

**UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA**

Anno Accademico 2008/2009



## Assistive Technology

Informatizzando la Disabilità

Luca Lischetti

Prof. Claudio Gentile

**FACOLTÀ DI SCIENZE MM.FF.NN.  
CORSO DI LAUREA IN INFORMATICA**

## Premessa

L'argomento affrontato sarà l'Assistive Technology relazionata al campo dell'Informatica. Questa tesi è basata in parte sull'esperienza maturata durante il periodo di internship presso l'azienda Assistive Solutions, in merito al progetto Erasmus Placement della Comunità Europea, e in parte su ricerche e approfondimenti personali.

Durante l'esperienza di studio/lavoro all'estero sono venuto a contatto con una realtà totalmente nuova per me, quella dell'informatica per disabili. Questo argomento mi ha affascinato a tal punto da proporlo al professore e tutor universitario Claudio Gentile come argomento per la mia tesi di laurea.

Questa tesi si suddivide in due parti, una prettamente teorica, nella quale si tenta di dare una visione di insieme su cos'è e come opera l'Assistive Technology in campo informatico, e una che analizza in modo sintetico ciò che ho sviluppato personalmente attraverso questa esperienza. Ci tengo a precisare che durante questo periodo di stage ho potuto creare e contribuire allo sviluppo di altri applicativi che, in quanto non attinenti al campo dell'Assistive Technology, non hanno trovato spazio in questo elaborato.

Ho tentato di utilizzare sia nella parte teorica che nella parte pratica un linguaggio più semplice possibile, coadiuvando spesso le spiegazioni con riferimenti esterni di chiarimento o approfondimento, per consentire la lettura a chiunque ne fosse interessato.

Il fine di questo opera vuole essere principalmente quello di far conoscere l'Assistive Technology, tecnologia fino ad ora per lo più oscura, e, perché no, anche una sorta di linea guida per chi decidesse di voler produrre contenuti accessibili e conformi alle specifiche relazionate a questo campo.

Colgo l'occasione per ringraziare chiunque abbia contribuito, direttamente o indirettamente, alla stesura di questa tesi. La Comunità Europea per aver consentito di arricchirmi sotto il punto di vista accademico, professionale e personale, prima permettendomi di studiare in Spagna e poi di lavorare nel Regno Unito. La mia famiglia che ha sempre compreso e rispettato le mie esigenze di vivere all'estero. Tutti i miei amici (su tutti Paolo, Federico ed Erika), i compagni di viaggio e i coinquilini che mi hanno accompagnato e sostenuto durante questi anni. Alan e David per avermi introdotto nel mondo dell'Assistive Technology. La città e soprattutto la gente di Londra per avermi insegnato e arricchito di molte cose.

