Corso Progettista web

ed.2020

01 - Cosa è wordpress

"Wordpress è una piattaforma di personal publishing ed anche CMS"

se volete conoscere di più sulla sua storia potete partire da qui:
http://it.wikipedia.org/wiki/WordPress

Caratteristiche

- Interfaccia di amministrativa della piattaforma
- Estensione delle funzionalità tramite plugins;
- Disponibilità di migliaia di temi gratuiti, o a pagamento, per personalizzare l'aspetto del blog;
- Gestione delle pagine con template (se previsto dal tema scelto);
- URL permanenti che aiutano l'indicizzazione nei motori di ricerca;
- Gestione dei post per categorie;
- Gestione dei post per tag;
- Funzioni di Trackback e Pingback;

- Editor WYSIWYG per la formattazione dei testi;
- Creazione di pagine statiche;
- Supporto multi-autori;
- Supporto multi-sito (attivabile dalla versione 3.0, o usando la variante WordPress MU);
- Log degli utenti che visitano il blog;
- Blocco di utenti in base all'indirizzo ip;
- Possibilità di specificare meta-tag;
- Dalla versione 3.7 è presente un sistema di aggiornamento automatico.

02 - Differenza tra wordpress.org wordpress.com

"wordpress.org è il sito della comunità di wordpress"

"wordpress.com invece vi vende il servizio di hosting per wordpress."

dal **primo** (.org) potrete scaricare il pacchetto con cui installare il vostro wordpress.
Fornisce supporto agli sviluppatori ed agli utilizzatori di wordpress, li troverete i temi ed i plugin che arricchiranno la vostra installazione.

Il **secondo** (.com) è un servizio di hosting per wordpress gestito da Automattic, si basa sulla versione multi utente del software WordPress.
Consente la creazione di uno o più blog gratuitamente ed offre alcuni servizi accessori a pagamento. Per ogni blog registrato il servizio offre gratuitamente un indirizzo di terzo livello del tipo nomescelto.wordpress.com
Ma anche le cosa gratuite hanno "un prezzo"... limiti forti sulla gestione della vostra installazione di wordpress, nessun plugin e molti temi non potranno essere installati.

Per i curiosi qui potete trovare un raffronto tra i due,
http://en.support.wordpress.com/com-vs-org/

03 - Come installare Wordpress

Bitnami Application

https://bitnami.com/stack/wordpress
https://wiki.bitnami.com/Applications/BitNami_Wordpress
Da qui possiamo scaricare l'installer per il nostro sistema operativo, farà lui il lavoro per noi...

Per i più temerari ecco la...

Procedura manuale... paura eh!

innanzitutto scaricare l'ultima versione di wordpress da https://it.wordpress.org/
https://codex.wordpress.org/it:Installare_WordPress

• Scaricate e decomprimete il pacchetto compress di wordpress se non lo avete ancora fatto • Create un database MySQL per la vostra installazione di wordpress sul vostro server (Locale o Web), create anche un utente MySQL con tutti i privilegi di accesso e modifica al database appena citato • individuate il file wp-config-sample.php e rinominatelo wp-config.php. <--- opzionale	• Aprite wp-config.php con un editor testuale ed inserite i dettagli di configurazione del vostro DB (nome del server, nome del DB, utente e password) (https://codex.wordpress.org/it:Installare_WordPress#Step_3:_Set_up_wp-config.php) come spiegato qui https://codex.wordpress.org/it:Modificare_wp-config.php, questo vi permetterà di generare un password per la vostra chiave di criptazione... ... facile no? • Caricate i file di wordpress (non la cartella che li conteneva) nella posizione desiderata sul vostro server • Eseguite lo script di installazione accedendo a wp-admin/install.php tramite il vostro browser web.

Requisiti minimi

Se intendete installare wordpress su di un server web assicuratevi che quest'ultimo possegga i seguenti requisiti minimi (https://wordpress.org/about/requirements/).
Se non ne avete idea ... semplicemente chiedetelo al gestore del vostro spazio web.
Nella pagina riferita dal link indicato troverete un prototipo del messaggio da copiare, incollare ed inviare al vostro gestore.

04 - Schermo di amministrazione e la dashboard

"E qui comando io, e questa è casa mia..."

