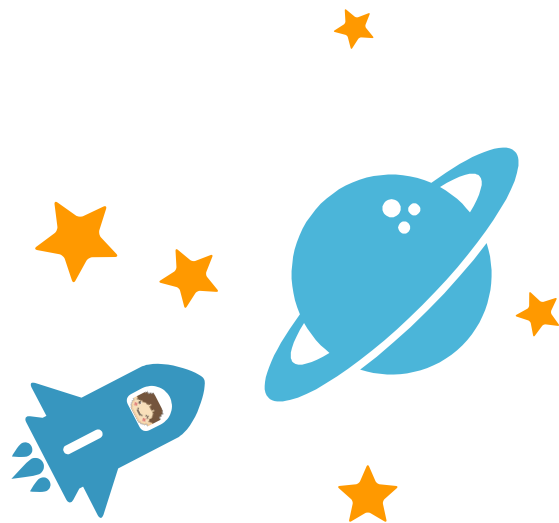
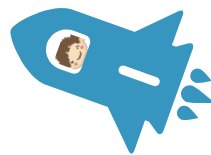


# CSS Animation

By Vincent





# “CSS製作動畫的2種方式”

# Transition

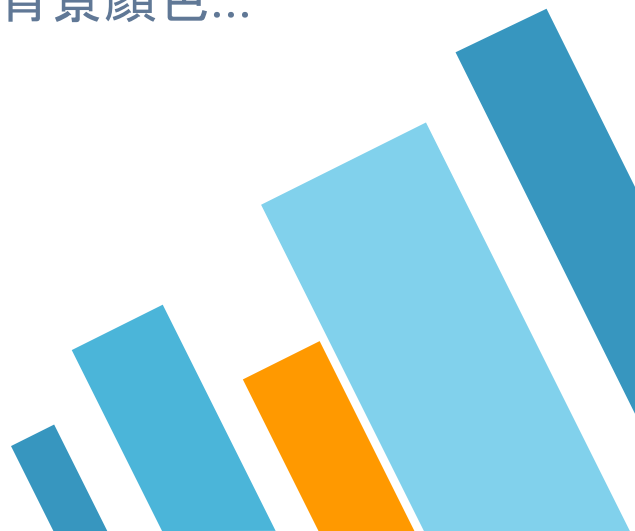
# Animation





# CSS Transition

# transition: **property** **duration** **timing-function** **delay**;

- » **Property:** 物件的長、寬、位置、字體大小、背景顏色...
  - » **Duration:** 動畫時間長
  - » **Timing Function:** 動畫進行的速度曲線
  - » **Delay:** 延後動畫開始的時間
- 



# CSS Transition Example

```
.cube {  
  width: 100px;  
  height: 100px;  
  background-color: #828282;  
  transition: transform 1s;  
}  
  
.cube.move {  
  transform: translate3d(200px, 0, 0) rotate(360deg);  
}
```



```
setInterval(function() {  
  $('.cube').toggleClass('move');  
}, 1000);
```





# Timing Function (控制動畫進行的速度曲線)

有ease(default)、linear、ease-in等等可選，也可自訂cubic-bezier(n,n,n,n)

```
transition: transform 1s; X transition: transform 1s cubic-bezier(.68, -.55, .265, 1.55);
```


# 自訂曲線工具網站: [cubic-bezier.com](https://cubic-bezier.com)





# CSS Animation


# animation: **name** **duration** **timing-function** **delay**  
iteration-count **direction** fill-mode **play-state**;

- » **Name:** @keyframes 動畫名稱
  - » **Duration:** 動畫時間長
  - » **Timing Function:** 動畫進行的速度曲線
  - » **Delay:** 延後動畫開始的時間
  - » **Iteration Count:** 動畫重複次數
  - » **Direction:** 動畫播放方向
  - » **Fill Mode:** 動畫開始前後是否保持動畫設定
  - » **Play State:** 控制動畫播放狀態
- 