*Londra, 14 marzo 2010*

## Indice

|  |     |
|--|-----|
| Prefazione .....   | 2   |
| <b>1. Introduzione all'Assistive Technology</b>                    |     |
| 1.1 Definizione .....  | 4   |
| 1.2 Storia .....   | 5   |
| 1.3 Classificazioni .....  | 7   |
| <b>2. Accessibilità</b>  |     |
| 2.1 Definizione .....  | 9   |
| 2.2 Tecnologie .....   | 10  |
| 2.2.1 Hardware .....   | 10  |
| 2.2.2 Software .....   | 21  |
| <b>3. Accessibilità Web</b>  |     |
| 3.1 Definizione .....  | 28  |
| 3.2 Il World Wide Web Consortium (W3C) .....                       | 30  |
| 3.3 Web Accessibility Initiative (WAI) .....                       | 32  |
| 3.3.1 WAI - Accessible Rich Internet Applications (WAI-ARIA) ..... | 35  |
| 3.3.2 Web Content Accessibility Guidelines (WCAG) .....            | 36  |
| 3.3.3 Ulteriore documentazione rilasciate dal WAI .....            | 41  |
| 3.4 Migliorare ulteriormente l'accessibilità .....                 | 42  |
| 3.4.1 Validazione del codice .....                                 | 42  |
| 3.4.2 Access Key .....   | 43  |
| 3.4.3 Access Bar - Barra dell'accessibilità .....                  | 45  |
| 3.4.4 Accessibilità via browser .....                              | 47  |
| 3.5 Accessibilità Web - Conclusione .....                          | 49  |
| 3.6 Legislazione Italiana .....                                    | 50  |
| <b>4. Succed</b>   |     |
| 4.1 Introduzione .....   | 58  |
| 4.2 Caratteristiche del software .....                             | 62  |
| 4.2.1 Caratteristiche tecniche del software .....                  | 64  |
| 4.3 Struttura interna del software .....                           | 66  |
| 4.4 Il layout .....  | 68  |
| 4.4.1 La struttura .....   | 68  |
| 4.4.2 Il codice .....  | 70  |
| 4.5 Access Bar .....   | 75  |
| 4.5.1 Le funzioni JavaScript .....                                 | 76  |
| 4.5.2 Esempi .....   | 86  |
| 4.5.3 Funzioni aggiuntive .....                                    | 88  |
| 4.6 Embed video .....  | 92  |
| 4.7 Rounded corners .....  | 94  |
| <b>Appendici</b>   |     |
| A. Il Braille .....  | 99  |
| B. ISO 9999/2007 .....   | 100 |
| C. Legge Stanca .....  | 101 |

## **1. Introduzione all'Assistive Technology**

### **1.1 Definizione**

Con Assistive Technology (AT) si intende qualsiasi tecnologia, sistema, oggetto o parte di esso che viene usato per accrescere, mantenere o migliorare le capacità di un individuo affetto da qualsivoglia tipo di disabilità e, indirettamente, di qualsiasi persona. In modo più generale qualsiasi tecnologia che consente a qualcuno di realizzare qualcosa che normalmente non potrebbe realizzare può essere identificata anche come Assistive Technology.

L'AT è una tecnologia usata da persone diversamente abili per usufruire di servizi che altrimenti risulterebbero difficili o impossibili per loro. Questa tecnologia è caratterizzata da un insieme di tecniche e dispositivi particolari creati in funzione di ogni tipo di handicap. Per quanto riguarda i dispositivi si possono individuare diverse categorie, da quelli che operano come ausilio motorio (per esempio stampelle o sedie a rotelle), a quelli che operano come ausilio didattico (come ad esempio autocorrettori o gestori di flow-chart), fino ad arrivare a veri e propri dispositivi "informatici", che aiutano la persona ad interfacciarsi sia ad un computer, sia in modo più generale a tutte le diverse tecnologie informatiche.

Trattando in modo più approfondito i cosiddetti dispositivi informatici si può effettuare un'ulteriore divisione in due sottogruppi molto distinti, ossia i dispositivi hardware e quelli software. Un semplice esempio di hardware può essere una tastiera con tasti molto larghi o uno speciale mouse per facilitare l'uso di queste periferiche a persone con difficoltà motorie o articolari alle mani. Per quanto riguarda invece il software si può pensare ad un programma in grado di leggere parti di testo a schermo, generando una voce computerizzata, per consentire la lettura di documenti o pagine web da parte di un utente con difficoltà visive o addirittura non vedente.

L'Assistive Technology è sicuramente un campo molto complesso e con molte sfaccettature anche se alcune volte la risoluzione dei problemi può sembrare relativamente semplice.

Ad una prima semplicistica analisi l'Assistive Technology potrebbe sembrare una tecnologia rivolta solamente a persone affette da disturbi, effettivamente il suo scopo è proprio questo, ma in realtà non è solo così. Tutti gli utilizzatori, dei più disparati dispositivi, possono infatti trarre benefici da questa tecnologia che tenta di rendere più accessibili, e di conseguenza più semplici da usare, un numero sempre più elevato di strumentazioni.