Lo schermo di amministrazione

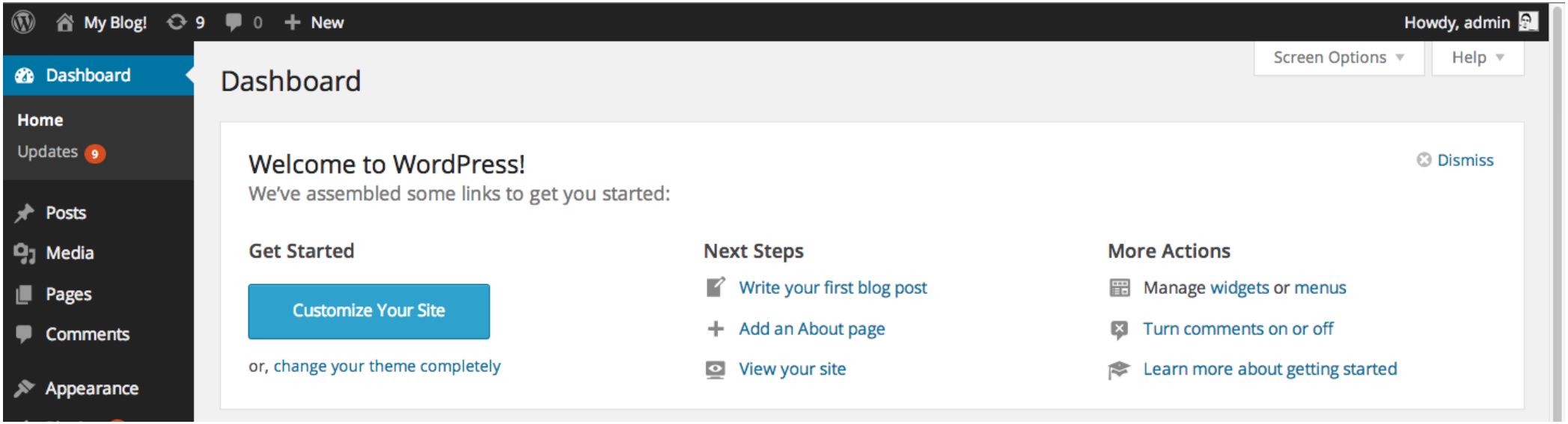
Lo schermo di amministrazione raccoglie tutte le funzionalità di gestione di wordpress.
E' composto da un toolbar (nell'intestazione), una navigazione principale (la colonna nera a sinistra), una area di lavoro ed un footer.

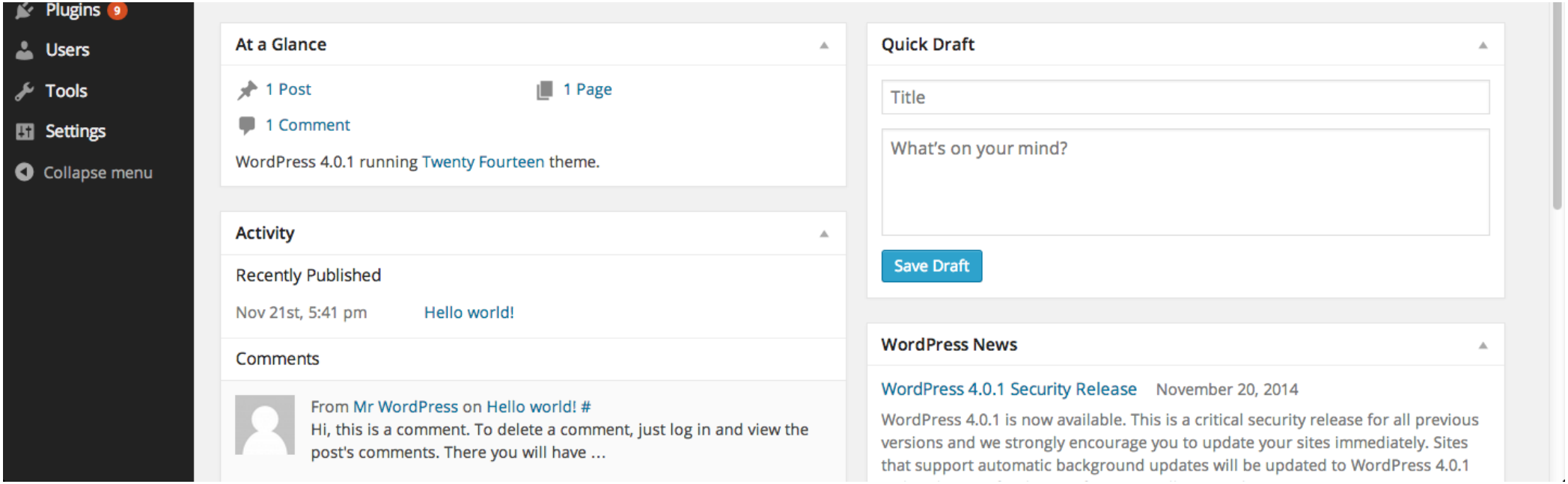
Per saperne di più: http://codex.wordpress.org/Administration_Screens

La Dashboard

E' una schermata personalizzabile, qui potrete avere anteprime sui contenuti del sito e informazioni provenienti dalla comunita di wordpress, statistiche, collegamenti rapidi a varie funzioni.

