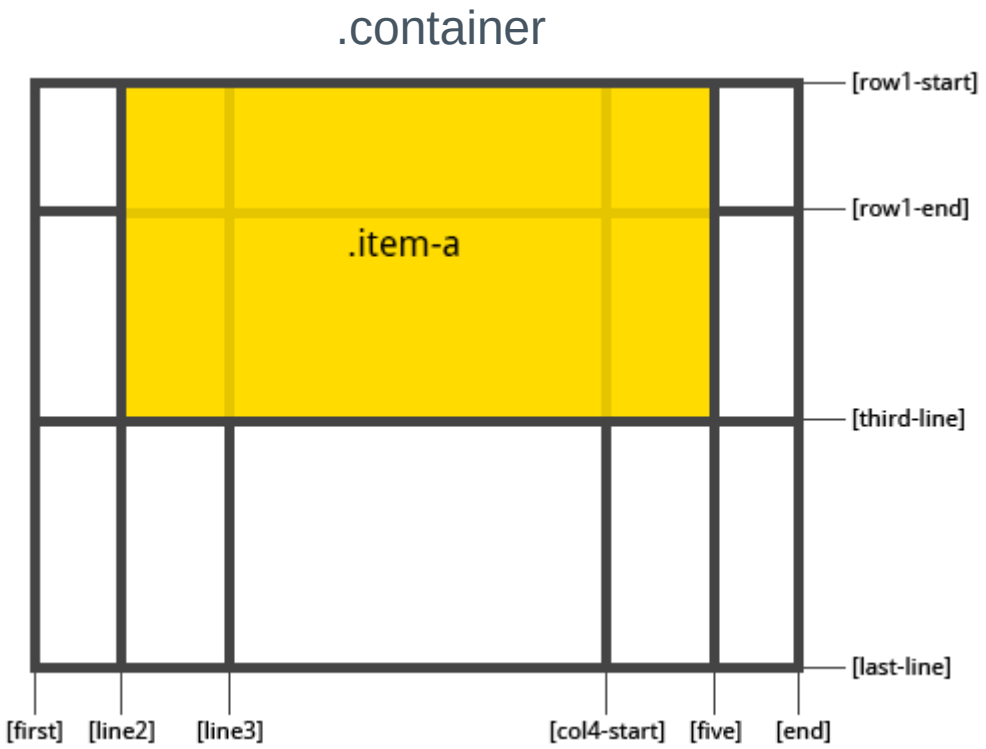
網頁程式設計
CSS樣式設計

Instructor: 馬豪尚

Grid Layout格線版面設計

Grid Layout格線版面

- Grid 和 Flex 最大不同之處，透過 grid template 來定義版型的結構，分別由 column 及 row 定義出直排與橫列的格線，內容再依格線作安排
- display: grid | inline-grid
  - grid : block層級的grid
  - inline gird : inline層級的grid



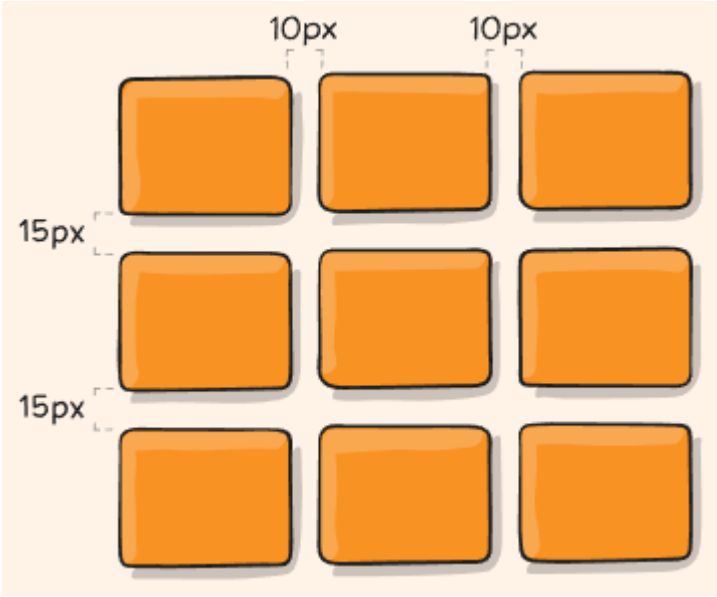
Grid 父元素屬性
(Grid Container)