## 1.2 Storia

Da un punto di vista storico è difficile delineare una linea temporale che caratterizzi l'evoluzione dell'Assistive Technology. Le prime tracce documentate risalgono agli inizi del diciannovesimo secolo.

Nel 1808 un tipografo italiano Pellegrino Turri inventò una prima rudimentale macchina da scrivere per una sua amica affetta da cecità. Fino ad allora ci furono sporadici tentativi di creare tecnologie atte ad aiutare persone diversamente abili, possiamo infatti trovare ad esempio sedie a rotelle, ma in questo caso si tratta di un dispositivo per agevolare funzioni base del corpo umano e non, come nel caso della macchina da scrivere, di funzioni avanzate quali appunto la scrittura.

Un importante passo avanti fu fatto nell'anno 1821 quando il francese Louis Braille inventò il famoso metodo di scrittura e lettura per ciechi, *Braille*<sup>1</sup>, che da lui stesso ne prende il nome. Questa invenzione fu una pietra miliare nell'ambito dell'Assistive Technology per persone non vedenti, in quanto in questo modo gli si è permesso di scrivere in un modo indipendente e senza l'ausilio di collaboratori.

La scoperta, e il successivo evolversi, delle telecomunicazioni (partendo dal telefono, fino ad arrivare a internet) ha svolto un ruolo cruciale per lo sviluppo dell'Assistive Technology e questo sviluppo è destinato a crescere a livello esponenziale. Le telecomunicazioni su larga scala hanno permesso la creazione di strumenti ausiliari alla sicurezza e l'incolumità di alcuni tipi di persone, definiti soggetti a rischio. Un chiaro esempio possono essere tutte quelle apparecchiature che operano in teleassistenza, mettendo se necessario in comunicazione la persona bisognosa di aiuto con il relativo ente referente.

Arrivando a tempi più recenti, più precisamente partendo dalle ultime due decadi del secolo scorso fino ad arrivare ai giorni nostri. Ora che l'utilizzo di computer e calcolatori si sta diffondendo sempre più capillarmente e i servizi erogati tramite questa nuova tecnologia aumentano giorno per giorno, l'accessibilità a tale mezzi diventa un punto fondamentale per l'integrazione di persone diversamente abili all'interno della società. In questi anni l'ideazione e la creazione di strumentazioni che si prefiggono come scopo l'interazione tra computer e persone diversamente abili ha subito un'enorme evoluzione. Conseguentemente anche le istituzioni si stanno muovendo di pari passo, creando a loro volta norme e leggi di salvaguardia e tutela di persone affette da problemi.

Una prima definizione formale di cos'è e di cosa di occupa esattamente l'Assistive Technology lo si può trovare in *"the Technology - Related Assistenze for Individuals with Disabilities Act"* del 1988 conosciuto anche come *The Tech Act*. In questo documento si prendono in esame gli scopi, le applicazioni e le norme relative all'Assistive Technology. Questo atto fu successivamente emendato nel 1994 dal governo degli Stati Uniti. Nel 1998 è stato poi rielaborato e rimpiazzato dal *"Assistive Technology Act"*<sup>2</sup> del 1998 (AT Act), senza che però la sua struttura base fosse stata modificata.

---

<sup>1</sup> Per un approfondimento sul Braille si rimanda alla lettura dell'Appendice A ad esso dedicata.

<sup>2</sup> <http://www.section508.gov/docs/AT1998.html> - Testo completo dell'Atto Governativo

Conseguentemente anche il mondo del Web ha iniziato a muoversi nella medesima direzione. Nel 1999 il *World Wide Web Consortium*<sup>3</sup> (W3C) ha infatti rilasciato la specifica *Web Content Accessibility Guidelines*<sup>4</sup> (WCAG), in questo documento tecnico viene specificato come gli sviluppatori di contenuti web dovrebbero operare per rendere accessibili le loro pagine al maggior numero di utenti possibili. Questa parte verrà approfondita in seguito quando si prenderà in esame l'accessibilità Web.