Per saperne di più: http://codex.wordpress.org/Dashboard_Screen





05 - I tipi di post

Come i tipi di post? Quanti post ha wordpress?

Wordpress possiede 5 tipi di post standard disponibili agli utilizzatori o utilizzati internamente dal software:

- Post (Post Type: 'post')
- Page (Post Type: 'page')
- Attachment (Post Type: 'attachment')
- Revision (Post Type: 'revision')
- Navigation menu (Post Type: 'nav_menu_item')

in più è possibile definire dei post personalizzati

- Custom post types (occhio è #robbadasviluppatori)

Il riferimento tecnico
http://codex.wordpress.org/Post_Types

06 - I ruoli

"A ciascuno il suo"

Wordpress fornisce la possibilità di aggiungere nuovi utenti che possano collaborare nella gestione e nell'aggiornamento del sito, la libertà di azione che avranno dipende dalle capacità che gli derivano dal ruolo.

SuperAdmin
Colui che tutto può, ha senso solo per i network di siti wordpress (e come ogni divinità appare a pochi)

Administrator
L'administrator gestisce il sito in maniera completa e ha la possibilità di intervenire in ogni area del pannello di amministrazione (uno lo avete creato durante l'installazione di wordpress)

Editor
L'editor può modificare il proprio profilo, può creare articoli e pagine, modificare articoli e commenti pubblicati dagli altri utenti. L'editor può anche creare nuove categorie di articoli.

Author
L'autore può modificare il proprio profilo, può creare articoli, modificarli o cancellarli, non ha la possibilità di intervenire sugli articoli degli altri utenti o di creare pagine.

Contributor
Il collaboratore può modificare il proprio profilo, può scrivere e modificare i propri articoli ma non può pubblicarli (la pubblicazione avviene con l'approvazione dell'administrator o dell'editor), non ha la possibilità di creare pagine.

Subscriber

Subscriber

Il subscriber può solo fare il login all'interno della piattaforma e modificare il proprio profilo… a cosa serve allora? … Mai visto pagine visibili solo se ci si è autenticati in backend sul sito (pagine private in wordpress)?

07- I temi

"E' questione di pelle"

Un tema di WordPress vi permette di controllare lo stile del **front end** del vostro sito WordPress.
La maggior parte dei temi WordPress comprendono:

- il design o stile complessivo del sito
- Lo stile dei font utilizzati
- i colori
- i punti in cui sono piazzati i widget (??? Cosa sono?)
- i layout di pagina (o modelli se previsti)
- stili per i post del blog e le pagine archivio del blog
- qualsiasi altro dettaglio stilistico

Tutto questo senza toccare la "parte di programmazione" che agisce sul sito!

E' possibile scegliere il tema preferito tra migliaia di temi, vederne l'effetto in anteprima ed apportare alcune modifiche al tema stesso dalla sezione specifica dello schermo di amministrazione.

Temi gratuiti vs. temi a pagamento

Proprio come per i plugin di WordPress, è possibile scegliere tra temi gratuiti o a pagamento presenti in WordPress Theme Directory.
Generalmente gli ultimi, indicati come premium o commercial, offrono una qualità superiore, maggiori possibilità di personalizzazione (più avanti ne vedremo almeno uno) ed il supporto nel caso in cui abbiate problemi nell'utilizzo del tema.

Per saperne di più sui temi : http://codex.wordpress.org/it:Utilizzare_i_Temi

Per saperne di più sui widget: http://codex.wordpress.org/WordPress_Widgets

08 - I menu

"... noi volevamo sapere, per andare, dove dobbiamo andare, per dove dobbiamo andare? Sa è una semplice informazione..."

I menu (per wordpress) sono dei semplici raggruppamenti di link, vengono normalmente prelevati dall'alberatura principale del sito e presentati nella zona comunemente chiamata area di navigazione principale.
Se il tema scelto lo permette sarà possibile creare anche dei custom menus, raggruppamenti di link a piacimento, anche pagine che mostrino solo una (o più) categoria di post... ho detto niente...

