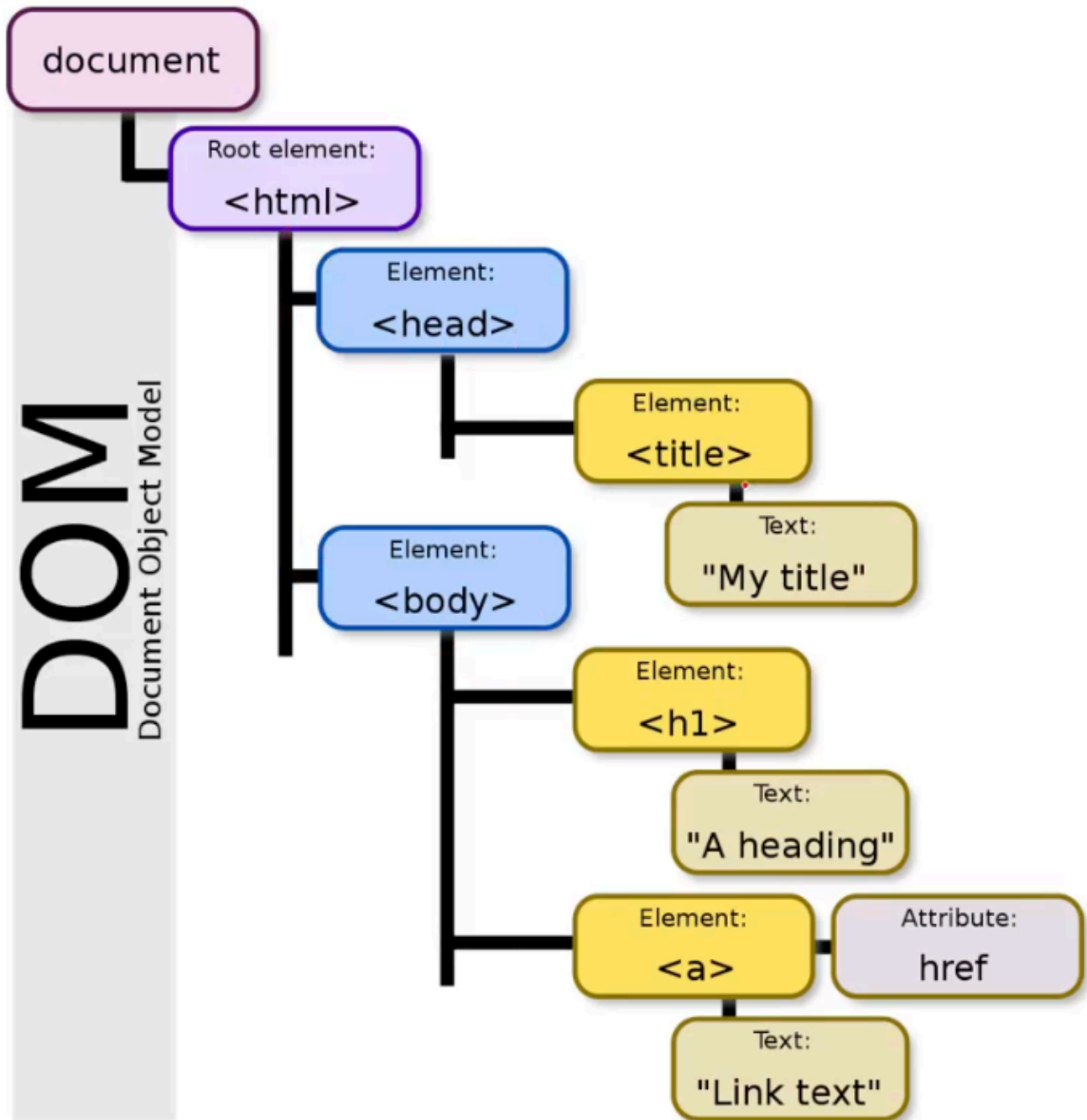


CSS

Cascading Style Sheets，用來控制HTML的外觀與佈局。包含字體、顏色、尺寸、間距、邊框、背景、浮動、定位等等。

DOM Tree

Document Object Model，加載到瀏覽器當中的樹狀表示。



Inside nesting structure:

- Parent Node (Parent element)
- Child Node (Child element)

這兩者可以互相嵌套，形成一個樹狀的結構。

可放置的位置

- **Inline** only apply to current element

```
<h1 style="color: green"></h1>
```

設定對象較為侷限，不易維護。

- **Internal** only apply to current page

```
``` html
<head>
 <style>
 h1 {
 color: green;
 }
 </style>
</head>
```

若頁面較多，則需要在每個頁面都加入，不易維護，且會造成頁面載入時間過長。

- **external** apply to all pages linked to this file