Non è semplice tentare di fare previsioni sugli sviluppi futuri dell'Assistive Technology, è ovvio che più passa il tempo più questa "relativamente" nuova tecnologia si affina e si espande e, credo, sia destinata ad evolversi fin quando le differenze di utilizzo, tra disabili e non, si siano finalmente annullate.

---

<sup>3</sup> <http://www.w3.org/> - pagina ufficiale del World Wide Web Consortium

<sup>4</sup> <http://www.w3.org/TR/WCAG20/> - testo completo della specifica WCAG 2.0

## 1.3 Classificazioni

### 1.3.1 Panoramica disabilità

Una volta introdotto il concetto di Assistive Technology è giunto il momento di prendere in considerazione su effettivamente quali classi di disabilità essa va ad operare. Si possono dividere tre grosse branche di invalidità: *disabilità cognitiva, motoria e sensoriale*.

Chiaramente ognuna di queste categorie necessita trattamenti adeguati e mirati e, anche all'interno di esse, si possono trovare notevoli differenziazioni che ora verrano esaminate più da vicino.

*Nota: In questo paragrafo non andrò ad analizzare nel dettaglio le varie tecnologie a disposizione per ogni genere di disabilità, mi limiterò a semplici esempi per facilitare la comprensione dei disturbi presi in esame.*

#### A. Disabilità Cognitive

*Questa categoria è contraddistinta da disabilità intellettive. Esse possono manifestarsi come semplici difficoltà di apprendimento, come ad esempio la dislessia, oppure come più gravi disturbi di ritardo mentale, che possono assumere forme più o meno serie. La medicina con il tempo ha creato veri e propri percorsi terapeutici per cercare di migliorare, e in alcuni casi risolvere, queste tipologie di problemi. L'Assistive Technology può venire particolarmente in aiuto per questo genere di patologie, le tecnologie utilizzate in questo settore sono quasi esclusivamente di livello software. Infatti esistono programmi in grado di assistere la persona durante la digitazione di testi a computer, questi aiuti possono spaziare dal semplice correttore ortografico/grammaticale a più complessi meccanismi di rilettura dei testi.*

#### B. Disabilità Motorie

*Sotto disabilità motorie sono raggruppati tutti quei disturbi collegati al movimento del corpo della persona invalida, essi vanno dai problemi di deambulazione a difficoltà più capillari che riguardano i singoli arti o parti del corpo. Questo genere di disabilità può essere congenito oppure conseguente a qualche tipo di trauma (es. traumi alla spina dorsale, o paralisi causate da ictus). L'Assistive Technology fornisce strumenti che consentono l'ausilio nella deambulazione di persone inferme, ad esempio carrozzine e scooter elettrici sono alla base di questo tipo di dispositivi. Per quanto riguarda difficoltà motorie alle mani il discorso si fa più complicato e sfaccettato in quanto ci si può trovare di fronte ad una maggior casistica. In linea di massima esistono dispositivi, adattabili alle varie esigenze, che consentono l'accesso a terminali senza l'uso delle mani, detti hands-free, un esempio può essere un mouse comandabile con i piedi (footmouse). Esistono inoltre altre tecnologie che consentono l'accesso ai computer senza l'uso delle mani. Le due tecnologie più conosciute sono chiaramente il riconoscimento vocale (speech-recognition) e il meno noto eye-gaze, che consente di interagire con un computer mediante il movimento degli occhi.*

#### C. Disabilità Sensoriali

*Con disabilità sensoriali si intendono principalmente disturbi legati alla vista e all'udito. Quindi fanno parte di questa categoria persone con difficoltà visive, fino ad arrivare alla cecità, e con difficoltà ad udire, nel caso massimo sordità. L'Assistive Technology si è sviluppata molto in questo settore, infatti possiamo trovare vecchie tecnologie quali audiolibri o orologi parlanti, oppure strumenti più evoluti quali invece software di sintesi vocale, che convertono testi computerizzati in voce. Per individui con solamente problemi di vista possono essere invece di aiuto grandi display digitali o tastiere con grossi tasti. Esistono anche soluzioni software per questo genere di problemi detti display magnifer,*