格線定義

- 定義欄位
  - grid-template-columns: 長度 | 百分比 | auto | 分數 (分數的部分需使用 fr 單位)
    - fr單位: 這個單位能夠將可用的 剩餘空間 做比例分割
    - 1fr 2fr 為例，剩餘空間將被分割成兩個1:2的空間
  - 格線可以自行取名: [line-name]
    - [first] 40px [line2] 50px [line3] auto [col4-start] 50px [five] 40px [end]
  - 重複格線: repeat({次數}, {格線...} | {格線...})
    - repeat(2, [line] 40)

格線定義

- 定義列
  - grid-template-rows:長度 | 百分比 | auto | 分數
  - 也可以取名和使用重複寫法
- 格線間隔
  - 定義格線的寬度
  - column-gap: 長度
  - row-gap: 長度



格線區域定義

- grid-template-areas: <grid-area-name> | . | none
- 透過 area 定義區塊在 template 上的位置，概念就是在畫面上登記屬於該元素的空間

header	header	header	header	header
side	main	main	main	main
side	footer	footer	footer	footer

```
grid-template-areas:
"header header header header header"
"side  main main main main"
"side footer footer footer footer";
```

格線定義樣板速記

- grid-template: <grid-template-rows> || <grid-template-columns> || < grid-template-areas>

樣板速記

```
.container {
  grid-template:
    [row1-start] "header header header" 25px [row1-end]
    [row2-start] "footer footer footer" 25px [row2-end]
    | auto 50px auto;
}
```

分開格線定義

```
.container {
  grid-template-rows: [row1-start] 25px [row1-end row2-start] 25px [row2-end];
  grid-template-columns: auto 50px auto;
  grid-template-areas:
    "header header header"
    "footer footer footer";
}
```

自動分割欄位或列

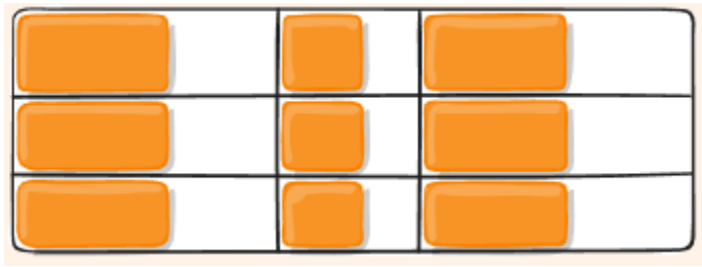
- grid-auto-columns, grid-auto-rows: 長度
  - 會自動依照設定的長度重複分割父元素寬度或高度
  - 適用較簡單的完全制式化版型
  - 可以搭配gap使用
- grid-auto-flow: row | column | row dense | column dense
  - Grid 的排列方式，搭配自動分割欄位或列使用，如果沒有設定會變成垂直排列。

Grid 內容對齊方式

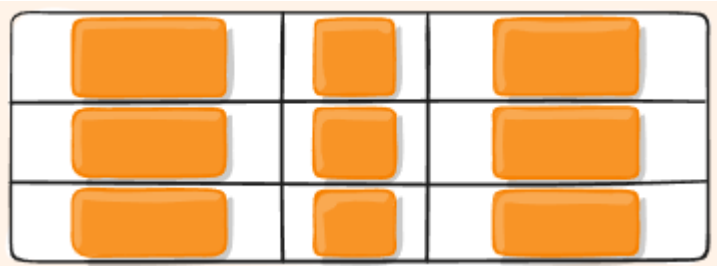
- 套用在父元素格線版面上(一次對齊所有項目)，基本上和flex item的對齊方式相同
- justify-items: start | end | center | stretch;
  - 全部子元素的水平對齊方式
- align-content
  - 全部子元素的垂直對齊方式
- justify-content
  - 子元素多行對齊方式