# CSS Animation

```
# @keyframes animationName {  
  keyframes-selector { css-styles; }  
}
```

- » **Animation Name:** 動畫名稱(自訂)
  - » **Keyframes Selector:** 物件在動畫的某個時間點的狀態
- 

# CSS Animation Example

```
.cube {  
  width: 100px;  
  height: 100px;  
  background-color: #828282;  
  animation: mimic 2s infinite;  
}  
  
@keyframes mimic {  
  0% { transform: translate3d(0, 0, 0) rotate(0deg); }  
  50% { transform: translate3d(200px, 0, 0) rotate(360deg); }  
  100% { transform: translate3d(0, 0, 0) rotate(0deg); }  
}
```





# CSS Animation Example

```
@keyframes mimic {  
  0% {  
    transform: translate3d(0, 0, 0);  
  }  
  25% {  
    transform: translate3d(100px, 0, 0);  
    background-color: #4bb5d9;  
  }  
  50% {  
    transform: translate3d(100px, 100px, 0);  
    background-color: #81d1ec;  
  }  
  75% {  
    transform: translate3d(0, 100px, 0);  
    background-color: #ff9900;  
  }  
  100% {  
    transform: translate3d(0, 0, 0);  
  }  
}
```





# Animation Direction (控制動畫播放方向)

```
animation-direction: normal | reverse | alternate | alternate-reverse | initial | inherit;
```

```
animation: mimic 1s linear infinite alternate;
```

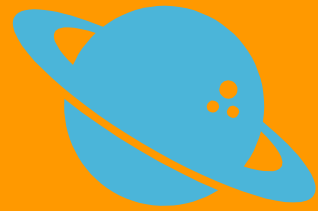
```
animation: mimic 1s ease infinite alternate;
```



# TRANSITION VS. ANIMATION

- Transition較常使用, 設定簡單
- 複雜的分鏡動畫只能用Animation做成

# Achieving **60 FPS** Animations with **CSS3**

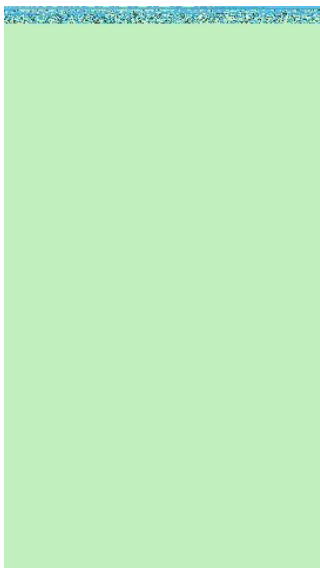


# Browser's rendering pipeline



- » **Styles:** 樣式計算, 計算每個元素的最終樣式。
- » **Layout:** 計算版面配置(width/height/left/top/right/bottom)。
- » **Paint:** 繪製物件像素圖層(box-shadow/color/background-color)。
- » **Composite:** 將圖層依順序繪製到畫面上(transform/opacity)。

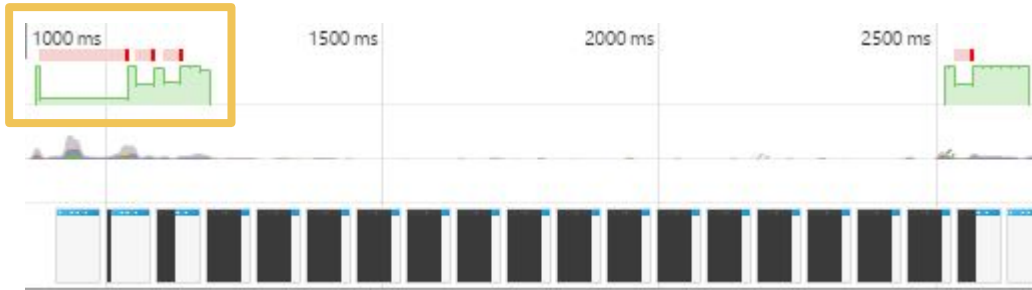
# Achieve the 60 Frames Per Second Mark



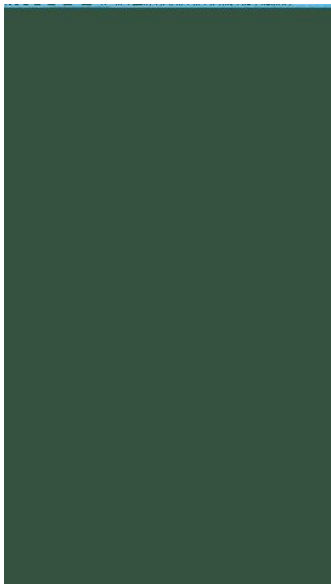
```
.app-menu {  
  left: -300px;  
  transition: left 300ms linear;  
}  
  
.app-menu-open .app-menu {  
  left: 0px;  
  transition: left 300ms linear;  
}
```

