | 1| 500
石垣苺ロール | 1| 500
伊達の桃ジュース | 1| 500
-
```

## ● 出力イメージの例

品名	数量	金額
塩ゆで落花生	1	500
田無キャベツ塩だれ	1	500
信州レタスのサラダ	1	500
石垣苺ロール	1	500
伊達の桃ジュース	1	500

## ● ReceiptLine 文書の例

```
{border: line; width: * 4 8}
-
|品名|数量|金額|
-
塩ゆで落花生 | 1| 500
田無キャベツ塩だれ | 1| 500
信州レタスのサラダ | 1| 500
石垣苺ロール | 1| 500
伊達の桃ジュース | 1| 500
-
```

- 出力イメージの例

品名	数量	金額
塩ゆで落花生	1	500
田無キャベツ塩だれ	1	500
信州レタスのサラダ	1	500
石垣苺ロール	1	500
伊達の桃ジュース	1	500

### 5.3 列の位置揃え

列の位置揃えは、列の区切り文字「|」に隣接する 1 文字以上の空白「_」の有無により決定する。列の内容は、縦線に引き寄せられ、空白に突き放される。

列の位置揃えは、テキスト、画像、バーコード・二次元コード、コマンドに作用する。水平線には適用しない。

列の位置揃え	説明
<i>column</i>   <i>column</i>    _ <i>column</i> _	中央揃え
<i>column</i>   <i>column</i> _   <i>column</i> _	左揃え
<i>column</i>    _ <i>column</i>    _ <i>column</i>	右揃え

## 6. テキスト

### 6.1 文字列の折り返し

文字列の折り返しは、`text` プロパティで設定することができる。

文字列の折り返し	説明
<code>wrap</code> (既定値)	文字列を折り返す
<code>nowrap</code>	文字列を折り返さない

文字列を折り返す場合、列の桁数を超える文字列や「`\n`」出現後の文字列は、列の内部で段を作成して出力する。

段の増加により、列の高さが増加し、行の高さが増加する。同一行の各列は段を上揃えで配置する。また、各列の同一段の文字はベースラインを基準に合わせる。

- ReceiptLine 文書の例

```
{width: *,10; border: line}
トマトとオニオンとミートソースにマヨ
ネーズとマスタードが効いたハンバーガ
ー | 390
```

- 出力イメージの例

トマトとオニオンとミート ソースにマヨネーズとマス タードが効いたハンバーガ ー	390
---------------------------------------------------	-----

- ReceiptLine 文書の例



```
{width: *,10; border: line}
トマトと¥nオニオンと¥nミートソースに
¥nマヨネーズと¥nマスタードが効いた¥n
ハンバーガー | 390
```

- 出力イメージの例

トマトと オニオンと ミートソースに マヨネーズと マスタードが効いた ハンバーガー	390
-----------------------------------------------------------	-----

文字列を折り返さない場合、列の桁数を超える文字列や「¥n」出現後の文字列は、出力しない。

- ReceiptLine 文書の例

```
{text: nowrap}
{width: *,10; border: line}
トマトとオニオンとミートソースにマヨ
ネーズとマスタードが効いたハンバーガ
ー | 390
```

- 出力イメージの例

トマトとオニオンとミート	390
--------------	-----

- ReceiptLine 文書の例