```
<link rel="stylesheet" href="style.css">
```

一個獨立的檔案，可在所有頁面中使用，且只需載入一次，容易維護且可大幅減少頁面載入時間。

Priority: *inline* > *internal* > *external*

## CSS 重要概念

Cascading Style Sheets，層疊樣式表

- Inheritance繼承

常見可被繼承的屬性有：

- color
- font-family
- font-size
- list-style-type
- text-align

更多可參考[連結](#)。

User agent (or user stylesheet)優先度較高，可覆蓋繼承的屬性。而瀏覽器預設為User的代理人，因此要注意瀏覽器預設的樣式可能會影響到網頁的呈現，因此 `<a>` 通常需要額外設定。

- Conflicting Styling

同一個 tag 若有兩個以上的樣式設定，則會以最後一個為主。

若衝突是發生在不同的樣式表中，則會以最後載入的樣式表為主。

衝突處理原則：

- Priority 優先度
  1. inline style
  2. user stylesheet
  3. user agent stylesheet
  4. inheritance
- Specificity 特定度  
當一個 tag 同時有 id、class、tag 時，會依照以下順序優先度較高：
  1. id (100)
  2. class (010)
  3. tag (001)
- Order 順序  
若優先度與特定度相同，則會依照樣式表的順序來決定。

## 顏色設定

---

- Color name

```
h1 {
 color: red;
}
```

有限的顏色名稱，不易維護。

- RGB

```
h1 {
 color: rgb(255, 0, 0);
}
```

紅、綠、藍，0為最暗，255為最亮，每個值都介於0~255之間，每個值都用1byte（8bit）來表示。

- rgba

```
h1 {
 color: rgba(255, 0, 0, 0.5);
}
```

第四個為透明度，0為完全透明，1為完全不透明。

- HEX

```
h1 {
 color: #ff0000;
}
```

16進位，每個值都用2byte（16bit）來表示。

- HSL

```
h1 {
 color: hsl(0, 100%, 50%);
}
```

Hue（色相）、Saturation（飽和度）、Lightness（亮度），色相為0~360，飽和度為0~100%，亮度為0~100%。

For more information, please refer to [CSS Color](#).

## CSS Selectors

### Selector

- Universal Selector

```
* {
 color: red;
}
```

選取所有元素。

- Element Selector

```
h1 {
 color: red;
}
```

選取所有 `h1` 元素。

- ID Selector

```
#my_id {
 color: red;
}
```

開頭 `#` 選取所有 `id` 為 `my_id` 的元素。`id` 為唯一值，不可重複。

- Class Selector

```
.my_class {
 color: red;
}
```

開頭 . 選取所有 class 為 my\_class 的元素。 class 可重複，多個 class 用空格分開。

```
<h1 class="my_class1 my_class2">Hello World</h1>
```

```
.my_class1.my_class2 {
 color: red;
}
```

ID, Class, Element Selector 以及其他tag可以互相組合使用，例如：

```
/*eg1*/
#my_id.my_class {
 color: red;
}

/*eg2*/
a.my_class {
 color: red;
}

/*eg3*/
a#my_id {
 color: red;
}
```

Selector Priority:

ID Selector > Class Selector > Element Selector

- Group Selector

```
h1, h2, h3 {
 color: red;
}
```

選取所有 h1 、 h2 、 h3 元素。

- Descendant Selector

```
<h2>
 Hello World
</h2>
```

```
h2 span {
 color: red;
}
```

選取所有 `h2` 元素中的 `span` 元素。依據DOM Tree，的結構，範例中 `span` 為 `h2` 的子元素，非透過 `id` 或 `class` 來選取。

- Attribute Selector

```
input[type="password"] {
 color: red;
}
```

- Others

- Pseudo-class

指定元素的特殊狀態。例如當元素被點擊時，會變成 `:active` 狀態；或是 `:hover` 則是當滑鼠移動到元素上時會變成的狀態。

```
input[type="text"]:active {
 color: green;
}
```

- Pseudo-element

添加到選擇棄的關鍵字，用來設置所選元素的特定部分的樣式。

```
p::before {
 content: "This is a paragraph: <<<";
 color: blue;
}
```

## CSS 單位

- absolute

- `px`: pixel, 2.54cm(1 inch) = 96px
- `cm`
- `mm`
- `in`

- relative

- `em`: 相對 parent element 的長度，越深層的元素，`em` 的值越小，也越難計算，因此會盡量避免使用。1em就等於 parent element 的大小。
- `rem`: root em，會找到 `<html>` 的大小，然後再依照em的規則來計算。
- `vw`: viewport width，1vw就等於 viewport (瀏覽器視窗)寬度的 1/100。

- `vh`: viewport height，1vh就等於 viewport (瀏覽器視窗)高度的 1/100。
- `%`: 相對於 parent element 的值。

預設的設定大小可以到 [這邊](#) 查看。

## 文字樣式 text-styling

---

- font-size: 設定值參考 [CSS 單位章節](#)
- text-align