*essi consentono di ingrandire parti di schermo un numero esponenziale di volte. Per persone sordi invece l'attenzione si sposta sulle notifiche visuali, ad esempio ponendo in determinati punti della casa segnali luminosi nel caso qualcuno suonasse il campanello. Per quanto riguarda l'interazione informatica solitamente la parte audio non è molto importante quando si opera con un computer, infatti solitamente un dispositivo produce solamente suoni di feedback, in questo caso è relativamente semplice sostituirle con notifiche a video. Per le persone non completamente sordi invece possono essere di grande aiuto amplificatori portatili. Questo campo è notevolmente in crescita, negli ultimi anni sono stati creati amplificatori wireless, questi dispositivi possono essere particolarmente utili in teatri oppure in ambienti scolastici.*

### **1.3.2 Classificazione Assistive Technology**

Prima di iniziare ad analizzare le singole tecniche e tecnologia in cui si snoda l'Assistive Technology è bene cercare di dare una classificazione dei vari campi in cui essa si suddivide.

Ci sono svariati modi per suddividere le varie disabilità e di conseguenza per categorizzare i vari tipi di Assistive Technology. La classificazione più valida è quella fornita dalla specifica **ISO 9999:2007<sup>5</sup> Classification of Technical Aids<sup>6</sup>** e viene sviluppata in forma product-oriented.

Vengono delineate dieci tipologie di *Assistive Device*:

- *Ausili per trattamenti sanitari personali*
- *Ausili per l'esercizio di abilità*
- *Ortesi e protesi*
- *Ausili per la cura e la protezione personale*
- *Ausili per la cura della casa*
- *Mobilia e adattamenti per la casa o per altri ambienti*
- ***Ausilio per comunicazioni, informazione e segnalazione***
- *Ausili per manovrare oggetti o dispositivi*
- *Adattamenti dell'ambiente, utensili e macchine*
- *Ausili per le attività di tempo libero*

Questa classificazione, seppur perimetrale rispetto all'argomento principale di questa tesi, serve a fornire un panorama completo di tutto quello che si intende per Tecnologia Assistenziale.

---

<sup>5</sup> [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=38894](http://www.iso.org/iso/catalogue_detail.htm?csnumber=38894) - Pagina ufficiale della specifica ISO

<sup>6</sup> Una classificazione più approfondita si può trovare in appendice. Appendice B - Classificazione Assistive Device

## **2. Accessibilità**

### **2.1 Definizione**

*«accessibilità»: la capacità dei sistemi informatici, nelle forme e nei limiti consentiti dalle conoscenze tecnologiche, di erogare servizi e fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari. (D.M. 8 Luglio 2005)<sup>7</sup>*

*«tecniche assistive»: gli strumenti e le soluzioni tecniche, hardware e software, che permettono alla persona disabile, superando o riducendo le condizioni di svantaggio, di accedere alle informazioni e ai servizi erogati dai sistemi informatici. (D.M. 8 Luglio 2005)*

La legge Italiana, mediante la *Legge Stanca*<sup>8</sup> del 9 gennaio 2004, successivamente pubblicata nella *Gazzetta Ufficiale* il 17 gennaio 2004 e resa operativa col *Decreto Ministeriale dell'8 Luglio 2005*, da una definizione sia di accessibilità, sia di tecnologie assistive (traduzione italiana di Assistive Technology). C'è da sottolineare che questa definizione di Assistive Technology è riduttiva in quanto viene solamente preso in considerazione il ramo informatico di questa tecnologia.

Sostanzialmente l'accessibilità, intesa come capacità di un sistema, di un dispositivo o di una risorsa di essere utilizzabile dal maggior numero possibile di persone, è un campo della più vasta categoria della Assistive Technology. Come si può notare la definizione di accessibilità si rispecchia fedelmente in quella data nel capitolo scorso di Assistive Technology, ma la caratterizza adattandola al campo più specifico dell'informatica. A sua volta l'accessibilità può essere ulteriormente specificata, è il caso dell'*accessibilità Web*.