Per saperne di più sui menu: http://en.support.wordpress.com/menus/

09 - Come estendere wordpress: i plugin

"Miwa! lanciami i componenti!"

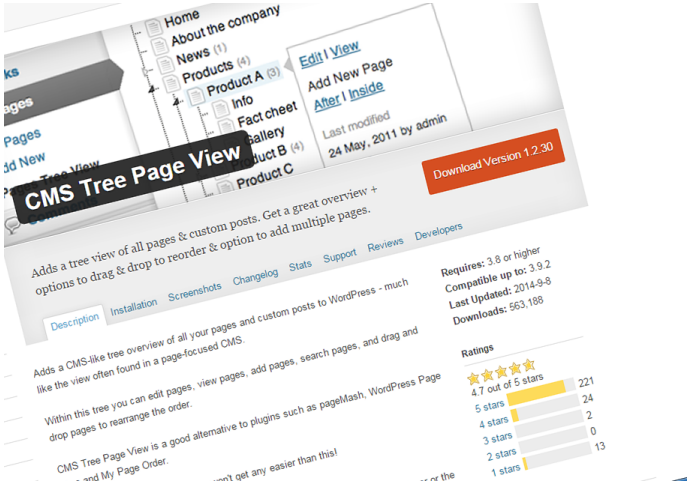
I plugins sono componenti per aggiungere funzionalità a worpdress... immaginate una funzionalità... molto probabilmente esiste già il plugin per ottenerla senza dover intervenire sul codice.
Potete cercarli ed installarli dallo schermo di amministrazione, sezione omonima, oppure da qui: https://wordpress.org/plugins/

Come scegliere i plugin

Lanciata una ricerca tra le migliaia di plugin (più di 36.000) ed individuato uno o più plugins assicuratevi che:

- il plugin sia compatibile con la vostra installazione di wordpress;

- che sia costantemente aggiornato;
- come lo hanno valutato altri utilizzatori (ratings);
- è difficile da installare?
- è possibile ottenere assistenza?



Per saperne di più sui plugins: http://codex.wordpress.org/Writing_a_Plugin
Per saperne di più su *Jeeg Robot d'Acciaio*: http://it.wikipedia.org/wiki/Jeeg_robot_d%27acciaio

10 - Child Theme

"Sarebbe un lavoro da sviluppatore, ma con il plugin giusto...
Creeremo un nostro child theme in men che non si dica"

Si, ma… che cosa è un child theme?

E' la derivazione di un tema, è il tema su cui lavorerete, sbagliando anche … senza distruggere il tema da cui siete partiti.
Il child theme erediterà tutte le funzioni, i fogli di stile, caratteristiche del tema padre, noi dovremo fare solo le modifiche che ci servono.

Mi piace quel tema, lo scarico, creo un child theme e lo stravolgo! Senza rompere l'originale.
Il riferimento tecnico: http://codex.wordpress.org/it:Temi_Child
Il plugin: https://wordpress.org/plugins/child-theme-configurator/
come funziona il plugin dal sito del produttore http://www.lilaeamedia.com/plugins/child-theme-plugin-styles/

altre risorse:
https://developer.wordpress.org/themes/basics/
https://premium.wpmudev.org/blog/how-to-create-wordpress-child-theme/

11 - Il Loop

Il loop è il codice php che mostra i post di wordpress
https://codex.wordpress.org/it:Il_Loop

12 - Template Hierarchy

Spiega come WordPress determina quale o quali template ("file modello") utilizzare nelle pagine che intendete creare o modificare. Se si desidera personalizzare un tema esistente WordPress vi aiuterà a decidere quale file deve essere copiato e modificato.
https://developer.wordpress.org/themes/basics/template-hierarchy/

https://developer.wordpress.org/themes/basics/template-hierarchy/
https://codex.wordpress.org/images/1/18/Template_Hierarchy.png

Pagine per una categoria

https://codex.wordpress.org/Category_Templates

La template Hierarchy per le pagine di categoria è la seguente

1. category-slug.php
2. category-ID.php
3. category.php
4. archive.php
5. index.php

Se dovete creare una template per category-slug o ID copiate e modificate category.php, se non lo trovate nel vostro tema o nel tema perda cercate archive, se non lo trovate index.php

Realizzare template di pagina

https://code.tutsplus.com/it/tutorials/dynamic-page-templates-in-wordpress-1--cms-28366
https://code.tutsplus.com/it/tutorials/dynamic-page-templates-in-wordpress-part-2--cms-28393
https://code.tutsplus.com/it/tutorials/dynamic-page-templates-in-wordpress-part-3--cms-28514

13 - Alcuni temi personalizzabili

"Ed ora si fa come dico io..."

