HTML5 - Introduzione

- **HyperText Markup Language**
- Linguaggio di markup per strutturare contenuti web
- Standard mantenuto dal W3C (World Wide Web Consortium)
- Versione attuale: HTML5 (2014)

Breve Storia di HTML

Anno	Versione	Note
1991	HTML	Nato per condividere documenti scientifici
1995	HTML2	Primo standard ufficiale
1997	HTML4	Introduzione CSS
2000	XHTML	Versione XML-based
2014	HTML5	Nuovi elementi semantici, API

Struttura Base HTML5

```
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <title>Titolo Pagina</title>
</head>
<body>
    <!-- Contenuto -->
</body>
</html>
```

Tag Classici HTML

- `h1` - `h6` : Intestazioni
- `p` : Paragrafo
- `a` : Link (`<a href="...">`)
- `img` : Immagine (`<img src="..." alt="...">`)
- `ul/ol/li` : Liste
- `table` : Tabella

Esempi Liste

```
<!-- Lista non ordinata -->
<ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
</ul>

<!-- Lista ordinata -->
<ol>
    <li>Primo</li>
    <li>Secondo</li>
</ol>
```

Nuovi Tag Semantici HTML5

- `header` : Intestazione
- `nav` : Menu di navigazione
- `main` : Contenuto principale
- `article` : Articolo autonomo
- `section` : Sezione logica
- `aside` : Contenuto correlato
- `footer` : Piè di pagina

Esempio Tag `nav`

```
<nav>
    <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/about">About</a></li>
        <li><a href="/contact">Contact</a></li>
    </ul>
</nav>
```

Il Tag `div`

- Contenitore generico senza significato semantico
- Usato per raggruppare elementi e applicare stili CSS
- Fondamentale per il layout prima di Flexbox/Grid

```
<div class="container">
    <div class="box">Box 1</div>
    <div class="box">Box 2</div>
</div>
```

Esempio Layout con `div`

```
<div class="page">
    <div class="header">Intestazione</div>
    <div class="content">
        <div class="sidebar">Menu</div>
        <div class="main">Contenuto</div>
    </div>
    <div class="footer">Footer</div>
</div>
```

Esempio Tabella

```
<table>
    <thead>
        <tr>
            <th>Nome</th>
            <th>Età</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Mario</td>
            <td>30</td>
        </tr>
    </tbody>
</table>
```

CSS - Introduzione

- **Cascading Style Sheets**
- Linguaggio per lo stile delle pagine web
- Separa contenuto (HTML) dalla presentazione (CSS)
- Versione attuale: CSS3

Sintassi CSS

```
selettore {
    proprietà: valore;
    /* Commento */
}
```

Esempio:

```
p {
    color: red;
    font-size: 16px;
}
```

Tipi di Selettori CSS

1. **Tag**: `p`, `div`, `h1`
2. **Classe**: `.menu`
3. **ID**: `#header`
4. **Attributo**: `[type="text"]`
5. **Pseudoclassi**: `:hover`, `:first-child`
6. **Combinatori**: `div p`, `ul > li`

Esempi Selettori

```css
/* Tutti i paragrafi */
p { color: blue; }

/* Elementi con classe "highlight" */
.highlight { background: yellow; }

/* Elemento con ID "main" */
#main { width: 80%; }

/* Link al hover */
a:hover { text-decoration: underline; }
```

Specificità CSS

Peso dei selettori (da meno a più specifico):

1. Elementi e pseudoelementi (`p` , `::before`)
2. Classi, attributi, pseudoclassi (`.class` , `:hover`)
3. ID (`#id`)
4. Stili inline (`style="..."`)

!important sovrascrive tutto (da usare con cautela)

Calcolo Specificità

Sistema a punteggio (a, b, c, d):

- **a**: Stili inline (1-0-0-0)
- **b**: ID (0-1-0-0)
- **c**: Classi/attributi/pseudoclassi (0-0-1-0)
- **d**: Elementi (0-0-0-1)

Esempio: `#nav .item:hover` = 0-1-2-0

Flexbox - Introduzione

- Modello di layout 1D (una dimensione alla volta)
- Controllo su distribuzione, allineamento e ordine elementi
- Basato su **container** e **item**

```
.container {
    display: flex;
}
```

Terminologia Flexbox

- **Container**: Elemento padre con `display: flex`
- **Items**: Figli diretti del container
- **Main Axis**: Asse principale (definito da `flex-direction`)
- **Cross Axis**: Asse perpendicolare

Flex Direction

Definisce l'asse principale:

```
.container {
    flex-direction: row; /* Default (sinistra a destra) */
    flex-direction: row-reverse;
    flex-direction: column; /* Alto a basso */
    flex-direction: column-reverse;
}
```

Flex Wrap

Controlla il wrapping degli items:

```
.container {
    flex-wrap: nowrap; /* Default (no wrap) */
    flex-wrap: wrap; /* Wrap normale */
    flex-wrap: wrap-reverse; /* Wrap inverso */
}
```

Justify Content

Allineamento lungo l'asse principale:

```
.container {
    justify-content: flex-start; /* Default */
    justify-content: flex-end;
    justify-content: center;
    justify-content: space-between;
    justify-content: space-around;
    justify-content: space-evenly;
}
```

Align Items

Allineamento lungo l'asse secondario:

```
.container {
    align-items: stretch; /* Default */
    align-items: flex-start;
    align-items: flex-end;
    align-items: center;
    align-items: baseline;
}
```