L'accessibilità Web è un particolare campo dell'accessibilità informatica, che si occupa solo ed esclusivamente di prendere in esame la fruibilità delle informazioni e la facilità di utilizzo di un sito web. Comunque questo argomento verrà trattato in maggior dettaglio in seguito.

---

<sup>7</sup> <http://www.governo.it/Presidenza/web/dm08072005.html> - Testo integrale del Decreto Ministeriale

<sup>8</sup> <http://www.parlamento.it/parlam/leggi/04004I.htm> - Testo integrale della Legge Stanca

## 2.2 Tecnologie

Lo scopo di questo capitolo è prendere in esame le principali tecnologie finora create per garantire un sempre maggiore livello di accessibilità dei servizi informatici.

Come già accennato in precedenza quando si parla di accessibilità informatica si deve tener conto dei due gruppi da cui essa è formata ovvero l'accessibilità hardware e quella software.

### 2.2.1 Hardware

Esistono varie tipologie di dispositivi AT, chiaramente differenziati in base a quale lacuna funzionale vanno a colmare nell'interazione di una persona diversamente abile con un computer. Chiaramente questi dispositivi possono essere combinati l'uno con l'altro per aumentarne le funzionalità oppure nel caso una persona soffrisse di più disabilità.

#### 2.2.1.1 Tastiera

La tastiera è un dispositivo che permette l'inserimento di dati all'interno di un computer mediante la pressione di uno o più tasti. Conseguentemente i problemi che si possono incontrare ad interfacciarsi con questo tipo di hardware sono legati alla vista e al movimento. Per una persona ipovedente può risultare difficile vedere o distinguere i vari tasti, mentre per una persona con difficoltà di movimento può risultare difficoltoso premere solamente i tasti desiderati. Per risolvere questo genere di problemi sono stati creati numerosi dispositivi che si prefiggono di risolvere sostituendo o riadattando le normali tastiere.

#### Tastiere XXL

Questo genere di tastiere è adatto a persone che soffrono di disturbi visivi. Sostanzialmente si tratta di una normale tastiera con la sola differenza che le lettere, stampate sui singoli tasti, ne occupano l'intera superficie, migliorandone decisamente il riconoscimento visivo.



*Tastiera Cherry XXL  
(fig. 1)*

#### Scudi

Gli scudi, anche chiamati "copri tastiera", sono dei dispositivi rimovibili che consentono, a chi ha problemi di controllo "fine" de movimento delle mani, di digitare con maggior cura e conseguentemente con un minor numero di errori di battitura. Per far questo l'utente può appoggiare l'intera mano sulla superficie dello scudo posizionato sulla sommità della tastiera e, infilando le dita in corrispondenza dei tasti desiderati, può



*Tastiera Cherry XXL con scudo  
(fig. 2)*

digitare quello che desidera prevenendo battiture involontarie. Solitamente sono di colore scuro, opposto a quello della tastiera, in modo da facilitare ulteriormente il riconoscimento dei tasti, in quanto li “incornicia”.

### Tastiere semplificate

Si tratta di tastiere con tasti di grande dimensione e generalmente dispongono di un numero inferiore di tasti, rispetto ai modelli canonici. Solitamente sono di un colore vivace per facilitarne la lettura. Possono essere alquanto utili a persone affette da ipovisione, oppure anche ad anziani che si avvicinano per la prima volta all'uso di un computer. Anche i bambini possono trarne giovamento dall'utilizzo, in quanto i tasti sono colorati in modo da raggruppare numeri, vocali, consonanti, punteggiatura e funzioni. Possono inoltre essere particolarmente adatte anche a persone affette da dislessia o più in generale adulti e bambini con lievi difficoltà di apprendimento.