```
{text: nowrap}
{width: *,10; border: line}
トマトと¥nオニオンと¥nミートソースに
¥nマヨネーズと¥nマスタードが効いた¥n
ハンバーガー | 390
```

- 出力イメージの例

トマトと	390
------	-----

## 6.2 文字装飾

特殊文字を用いて、テキストの一部の文字列を装飾することができる。

文字装飾の種類は、下線・強調・白黒反転・倍角で、それぞれに開始位置と終了位置を指定して組み合わせることができる。

文字装飾の有効範囲は列である。前の列の文字装飾は次の列に影響しない。文字装飾の開始位置のみ指定して、列の終わりで終了することができる。

同一行に高さの異なる文字が混在する場合、ベースラインを基準に揃える。

出力先が倍角に対応していない場合、文字幅を維持して他の文字装飾表現に置き換えることができる。

以下の表に文字装飾のパターンを示す。装飾する文字を *chars* で示す。

文字装飾	説明
<i>_chars</i>	下線の開始
<i>_chars_</i>	下線の範囲指定
<i>__</i>	空の下線
<i>"chars</i>	強調の開始
<i>"chars"</i>	強調の範囲指定
<i>""</i>	空の強調
<i>`chars</i>	白黒反転の開始
<i>`chars`</i>	白黒反転の範囲指定
<i>``</i>	空の白黒反転
<i>^chars</i>	横倍角の開始
<i>^chars^</i>	横倍角の範囲指定

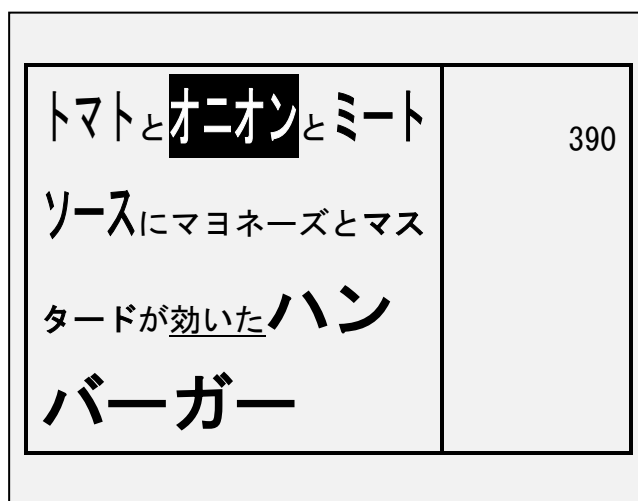
^^chars	縦倍角の開始
^^chars^^	縦倍角の範囲指定
^^^chars	縦横倍角の開始
^^^chars^^^	縦横倍角の範囲指定
^^^^chars	縦横 3 倍角の開始
^^^^chars^^^^	縦横 3 倍角の範囲指定
^^^^^chars	縦横 4 倍角の開始
^^^^^chars^^^^^^	縦横 4 倍角の範囲指定
^^^^^^chars	縦横 5 倍角の開始
^^^^^^chars^^^^^^^	縦横 5 倍角の範囲指定
^^^^^^^^chars (7 個以上)	縦横 6 倍角の開始
^^^^^^^^chars^^^^^^^^^ (7 個以上)	縦横 6 倍角の範囲指定

下線・強調・白黒反転・倍角は、組み合わせて使用することができる。

- ReceiptLine 文書の例

```
{width: *,10; border: line}
^^トマト^^と^^オニオン^^と^^ミート
ソース^^にマヨネーズと"マスタード"が
_効いた_^^"ハンバーガー | 390
```

- 出力イメージの例



### 6.3 文書の区切り

特殊文字を用いて、文書の区切りを付加することができる。  
 文書の区切りの種類は、水平線と用紙カットである。

- ReceiptLine 文書の例