justify-items

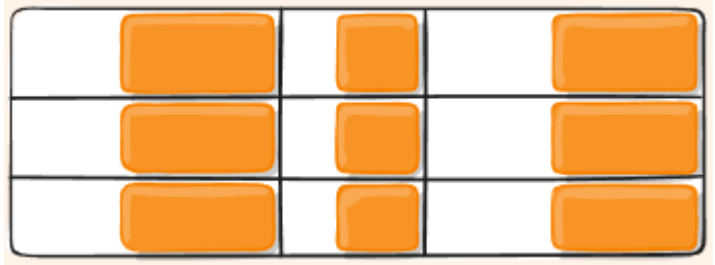
justify-items: start



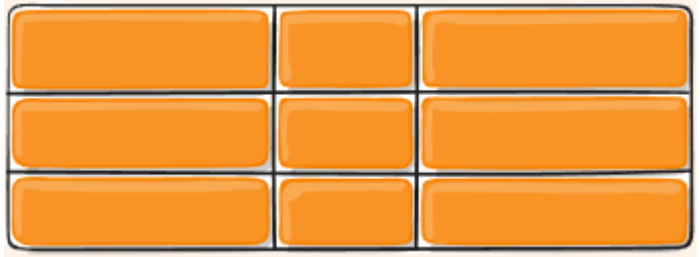
justify-items: center



justify-items: end

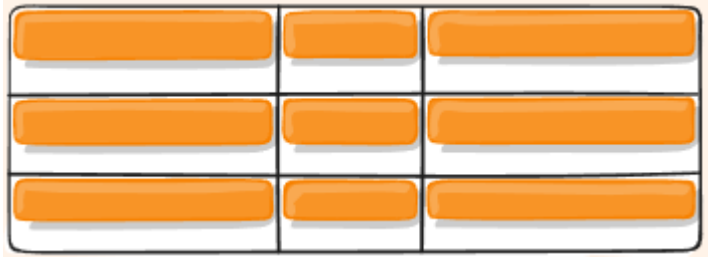


justify-items: stretch

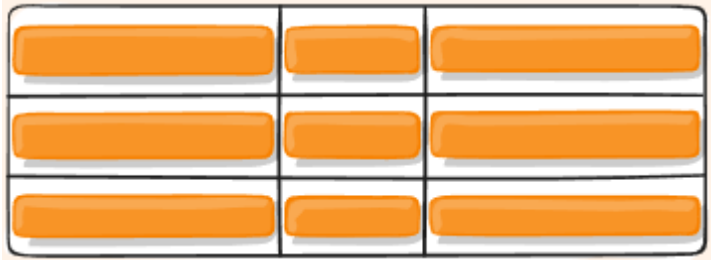


align-content

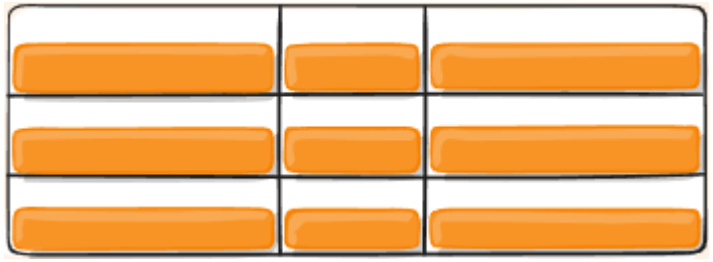
align-items: start



align-items: center



align-items: end



align-items: stretch

Grid 子元素屬性
(Grid Item)