Tastiera semplificata  
(Queenkey)  
(fig. 3)

### Tastiere espande

Sono tastiere generalmente di grande dimensione, nella *figura 4.b* si può notare la grandezza della tastiera in rapporto ad un mano, pensate per persone con gravi difficoltà motorie e di ipovisione. I tasti sono tondi con bordo incurvato, leggermente rientranti sul piano di appoggio. Come caratteristiche addizionali hanno la possibilità di controllo del puntatore del mouse, di regolazione della sensibilità dei vari tasti e della ripetizione ed infine il bloccaggio dei tasti di funzione. La funzione di controllo del mouse avviene mediante l'utilizzo di alcuni tasti su cui sono indicate anche le funzioni del mouse.



Tastiera espansa (Winking)  
(fig. 4.a)



Particolare tastiera (Winking)  
(fig. 4.b)

## Tastiere ridotte

Hanno le stesse funzionalità delle *tastiere espande*, l'unica differenza sta nelle dimensioni, in questo caso molto ridotte. Nella *figura 5.b* si può notare sempre la grandezza della tastiera in rapporto ad un mano. Sono utili a persone che hanno limitata forza e difficoltà nei movimenti ampi. I tasti, collocati molto vicini tra loro, sono a membrana. Anche queste possono sostituire le funzioni del mouse.



*Tastiera ridotta Winmini  
(fig. 5.a)*



*Particolare tastiera Winmini  
(fig. 5.b)*

## Tastiere a membrana programmabili

Integrano le normali periferiche di input aggiungendo nuovi tasti liberamente programmabili dall'utente. La programmazione consiste nell'assegnazione di funzioni o macro (combinazioni di tasti) anche personalizzabili per differenti tipi di programma. Forniscono un accesso facilitato al computer sul piano motorio, visivo e cognitivo. Inoltre è possibile impostare la sensibilità dei tasti, la ripetizione e altre funzioni utili agli utenti disabili e non solo.



*Tastiera a membrana programmabile  
X-keys<sup>9</sup> Desktop 20 tasti  
(fig. 6.a)*



*Tastiera a membrana programmabile  
X-keys Stick<sup>10</sup>  
(fig. 6.b)*

## Tastiere a video

Si tratta di programmi che consentono l'emulazione via software della tastiera hardware. Sono progettati per assistere principalmente persone con disabilità motorie. Il funzionamento è molto semplice, viene visualizzata a video una tastiera, in alcuni casi è anche personalizzabile in base ai vari programmi con la quale viene utilizzata. La

<sup>9</sup> <http://www.xkeys.com/xkeys/xkdesk.php> - pagina di riferimento della tastiera X-keys Desktop

<sup>10</sup> <http://www.xkeys.com/xkeys/xkstick.php> - pagina di riferimento della tastiera X-keys Stick

selezione dei tasti avviene mediante l'utilizzo del mouse, oppure tramite diversi tipi di sensori.



*Esempio di tastiera a schermo  
(fig. 7)*

### Tastiere programmabili

Sono formate da una base neutra da collegare al computer e da un variabile numero di schede che, mediante la base, comunicano i dati in input. Le schede generalmente sono suddivise per campi di interesse, ad esempio si va dalla scheda contenente le classiche lettere dell'alfabeto a quella specializzata per i movimenti del mouse (*fig. 8.b*) fino ad arrivare a quella specifica per le operazioni aritmetiche (*fig. 8.c*).

Oltre alle schede base prestampate è possibile crearne di nuove mediante appositi programmi software. Questi strumenti sono chiaramente adatti a chi soffre di disfunzioni motorie, ma possono essere anche di ausilio a normali utenti che cercano di semplificare lo svolgimento di operazioni ricorrenti.