```
{width: *,10; border: line}
=====
^^調理单品`【新規】 | ^^2卓
0001-01 2名 | 22:22
-----
^^グラスビール | ^^1
=====
^^調理单品`【新規】 | ^^2卓
0001-01 2名 | 22:22
-----
^^ビール | ^^1
```

- 出カイメージの例

✂	
<b>調理单品【新規】</b>	<b>2 卓</b>
0001-01 2名	22:22
<b>グラスビール</b>	<b>1</b>
✂	
<b>調理单品【新規】</b>	<b>2 卓</b>
0001-01 2名	22:22
<b>ビール</b>	<b>1</b>

## 7. プロパティ

### 7.1 プロパティの種類

プロパティは、列が1個の行で有効である。プロパティのキーが重複した場合、後に記述した値が有効である。未知のプロパティは無視する。

プロパティ	説明
<pre>{ image: base64data } { i: base64data }</pre>	画像 <ul style="list-style-type: none"> <li>・値は PNG 形式の base64 データ</li> <li>・モノクロ 2 階調, 補助チャンクなしを推奨</li> <li>・ピクセル等倍</li> <li>・同一行の優先順位は image &gt; code &gt; command</li> </ul>
<pre>{ code: textdata } { c: textdata }</pre>	バーコード・二次元コード <ul style="list-style-type: none"> <li>・値はバーコードデータの文字列</li> <li>・同じデータでもパターンが異なる場合がある</li> <li>・クワイエットゾーンなし</li> <li>・同一行の優先順位は image &gt; code &gt; command</li> </ul>
<pre>{ option: textdata } { o: textdata }</pre>	バーコード・二次元コード設定 <ul style="list-style-type: none"> <li>・値は順不同で省略可能</li> <li>・複数の値は「,」または1個以上の空白で区切る</li> <li>・大文字小文字を区別しない</li> <li>・同類の値は先に記述した値が有効</li> <li>・不正な値は無視する</li> <li>・設定は次行以降も有効</li> </ul> ----- <p>バーコード固有の設定</p> <ul style="list-style-type: none"> <li>・値は種類・幅・高さ・可読表示</li> <li>・種類は別途説明</li> <li>・モジュールの幅は「2」～「4」の整数</li> <li>・モジュールの高さは「24」～「240」の整数</li> <li>・可読表示の値は「hri」「nohri」</li> <li>・「hri」は可読表示を行う</li> <li>・値を省略した場合は「2」「72」「nohri」</li> </ul> ----- <p>二次元コード固有の設定</p>

	<ul style="list-style-type: none"> <li>・値は種類・セルサイズ・誤り訂正レベル</li> <li>・種類は別途説明</li> <li>・セルサイズは「3」～「8」の整数</li> <li>・QR Code の誤り訂正レベルは「1」「m」「q」「h」</li> <li>・QR Code で値を省略した場合は「3」「1」</li> </ul>
<pre>{ align: <i>textdata</i> } { a: <i>textdata</i> }</pre>	<p>行の位置揃え設定</p> <ul style="list-style-type: none"> <li>・値は「left」「center」「right」</li> <li>・大文字小文字を区別しない</li> <li>・不正な値は「center」とみなす</li> <li>・既定値は「center」</li> <li>・設定は次行以降も有効</li> </ul>
<pre>{ width: <i>textdata</i> } { w: <i>textdata</i> }</pre>	<p>列の桁数設定</p> <ul style="list-style-type: none"> <li>・値は各列の桁数または「auto」</li> <li>・複数の値は「,」または1個以上の空白で区切る</li> <li>・大文字小文字を区別しない</li> <li>・各列の桁数は「0」以上の整数または「*」</li> <li>・「*」は桁数を自動設定する</li> <li>・値を省略した場合は「0」とする</li> <li>・不正な値は「0」とみなす</li> <li>・「auto」を含む場合、すべての列で「*」とする</li> <li>・既定値は「auto」</li> <li>・設定は次行以降も有効</li> </ul>
<pre>{ border: <i>textdata</i> } { b: <i>textdata</i> }</pre>	<p>列の境界設定</p> <ul style="list-style-type: none"> <li>・値は「line」「space」「none」</li> <li>・桁数で指定する場合「0」～「2」の整数</li> <li>・「space」は「1」と同じ</li> <li>・「none」は「0」と同じ</li> <li>・大文字小文字を区別しない</li> <li>・不正な値は「space」とみなす</li> <li>・既定値は「space」</li> <li>・設定は次行以降も有効</li> </ul>
<pre>{ text: <i>textdata</i> } { t: <i>textdata</i> }</pre>	<p>文字列の折り返し設定</p> <ul style="list-style-type: none"> <li>・値は「wrap」「nowrap」</li> <li>・「nowrap」は文字列を折り返さない</li> <li>・大文字小文字を区別しない</li> <li>・不正な値は「wrap」とみなす</li> <li>・既定値は「wrap」</li> <li>・設定は次行以降も有効</li> </ul>

{ command: <i>textdata</i> }	コマンド
{ x: <i>textdata</i> }	<ul style="list-style-type: none"> <li>・ 互換性を失うため非推奨</li> <li>・ 値はデバイス固有コマンドの文字列</li> <li>・ 同一行の優先順位は image &gt; code &gt; command</li> </ul>
{ comment: <i>textdata</i> }	コメント
{ _: <i>textdata</i> }	<ul style="list-style-type: none"> <li>・ 値は任意の文字列</li> </ul>

## 7.2 バーコード・二次元コードの種類

バーコード・二次元コード設定プロパティで指定できる種類と、バーコード・二次元コードプロパティで有効なデータを示す。

バーコード・二次元コード	説明
upc	UPC <ul style="list-style-type: none"> <li>・ UPC-A は 11～12 桁の数字</li> <li>・ UPC-E は「0」で始まる 7～8 桁の数字</li> <li>・ チェックデジットは省略可能</li> <li>・ チェックデジットの誤りは修正する</li> </ul>
ean jan	EAN(JAN) <ul style="list-style-type: none"> <li>・ EAN-13(JAN-13)は 12～13 桁の数字</li> <li>・ EAN-8(JAN-8)は 7～8 桁の数字</li> <li>・ チェックデジットは省略可能</li> <li>・ チェックデジットの誤りは修正する</li> </ul>
code39	CODE39 <ul style="list-style-type: none"> <li>・ 数字, 英字(大文字), 記号(-. \$+%), スペース</li> </ul>
itf	Interleaved 2 of 5 <ul style="list-style-type: none"> <li>・ 偶数桁の数字</li> </ul>
codabar nw7	Codabar(NW-7) <ul style="list-style-type: none"> <li>・ スタートコード(ABCD)</li> <li>・ 数字, 記号(-. \$:/+)</li> <li>・ ストップコード(ABCD)</li> </ul>
code93	CODE93 <ul style="list-style-type: none"> <li>・ 数字, 英字, 記号, スペース, 制御文字(~¥x7f)</li> </ul>
code128 (既定値)	CODE128 <ul style="list-style-type: none"> <li>・ 数字, 英字, 記号, スペース, 制御文字(~¥x7f)</li> </ul>
qrcode	QR Code <ul style="list-style-type: none"> <li>・ 数字, 英数字, 8 ビットバイト, 漢字</li> </ul>