\* 錯誤示範

# Achieve the 60 Frames Per Second Mark



# Achieve the 60 Frames Per Second Mark



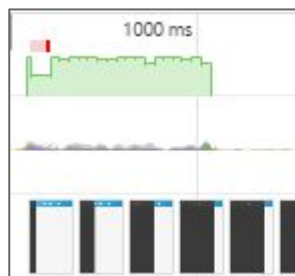
```
.app-menu {  
  -webkit-transform: translateX(-100%);  
  transform: translateX(-100%);  
  transition: transform 300ms linear;  
}  
.app-menu-open .app-menu {  
  -webkit-transform: none;  
  transform: none;  
  transition: transform 300ms linear;  
}
```



# Achieve the 60 Frames Per Second Mark



修改前



修改後

# Achieve the 60 Frames Per Second Mark

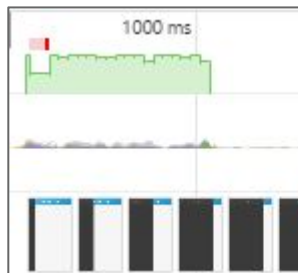
```
.app-menu {  
  transform: translate3d(-100%, 0, 0);  
  transition: transform 300ms linear;  
}  
.app-menu-open .app-menu {  
  transform: none;  
  transition: transform 300ms linear;  
}
```

利用特定CSS樣式強制觸發瀏覽器使用GPU繪製畫面

# Achieve the 60 Frames Per Second Mark



修改前



改transform位移



觸發GPU繪製

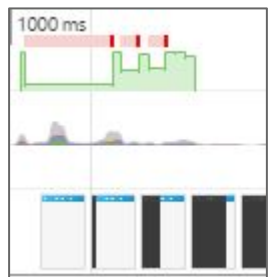
# Achieve the 60 Frames Per Second Mark

```
<div class="layout">
  <div class="app-menu"></div>
  <div class="header">
    <div class="menu-icon"></div>
  </div>
</div>
```

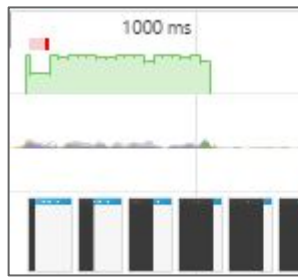


```
<div class="menu">
  <div class="app-menu"></div>
</div>
<div class="layout">
  <div class="header">
    <div class="menu-icon"></div>
  </div>
</div>
```

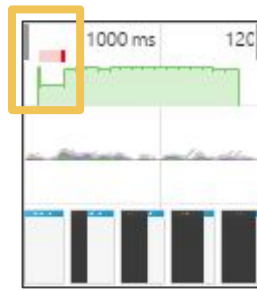
# Achieve the 60 Frames Per Second Mark



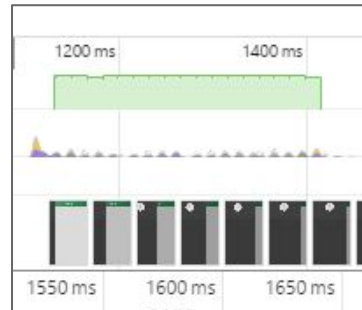
修改前



改transform位移



觸發GPU繪製



修改架構



## 動畫設定注意事項

- » 過場動畫時間不用設太長(300ms ~ 700ms)。
  - » 網頁功能正常為優先考量，動畫不是必要。
  - » 做各種瀏覽器測試，避免用到不支援的CSS語法。
- 