Elenchiamo qui alcuni temi relativamente semplici da personalizzare, non sono un dictat, sono solo un punto di partenza arbitrario, ne troverete altri e migliori:

- Shapely

- The Bridge
- responsive : https://wordpress.org/themes/responsive
- Customiz : https://wordpress.org/themes/customizr
- il TwentySeventeen!!! Sì quello di default

14 - alcuni plugin Utili

"il coltellino svizzero di WP..."

- Jetpack by WordPress.com (o simili, aggiungono features a wordpress, un casino di features … tra cui anche la condivisione tra social channels)
- Swifty Page Manager per gestire alberature complesse
- Wordfence Security (o simili, plugin che si occupano della sicurezza del sito)
- WordPress SEO (o plugin simili che si occupano della Search Engine Optimization)
- Clone Posts (o plugin simili che copiano post o pagine, in modo da tagliare i tempi di produzione)
- Custom Post Types (https://www.evernote.com/shard/s16/sh/cece246e-1725-4f1a-91ba-cd08cda6e946/f8f3aad79ea18e2b8793868b519815bb)
- All-in-One WP Migration (per migrare il proprio sito dal Web a locale e viceversa)
- … poi dipende cosa devi fare …

15 - Custom Post Types

Con essi potrete creare dei form di gestione di contenuti particolari (es. una lista di libri), potrete anche relazionarvi a tabelle presenti nel DB di wordpress o ad altre tabelle.

Un plugin molto potente:

- https://wordpress.org/plugins/custom-post-type-ui/
- e relativo tutorial https://www.fogliata.net/custom-post-type-wordpress/

E per chi vuole sporcarsi le mani...

ecco la teoria dei custom post types:
http://codex.wordpress.org/Post_Types

Questo invece è uno strumento online per creare un custom Post Types con la funzione register_post_type() : http://generatewp.com/post-type/

16 - Tassonomie

http://codex.wordpress.org/it:Tassonomie

Altre risorse utili
http://www.kevinleary.net/wordpress-dashicons-list-custom-post-type-icons/

17 - I CSS (Cascading Stile Sheets in italiano Fogli di stile)

"Separati … si ma in casa"

Sono regole che permettono di definire gli aspetti di layout di una pagina.
Con essi si sancisce la separazione tra layout (controllato con i CSS) e contenuto (costruito con l'HTML).
Separati… si ma in casa, perché un foglio di stile senza un html su cui applicarlo non avrebbe senso.

* .... *

.testobianco {color: white;}
.altraclasse {font-weight: bold;}

avrei potuto anche accodarle sommandole (l'applicazione avverrà da sinistra verso destra, prima verrà applicata .testobianco poi .altraclasse)

.testobianco.altraclasse{font-weight: bold;color: white;}

Che vantaggi ho ad utilizzare i css?

Se ben usati posso

- concentrare il controllo del layout di un sito di molte pagine in un unico file.
- adattare il layout in base al dispositivo (media) utilizzato
- semplificare l'html prodotto
- avere a disposizione moltissime opzioni di stile che l'html non ha

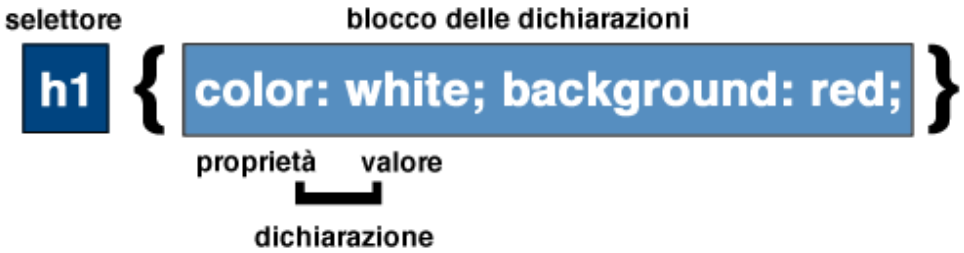
Come si scrivono i css?