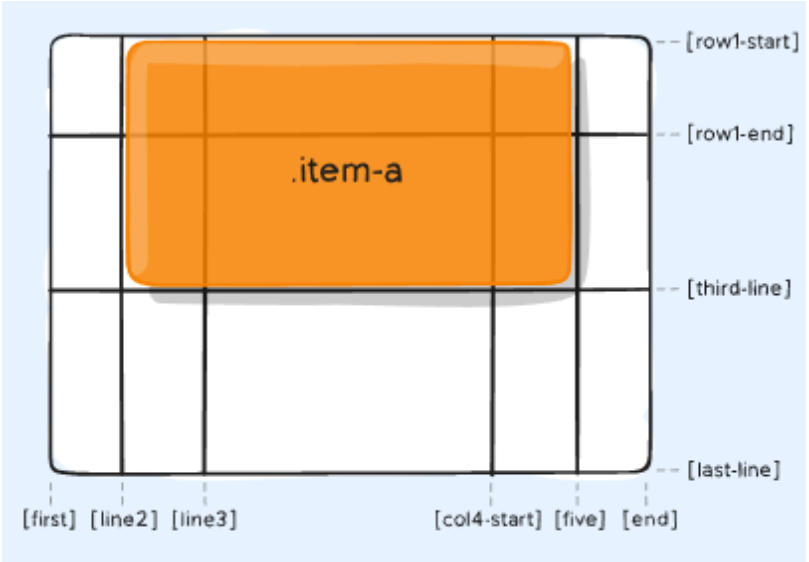
子元素位置的空間定義

- grid-column-start, grid-column-end, grid-row-start, grid-row-end
  - <line> : 可以是指定編號格線的數字，也可以是指定命名的格線的名稱
    - grid-column-start: 2
  - span <number> : 該項目將跨越指定的網格數
    - grid-column-end: span 2
  - span <name> : 該項目將跨越直到它到達具有指定的名稱的下一行
    - grid-column-end: span col1
  - auto : 表示自動放置、自動跨度或一個預設跨度

子元素位置的空間定義

Example

```
.item-a {
  grid-column-start: 2;
  grid-column-end: five;
  grid-row-start: row1-start;
  grid-row-end: 3;
}
```



Grid 內容對齊方式

- 套用在子元素項目上(單獨對齊某一個項目)，基本上和flex item的對齊方式相同
- justify-self: start | end | center | stretch
  - 個別子元素的水平對齊方式
- align-self : start | end | center | stretch
  - 個別子元素的垂直對齊方式
- place-self: < justify-self > || < align-self >


國立臺中科技大學
NATIONAL TAICHUNG UNIVERSITY OF SCIENCE AND TECHNOLOGY

CSS媒體查詢

媒體查詢Media

- HTML和CSS允許網頁設計人員針對不同的媒體類型或媒體的規格特徵去訂做不同的樣式
- 媒體類型
  - all: 全部裝置(預設)
  - screen: 螢幕類型
  - print: 列印裝置，包含預覽列印的功能
  - speech: 語音合成器
- 媒體查詢功能可以用來檢查許多事情，例如：
  - 可視區域寬度與高度
  - 裝置的寬度與高度
  - 裝置方向
  - 裝置解析度

媒體查詢語法

- media 媒體類型 and 表達式
  - @media not|onlymediatype and (media feature) {CSS-Code;}
- 在link元素裡面的media屬性
  - <link rel="stylesheet" media=" mediatype and|not|only (media feature)" href=" print.css ">

常見媒體特徵

特徵與設定值	說明	min/max prefixes
width:長度	可視區域的寬度	yes
height:長度	可視區域的高度	yes
device-width	裝置螢幕的寬度	yes
device-height	裝置螢幕的高度	yes
orientation: landscape | portrait	裝置的方向(直向 | 橫向)	no
aspect-ratio: 比例	可視區域的寬高比	yes
device-aspect-ratio: 比例	裝置螢幕的寬高比	yes
resolution: 解析度	裝置螢幕的解析度，以dpi或dpcm為單位	yes
color: 正整數或0	裝置螢幕的色彩位元數目，0表示非彩色裝置	yes
color-index: 正整數或0	裝置螢幕的色彩索引位元數目，0表示非彩色裝置	yes

媒體查詢Example

1. 如果媒體類型為screen且可視區域寬度大於480px

```
@media screen and (min-width: 480px) {
    body {
        background-color: lightgreen;
    }
}
```

2. 如果媒體類型只能是screen

```
@media only screen and (max-width: 500px) {
    .manu {
        width:100%;
    }
}
```

RWD響應式網頁設計

開發適用於不同裝置的網頁

- 行動裝置螢幕較小，如果要使用者拉近拉遠或捲動左右可能會造成閱讀上的困難
- 行動裝置操作方式以觸控為主，有些網頁的按鈕太小或觸控沒有回饋效果，可能造成使用者操作上的困擾
- 大部分行動裝置不支援flash動畫

