

# SBデザインオーバーフロー処理(1/2)

新規作成 2024.03.15  
最終更新 2024.04.13

## ■前提条件

- 各要素のサイズを決定する根源的な条件は、ブラウザの表示領域の大きさと写真のサイズとなる。
- ブラウザ表示領域の全体をbodyで定義する。
- bodyの中に写真を收容するときに、横方向か縦方向のいずれが制限条件となるかが最初に決定される。その制約条件から、写真の表示サイズが決定される。
- 写真を包含する最小の枠組みとしてcontainerのサイズが決定される。
- 以降、全てのボックスはこのcontainerのサイズ(縦サイズy\_c、横サイズx\_c)を前提条件として定義される。

## ■CASE1 : y\_cは左カラム(写真)のheightによって決定される。

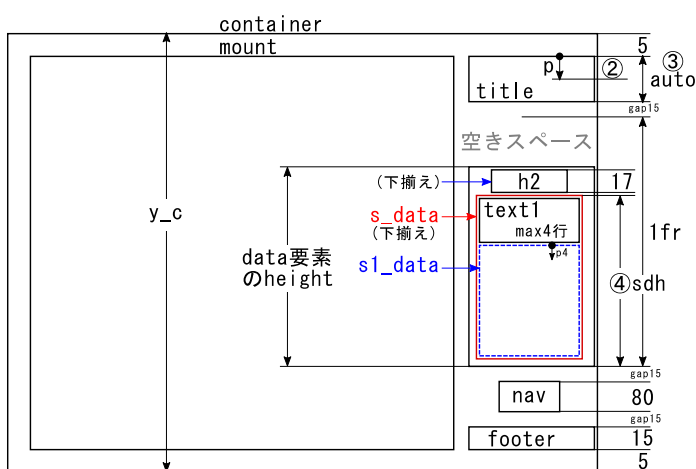


図-1 左カラムがクリティカルとなるケース

## ■data

- dataはgridの子要素で高さは1frであるが、CSSにalign-self:endが指定されているため、dataのheightは、dataの子要素のheightの合計となる。また、dataボックスは1frの中で下揃え配置される。

## ■s\_data

- s\_dataの中にs1\_dataの全体を收容することができる。overflowは発生しない。
- text1はスマホ横画面でも表示可能な行数(max4行程度)を目途に記載する(省略も可)

## ■s1\_data

- s1\_dataには単純に左上から順にテキストが流し込まれる。
- s1\_data要素はオプションで、省略可。
- s1\_dataはスマホ横画面では非表示とする。

## ■CASE2 : y\_cは右カラムの最低高と画面表示域の高さによって決定される。

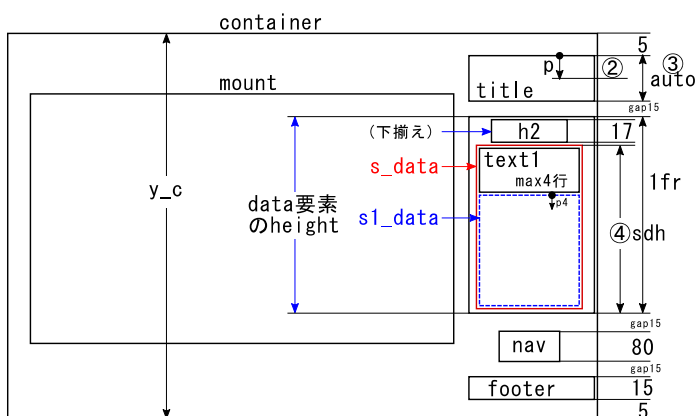


図-2 右カラムがクリティカルとなるケース

## ■data

- 上下方向の圧迫により、dataのheightはy\_cの制限により、取り得る最小値になる。
- dataにalign-self:endを指定して、dataのheight値を発生させる。

## ■s\_data

- s\_dataのheight値はdataのheight値による制限を受ける。そのため子要素の全てを收容しきれず、s\_dataでoverflowを発生させる。
- s\_dataはgrid子要素なのでheightが発生し、overflowの設定ができるが、overflowの処理に際して行の上半分が表示されるようなことを避けるため、s\_dataのheightをtext1要素のline-height刻みの高さとなるように動的に設定する。

## ■s1\_data

- s1\_dataはs\_dataの子要素であり、s\_dataのoverflow設定によって途中までの表示となる。しかし、その場合でもs1\_data要素としての元々のheight値はそのまま維持される。
- s1\_data要素はオプションで、省略可。
- s1\_dataはスマホ横画面では非表示とする。

$$\text{sdh1} = \textcircled{4}\text{sdn} = (\text{s\_data 要素の高さ})$$

$$\text{sdh2} = \textcircled{4}\text{sdn} = (\text{y\_c} - 130 - (\text{y\_c} - \text{vym}) / 2 + 10 + \textcircled{3} + 37)$$

# SBデザインオーバーフロー処理(2/2)

新規作成 2024.03.15  
最終更新 2024.04.13

## ■ s\_dataへのheight設定

- s\_dataに動的に設定するheight値には次の2つのアプローチによってそれぞれ設定値[sdh1]、[sdh2]が用意される。

[sdh1]

- s\_dataのheight値から算出する

$sdh1 = \text{document.getElementById('s\_data').clientHeight} + 4$

[sdh2]

- y\_cのheight値から算出する

$sdh2 = y\_c - \textcircled{3} - 167$

$\textcircled{3} = \text{document.getElementById('title').clientHeight}$

$167 = 5 + 15 + 17 + 15 + 80 + 15 + 15 + 5$

## ■ s\_dataのheight値を決定する要因

- [sdh1]と[sdh2]の2つのheight値のうちどちらを採用するかは、左右2つのカラムについて、どちらがcontainerのheightを決定する要因(クリティカル)になるかによって決まる。

### ■ CASE1 (左カラムクリティカル) 図-1参照

containerのheight(y\_c)が左カラムの写真のheightを制約している場合で、右カラム全体が必要とするheight値がy\_cより小さい場合。

この場合、右カラムで余ったheight値はGridの1frで定義された第2段落(data)に割り当てられる。

従って、dataには有り余るheightがあり、子要素のs\_dataを余裕を持って包含することができる。

s\_dataのheight値にはs\_dataのheightを設定する。

$s\_dataのheight = sdh = \textcircled{4} = s\_dataのheight \dots \dots \dots sdh1$

### ■ CASE2 (右カラムクリティカル) 図-2参照

containerのheight(y\_c)が右カラム全体のheight値を制約している場合で、右カラムのうち、Gridの1frで定義されているdataが縦方向圧縮の対象となる。

このとき、s\_dataが取り得るheight値は、

$s\_dataのheight = sdh = \textcircled{4} = y\_c - \textcircled{3} - 167 \dots \dots \dots sdh2$

となる。

s\_dataの全ての情報を表示できない場合がある。

## ■ どちらのheight値を採用するか判定 (CASEの判定)

- 2つのCASEのうち、どのCASEに該当するかは、[sdh1]の値と[sdh2]の値を比較することによって決定することができる。

[sdh1] < [sdh2]の場合 : CASE2 (y\_cから算出した値を採用)

[sdh1] > [sdh2]の場合 : CASE1 (s\_dataのheightから算出した値を採用)