ogni stile viene imposto tramite regole, voi scriverete delle regole.
le regole sono così composte:
selettore (che selezione cosa deve essere stilizzato) e una o più dichiarazioni, la dichiarazione è composta a sua volta da una proprietà e dal valore corrispondente.

```
selettore{
   proprietà: valore;
   proprietà: valore;
   proprietà: valore;
}
```

un esempio vero:

h1{color:white; background:red;}



- h1 è il selettore (il tag h1 in questo caso)
- { } aprono e chiudono il blocco delle dichiarazioni
- color e background sono le proprietà che state modificando
- white e red i valori delle rispettive proprietà

I Selettori

- **di tipo** (gli elementi html, ad es p, h1, ul…)
- **#id e .classi** (marcatori arbitrari che servono per identificare uno o più elementi html, nel caso degli id (identificati dal cancelletto) essi devono essere veramente univoci nella pagina html (ci deve essere solo un blocco con quell'id nella pagina html, nel foglio di stile potete scrivere quante regole volete che coinvolgano un id), es:

#menu{ width:40%;}

#menu a {color:blue;}

.voceMenu{line-height:120%;}

.voceMenu{line-height:120%;}

<ul id="menu">

*<li class="voceMenu">Voce menu 1
*

*<li class="voceMenu">Voce menu 2
*

*
*

- **di relazione**
  - selettore di discendenti (il più diffuso) *#contenitore p {color: white;}*
  - selettori di figli (>)
  - selettori di adiacenti (+)
  - selettori di fratelli (~)
- **pseudo classi**
  definiscono lo stato di un elemento
  a:hover{background-color:#fff;color:#000;}
- **pseudo-elementi**
  vengono utilizzati per lo stile di parti specifiche di un elemento
  ::after, ::first-letter

se volete approfondire
http://www.html.it/pag/42443/i-selettori-combinatori-o-di-relazione/

17.1 Ma come ce li metto gli stili in una pagina web?

I css posso essere :

- scritti inline attraverso la proprietà style del tag
- Interni
- esterni

rispetto ad una pagina web.

il modo migliore è l'ultimo, perchè?

Stili inline

Nel primo metodo ogni qualvolta avremo bisogno di stilizzare una pagina dovremo intervenire modificando la proprietà style di ogni singolo tag che dovrà essere stilizzato, per ogni singola pagina che compone il nostro sito.
Le informazioni di layout restano così mischiate al codice strutturale rendendo faticosa la lettura e l'interpretazione del codice.

```
<p style="color:#000; line-height:105%; padding-bottom:.5em;">Testo del paragrafo</p>
```

Stili interni (embedded)

Con il secondo metodo le cose migliorano leggermente, gli stili vengono raggruppati nella sezione <head></head> della pagina web, si utilizzeranno selettori di tipo tag, classe o id, ma anche così se dovessimo cambiare lo stile di un elemento (volendo restare uniformi) dovremmo intervenire su ogni singola pagina del nostro sito.
Le informazioni di formattazione sono ancora all'interno della nostra pagina web.

```
<head>
    <style>
    body {
        background-color: linen;
    }
    h1 {
        color: maroon;
        margin-left: 40px;
    }
</style>
</head>
```

Stili esterni (collegati)

L'ultimo metodo **è il più versatile**, tutte le pagine del nostro sito possono puntare ad un file esterno contenente le regole di formattazione.

Cambiando una regola in esso questa verrà riportata in tutte le pagine collegate ad esso (potenzialmente l'intero sito). Più di un file può essere collegato ad una o più pagine.
https://www.html.it/pag/16048/i-percorsi-assoluti-e-relativi/

<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>

@import

Non è una parolaccia... @import
E' una regola di importazione, tramite essa è possibile importare altri fogli di stile in un foglio di stile.
Wordpress lo usa sistematicamente quindi dovete sapere a che serve, il tipico esempio è il foglio di stile di un child theme.

/
Theme Name: twentyfourteen-child
Template: twentyfourteen
Author: Child Theme Configurator by Lilaea Media
Version: 1.0
Updated: 2014-11-29 11:57:37
**/*
@charset "UTF-8";
@import url('../twentyfourteen/style.css');

Approfondimento: http://www.html.it/pag/14211/inserire-i-fogli-di-stile-in-un-documento/

17.2 Ereditarietà e Cascata

L'ereditarietà e la cascata sono i due concetti cardine della teoria dei CSS

Ereditarietà

E' legata alle proprietà degli elementi HTML, per cui le (alcune) proprietà del padre vengono passate agli elementi figli.

La cascata

Spiega come le dichiarazioni css vengono applicate al documento, sopratutto in caso di conflitto tra esse.
Comprende tre regole

- importanza
- specificità
- posizione nel codice

la somma di tutte queste regole è la "Cascata" o "Cascade" da cui i fogli di stile prendono il nome.