針對不同裝置開發不同網站

- 網站可能會分成電腦版網頁和行動裝置版網頁
- 優點
  - 完全可以量身訂做不同裝置上的網頁瀏覽體驗
- 缺點
  - 開發與維護成本隨著網頁規模而遞增
  - 不同版本網頁會有各自的網址

響應式網頁

- 目的是為了讓網頁在各種尺寸的裝置下，畫面都能呈現合適比例的設計原則
- 優點
  - 網頁內容只有一種使得開發維護成本降低
  - 網址位置統一
- 缺點
  - 開發時需要針對不同尺寸的裝置做測試
  - 無法充分發揮裝置的特點

響應式網頁設計的主要技術

- 媒體查詢
  - 透過CSS媒體查詢功能來選擇使用者的裝置適合套用的樣式
- 流動圖片
  - 設定圖片或物件元素的大小的時候，根據其容器的大小比例做縮放，不要設定絕對大小(px pt等長度)
- 流動格線
  - Flex layout或Grid layout

RWD 基本實作方法

- 設定 viewport
- 決定 RWD 設計模式
- 套用 CSS media query
- 使用相對單位設定寬高、大小

設定 viewport

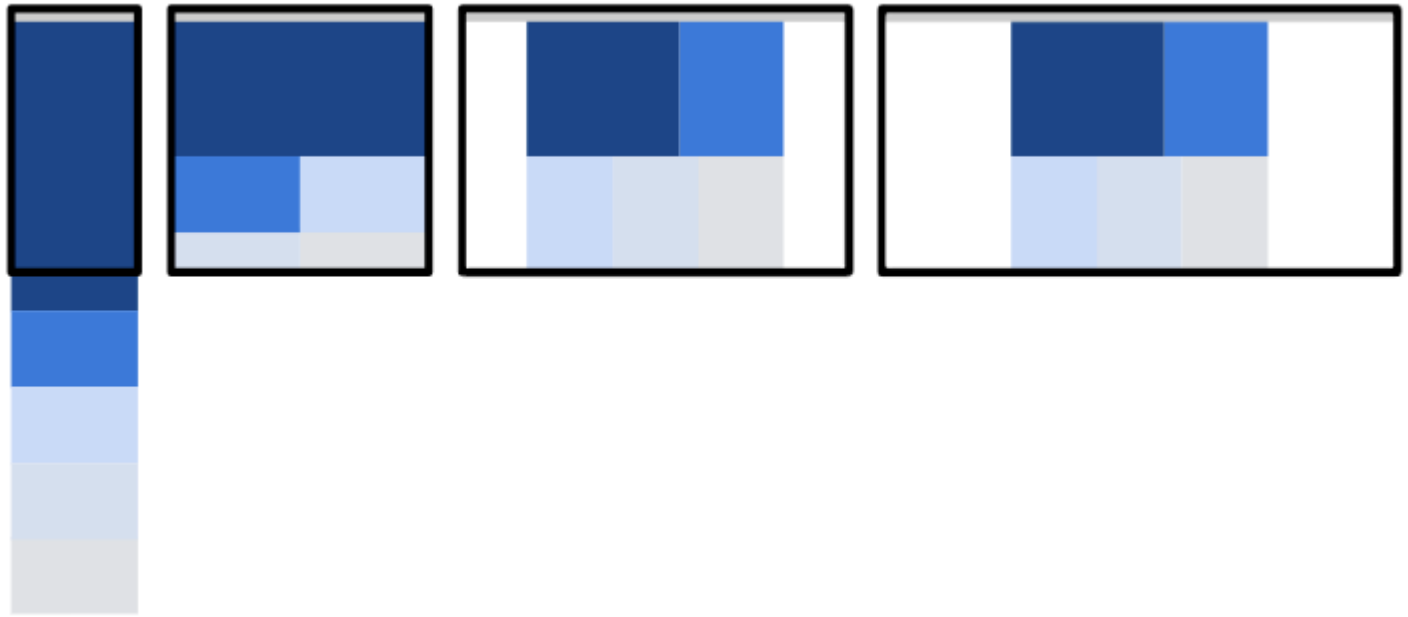
- viewport（檢視區）指的就是瀏覽網頁時瀏覽器顯示畫面內容的區域
- viewport設定基本是以手持裝置的螢幕大小滿版為原則
- 在html裡的head設定
  - 預設畫面為裝置寬度，載入初始縮放比例 100%
    - <meta name="viewport" content="width=device-width, initial-scale=1" >
  - 防止使用者做畫面縮放，將畫面鎖在縮放比例 100%
    - <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1" >
    - <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">