Esempio Pratico Flexbox

```
<div class="menu">
    <div class="item">Home</div>
    <div class="item">About</div>
    <div class="item">Contact</div>
</div>
```

```
.menu {
    display: flex;
    justify-content: space-around;
    align-items: center;
}
```

CSS Grid - Introduzione

- Modello di layout 2D (righe e colonne)
- Più potente di Flexbox per layout complessi
- Basato su **container** e **item**

```
.container {
    display: grid;
}
```

Terminologia Grid

- **Grid Container**: Elemento padre
- **Grid Items**: Figli diretti
- **Grid Line**: Linee divisorie
- **Grid Track**: Spazio tra due linee (riga/colonna)
- **Grid Cell**: Singola cella
- **Grid Area**: Gruppo di celle

Definire la Grid

```
.container {
    display: grid;
    grid-template-columns: 100px 200px auto;
    grid-template-rows: 50px 100px;
    gap: 10px; /* Spazio tra celle */
}
```

Unità `fr`

Unità frazionaria per distribuzione proporzionale:

```
.container {
    grid-template-columns: 1fr 2fr 1fr;
    /* 1 parte | 2 parti | 1 parte */
}
```

Nomi Aree

```
.container {
    grid-template-areas:
        "header header header"
        "sidebar main main"
        "footer footer footer";
}

.header { grid-area: header; }
.sidebar { grid-area: sidebar; }
/* ... */
```

Esempio Pratico Grid

```
<div class="page">
    <header>Header</header>
    <aside>Sidebar</aside>
    <main>Content</main>
    <footer>Footer</footer>
</div>
```

```
.page {
    display: grid;
    grid-template-columns: 200px 1fr;
    grid-template-areas:
        "header header"
        "sidebar main"
        "footer footer";
}
```

SASS/SCSS - Introduzione

- **Syntactically Awesome Style Sheets**
- Preprocessore CSS che aggiunge funzionalità
- Due sintassi:
  - SASS (indentazione)
  - SCSS (simile a CSS)

Variabili in SASS

```
// Definizione
$primary-color: #3498db;
$spacing-unit: 1rem;

// Utilizzo
.header {
    background: $primary-color;
    padding: $spacing-unit;
}
```

Nesting in SASS

```
nav {
    ul {
        margin: 0;
        li {
            display: inline-block;
            a {
                color: blue;
            }
        }
    }
}
```

Mixins

Blocchi riutilizzabili di stile:

```
@mixin flex-center {
    display: flex;
    justify-content: center;
    align-items: center;
}

.container {
    @include flex-center;
}
```

Mixins con Parametri

```
@mixin box($width, $height: $width) {
    width: $width;
    height: $height;
}

.square {
    @include box(100px);
}

.rectangle {
    @include box(100px, 200px);
}
```

Funzioni in SASS

```
@function calculate-rem($size) {
    @return $size / 16px * 1rem;
}

body {
    font-size: calculate-rem(18px);
}
```

Estendere Stili (@extend)

```
.message {
    padding: 10px;
    border: 1px solid #ccc;
}

.success {
    @extend .message;
    border-color: green;
}
```

Condizionali (@if, @else)

```
@mixin theme($dark: false) {
    @if $dark {
        background: black;
        color: white;
    } @else {
        background: white;
        color: black;
    }
}
```

Loop (@for, @each)

```
// @for
@for $i from 1 through 4 {
    .col-#{$i} {
        width: 25% * $i;
    }
}

// @each
$colors: red, green, blue;
@each $color in $colors {
    .#{$color}-text { color: $color; }
}
```

Modularità (@import)

```
// _variables.scss
$primary: #3498db;

// main.scss
@import 'variables';
body {
    color: $primary;
}
```

Partials in SASS

- File che iniziano con `_` (es. `_variables.scss`)
- Non compilati in CSS separati
- Importati con `@import` senza l'underscore

Differenze SASS/SCSS

Feature	SASS	SCSS
Sintassi	Indentazione	Parentesi
Estensione	.sass	.scss
Compatibilità	Meno	Più CSS-like

Esempio Complesso SASS

```
// _variables.scss
$colors: (
    primary: #3498db,
    secondary: #2ecc71
);

// _mixins.scss
@mixin responsive($breakpoint) {
    @media (min-width: $breakpoint) {
        @content;
    }
}

// main.scss
@import 'variables', 'mixins';
```

Compilazione SASS

1. Installa SASS: `npm install -g sass`
2. Compila: `sass input.scss output.css`
3. Watch: `sass --watch input.scss:output.css`

Best Practice SASS

- Usa partials per organizzare il codice
- Nomina variabili in modo descrittivo
- Evita nesting troppo profondo (> 4 livelli)
- Usa mixins per codice riutilizzabile
- Commenta il codice complesso

Altre Funzionalità HTML5

- **Canvas**: Disegno dinamico
- **Geolocation**: Posizione utente
- **LocalStorage**: Memorizzazione locale
- **Web Workers**: Esecuzione in background
- **Drag & Drop**: Interazione utente

Conclusioni

- **HTML5**: Struttura semantica
- **CSS**: Stile e layout (Flexbox/Grid)
- **SASS/SCSS**: CSS potenziato

Prossimi passi: JavaScript, Framework, Accessibilità

Grazie!

Domande?