"Se due dichiarazioni hanno la stessa importanza verrà usata la specificità per definire quale prevarrà sull'altra, se la specificità è la medesima allora verrà utilizzata la posizione nel codice (quella dichiarata dopo vincerà)."

Importanza

L'importanza di una dichiarazione CSS dipende da dove viene specificata.
L'eventuale conflitto tra dichiarazioni verrà risolto nel seguente ordine (dal meno importante al più importante):

- Fogli di stile dello User agent
- Dichiarazioni normali nei fogli di stile utente
- Dichiarazioni normali nei fogli di stile dell'autore (voi)
- Dichiarazioni !important in fogli di stile dell'autore
- Dichiarazioni !important nei fogli di stile dell' utente

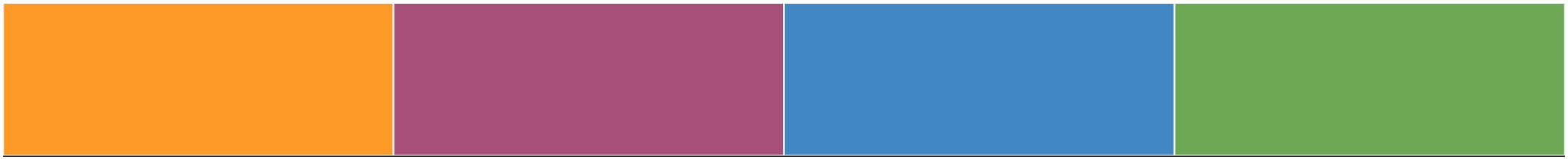
Perché le regole **!important** dell'utente possono, a differenza delle normali, sovrascrivere le regole dell'autore?
Per dare una possibilità all'utente di personalizzare il rendering degli elementi html secondo le proprie esigenze. Tipico è il caso di ipovedenti che variano dimensioni dei font, colori dei font e dello sfondo per rendere leggibili per loro stessi la pagina che avete stilizzato.

Alcune regole (poche, mi raccomando) possono essere definite !important, esse come potete vedere vinceranno sempre su tutte.

```
h1 {
color: maroon;
margin-left: 40px; !important
}
```

Specificità

Per specificità si intende la priorità che una regola ha sulle altre che si riferiscono allo stesso selettore. La proprietà con valore di specificità maggiore avrà la precedenza sulle altre;

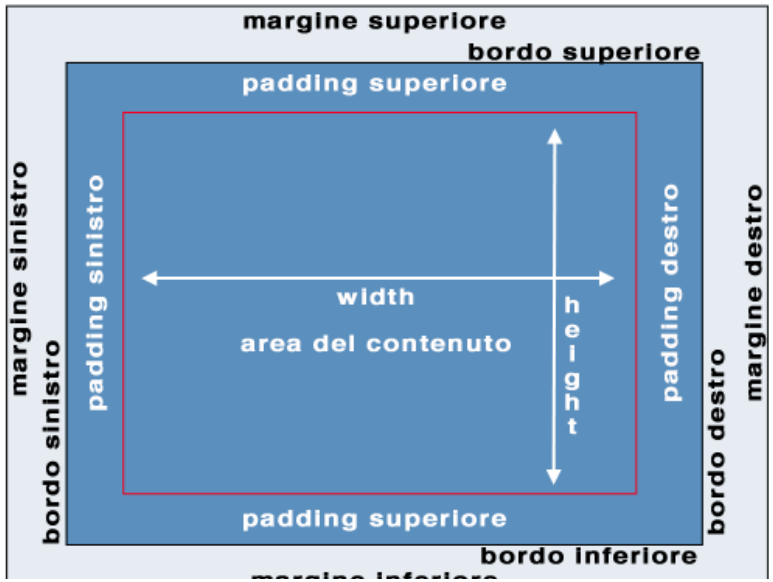


Posizione nel Codice

nel caso in cui due o più regole, come in precedenza, abbiamo tale valore uguale, verrà assegnata l'ultima inserita nell'ordine a cascata.