RWD設計模式

- 決定網頁內容在不同 viewport 下的版面如何配置、流動
- 常見設計模式
  - 局部流動 (mostly fluid)
  - 欄內容下排 (column drop)

局部流動 (mostly fluid)

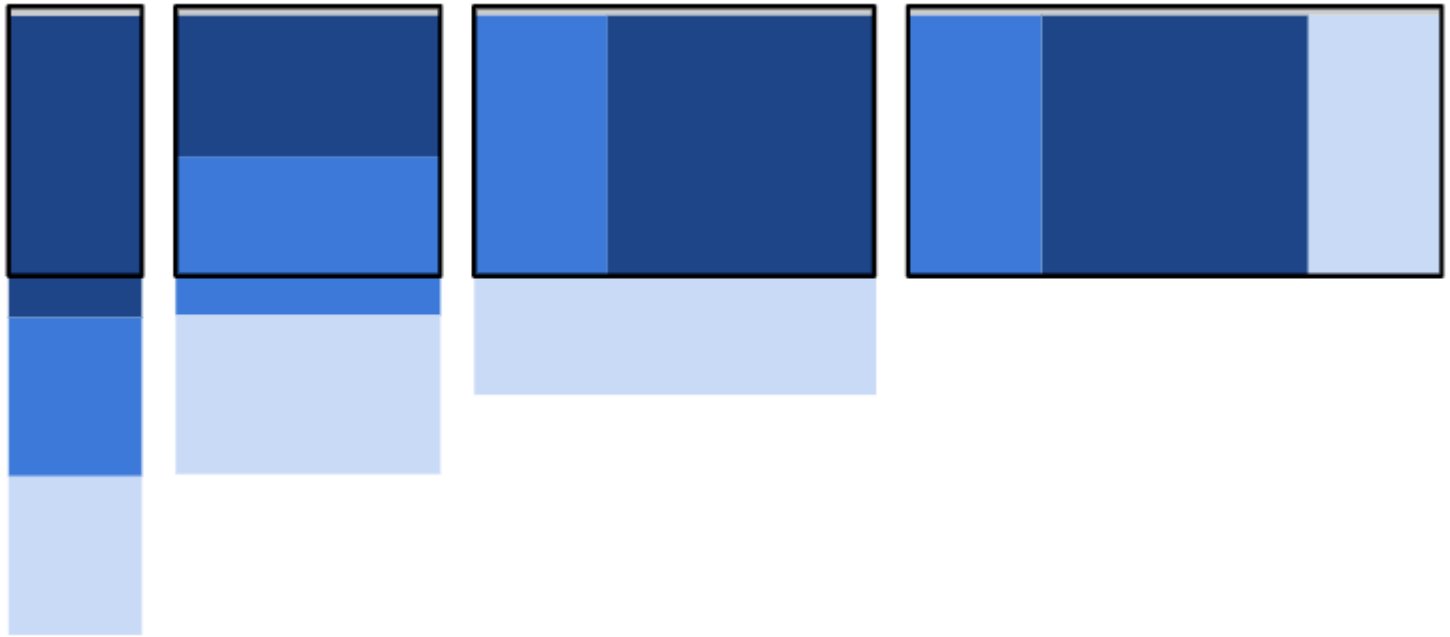
- mostly fluid 在小尺寸螢幕時，內容會以垂直堆疊的方式排列
- 隨著螢幕尺寸變大時，內容會流動成多欄的方式，直到達到某個尺寸後，將內容置中並停止流動



https://web.dev/learn/design/

欄內容下排（column drop）

- column drop 在大尺寸時會以多欄版面配置
- 隨著視窗寬度變窄時，內容會垂直堆疊所有欄



搭配媒體查詢

- RWD 中為了要在不同螢幕寬度做不同的 CSS 樣式調整，就需要使用 CSS3 中提供的媒體查詢語法