

這個程式借Knuth 的排版理論，但是為著beh予列位知影所以mā是佇tsia介紹：

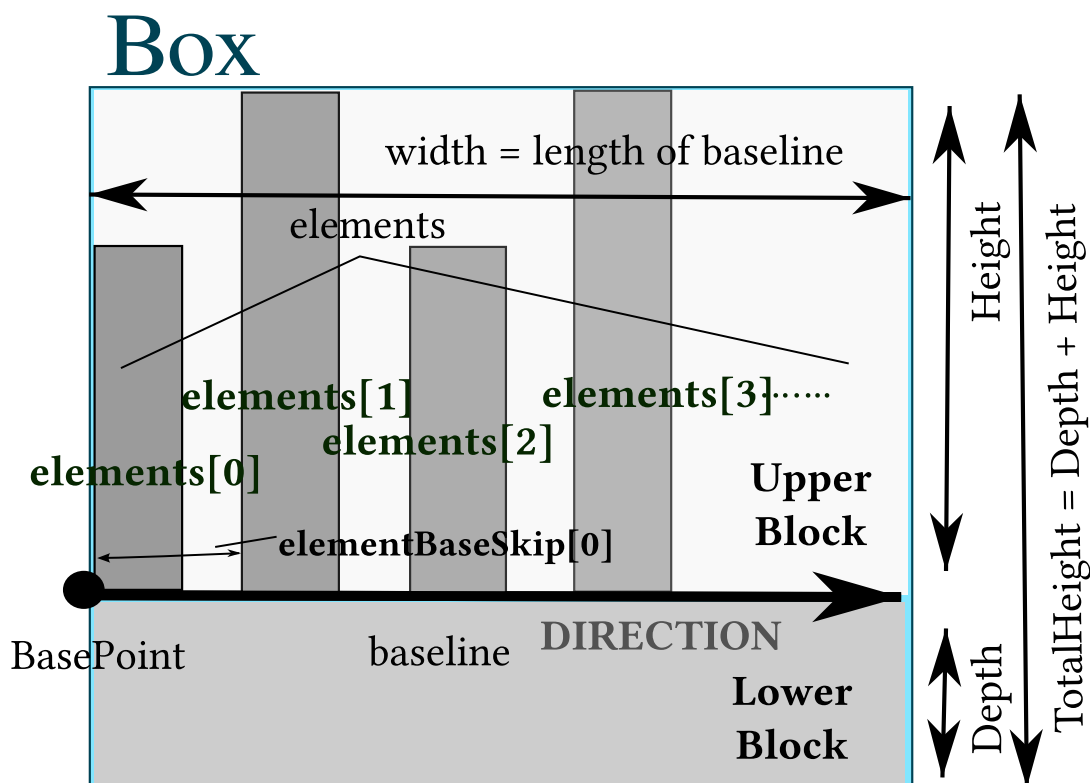


Figure 1: a.svg

2.2 一个盒仔的結構

2.2.1 咱tsit-má來看一个盒仔(Box) 的結構，e當分做若以上e圖講e部件：

Box is a subtype of Element

屬性英語名	型別	臺語解說
basePoint	Position	定義安khng盒仔的基點。
baseLineDirection	Rection	定義基線（安khng元素的基本線）的方向
width	float	基線的長度(pt)
height	float	BaseLine 射出去的方向的「倒ping」的長度 (pt) (m是kui个的懸度 totalHeight)
depth	float	BaseLine 射出去的方向的「正ping」的長度 (pt)
elements	Element List	所有內底收囡的元素列單(list)，照baseLineDirection 的方向，沿 baseLine 排in 的基點
elementsBaseSkip	float List	elementsSpaging[i] == elements[i] 基點之間距離 (BaseSkip) (px)
pageNo	Int	所屬仔的頁數

其中

```
Position = struct {x: Float, y : Float}
Direction = Up | Down | Left | Right
```

2.2.2 相關函數：

英語名	型別	臺語解說
len_of_elements	box -> int	len_of_elements box = length (box.elements)
totalHeight	box -> float	totalHeight box = box.height + box.depth
insert_element	box -> element List -> float -> Option	插入一个element List內底逐一个元素kàu一个Box，逐一个baseSkip是固定float，轉來Result Box á是Exception (that 滿時)。(例見下kha)

```
let a_box = Box{
  basePoint = (70.0,50.0),
  baseLineDirection = DOWN,
  width = 400.0,
  height = 700.0,
  depth = 0.0,
  elements = [],
  elementsBaseSkip = [],
  pageNo = 3
};
```

```
let a_char_list = List.map [天, 地, 儂] fun x -> CharToBox x
  12.0 (*font-size*)
  "AR PL New Sung" (*font-family*)
  300 (*font weight*)
  Normal (*font style*);;
```

```
let new_a_box = insert_element a_box a_char_list 15
(* new_a_box : Option Box *)
```

其中

```
Option = Result a | Exception str
```