17.3 Il box Model

il box model appartiene in realtà all'html, non ai css, è il modello che rappresenta il modo in cui vengono applicate le regole di stile agli elementi di blocco html



Se volete approfondire:
http://www.html.it/pag/14210/classificazione-di-elementi-e-tag-xhtml/ (elementi di blocco e inline)
http://www.html.it/guide/guida-css-di-base/

17.4 Posizionamento di un elemento

Fate attenzione al posizionamento di elementi, proprietà **position**, è una delle proprietà più potenti che avrete a disposizione perchè vi permetterà di posizionare un elemento dove vorrete.
Avremo quattro possibilità

- **static** (il default di ogni elemento) il contenuto è posizionato nella posizione che occupa nel flusso del documento
- **fixed**, l'elemento verrà posizionato nella posizione x ed y indicata rispetto al alla viewport del browser, non si sposterà nemmeno durante lo scorrimento della pagina
- **relative**, l'elemento verrà posizionato in base alla sua posizione che normalmente occupa nel flusso del documento (es. top:10px; sposterà l'elemento di 10 px più in basso rispetto alla sua normale posizione)
- **absolute**, riposizionerà l'elemento in base al primo elemento padre che avrà una posizione diversa da static (è sufficiente dichiarare nel padre position:relative; e questo praticamente non cambierà il comportamento del contenitore padre... se è quello che volete che succeda ovviamente)
  Se non vengono trovati elementi padri, a cui sia stata variata la position di default, verrà considerato l'elemento html.
  L'elemento viene staccato dal flusso della pagina e l'elemento che lo contiene collassa (si accorcia in altezza) dello spazio che occupava l'elemento con position absolute.

es.
header{
position:relative;
border:1px solid red;
height:180px;

```
}
#logotipo{
position:absolute;
right:0px;
top:0px;
}

<header>
<div id="logo">
<a href="#"><img src="/img/logo.jpg" alt="Politecnico di Milano"></a>
</div>
<div id="logotipo">
<a href="/">Dipartimento di Matematica</a>
</div>
<nav id="menuPrincipale">
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Ricerca</a></li>
<li><a href="#">Didattica</a></li>
<li><a href="#">Personale</a></li>
<li><a href="#">Laboratori</a></li>
<li><a href="#">Biblioteca</a></li>
</ul>
</nav>
</header>
```

il logotipo si posizionerà in base a header

Se desiderate approfondire e fare prove (try yourself):

http://www.w3schools.com/css/css_positioning.asp

17.5 - Pseudo classi e Pseudo elementi

Pseudo Classi

definiscono speciali stati di un elemento html (il più conosciuto è :hover)
http://www.w3schools.com/css/css_pseudo_classes.asp

Pseudo elementi

definiscono gli stili di una parte specifica di un elemento html (ad esempio ::before)
http://www.w3schools.com/css/css_pseudo_elements.asp

17.6 - Media Queries

"Chiedi e ti sará dato"

Cosa sono le media queries? Sono una caratteristica presente in CSS3 che permette di definire quando applicare alcune regole css

```
/* mostra l'immagine più grande finché la larghezza minima della viewport è uguale o maggiore 1200px, cioè da 1200px in su
si vedrà' l'immagineVeramenteGGrande, al di sotto no */

@media screen and (min-width: 1200px) {
#header-image {
    background-image:url('immagineVeramenteGGrande.gif');
    }
}
```

Riferimenti:

- http://cssmediaqueries.com/overview.html
- http://css-tricks.com/snippets/css/media-queries-for-standard-devices/

Attenzione! Per funzionare correttamente le nostre media queries hanno bisogno di una dichiarazione viewport nei tag meta delle pagine del nostro sito (tranquilli chi disegna temi di wordpress lo sa), un esempio:

<meta name ="viewport" content ="width=device-width">

E ora al lavoro...

18 - Ma tu... chi sei?

Michele Martinello

dal 2005 sviluppatore web presso l'area Public Engagement & Communication, Tools and Content Management unit del Politecnico di Milano, dove partecipa alla progettazione e realizzazione di siti afferenti l'Ateneo ed opera come formatore per corsi interni.
Conosce ed utilizza quotidianamente il linguaggio PHP, typoscript, javascript e Java, esperto nell'uso di HTML e CSS.
Dal maggio 2011 presta la sua opera anche come formatore esterno presso Fidia s.r.l. in corsi orientati al web.
Opera, progetta ed implementa soluzioni basate sui CMS Typo3 e Wordpress, convinto assertore dell'usabilità come opportunità e fiero membro del MilanUxBookClub, a tempo perso imperversa tra i fornelli attentando al girovita di parenti, amici e colleghi.
In preda ai sensi di "polpa" dal 2015 si dedica quasi quotidianamente alla pratica del "CrossFit".

Frase preferita (dopo "quando si mangia?")

"L'errore non è un problema, l'errore è una opportunità!"

Contatti

michele.martinello@icloud.com

+39 347 2738963