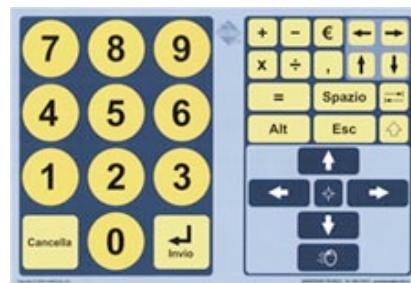


*Tastiera programmabile  
(Intellikeys)  
(fig. 8.a)*

Di seguito sono mostrati due esempi di schede prestampate (Intellikeys<sup>11</sup>):



*Tastiera programmabile (Intellikeys)  
Scheda movimenti del mouse  
(fig. 8.b)*



*Tastiera programmabile (Intellikeys)  
Scheda operazioni aritmetiche  
(fig. 8.c)*

<sup>11</sup> <http://www.intellitools.com/> - sito Web del produttore Intellitools

### 2.2.1.2 Mouse

Il mouse è un dispositivo in grado di inviare un input ad un computer in modo tale che ad un suo movimento ne corrisponda uno analogo di un indicatore sullo schermo detto puntatore. È inoltre dotato di uno o più tasti ai quali possono essere assegnate varie funzioni. Conseguentemente i problemi legati a questo genere di dispositivi sono esclusivamente di natura motoria. In questo caso l'AT si preoccupa di realizzare dispositivi per coloro che non riescono a gestire il cursore, o i tasti funzione, tramite un mouse tradizionale. Esistono principalmente due sotto categorie, i così detti sistemi di puntamento alternativo oppure gli emulatori di mouse per disabilità più gravi.

Esistono parecchi dispositivi atti a sostituire la figura del classico mouse per interagire con un computer. Di seguito verranno analizzati solo i dispositivi principali e non tutte le varie sottoclassi disponibili.

#### Trackpad

Il trackpad, a differenza della *trackball*, gestisce il movimento del cursore mediante il movimento di uno o più dita sulla sua superficie.

Comunemente utilizzati sui computer portatile possono essere anche installati su computer desktop. Il movimento necessario per lo spostamento del cursore può essere limitato al solo lavoro di un dito della mano, questa funzione risulta notevolmente di aiuto a chi affetto da problemi di origine motoria.

Anche i trackpad chiaramente sono muniti di tasti, anche addizionali, e personalizzabili.



Trackpad  
(Easycat)  
(fig. 1)

#### Trackball

Sono dispositivi che consentono di controllare il cursore del mouse compiendo minimi movimenti degli arti superiori. La rotazione del controller simula il movimento della sfera del mouse. Questo dispositivo è particolarmente adatto a chi soffre di problemi motori o impossibilitato a fare costanti movimenti di polso. Può essere corredata da tasti opzionali personalizzabili che consentono la memorizzazioni di funzioni e/o macro.



Trackball  
(fig. 2)

## Joystick

Chiamato anche in alcuni casi mouse ergonomico, sostituisce il movimento trascinatorio della mano sul mouse con un semplice movimento basculante di una cloche. Il movimento del cursore è chiaramente affidato al movimento della leva, mentre le funzioni di click sono sostituite da tasti posti sulla cloche (come in *figura 1.a*) oppure sulla base del joystick (come in *figura 1.b*)<sup>12</sup>. Questo tipo di dispositivi è particolarmente adatto per chi soffre della sindrome del tunnel carpale del polso.



*Joystick  
(Mouse ergonomico ANRI)  
(fig. 3.a)*



*Joystick  
(Roller Plus Joystick)  
(fig. 3.b)*

## Emulatori a soffio

Sono dispositivi progettati per emulare il movimento del mouse mediante un uso combinato di bocca e testa. Serrando il beccuccio fra le labbra è possibile muovere il cursore con i soli movimenti della testa. Le funzioni di click sinistro e destro, doppio clic, trascinamento vengono esercitate soffiando o aspirando aria nel beccuccio, chiaramente solo con un minimo sforzo respiratorio. Questi dispositivi, integrati da tastiere a video, consentono un totale controllo di un computer anche a persone completamente invalide negli arti superiori.



*Emulatore a soffio  
(Integramouse<sup>13</sup>)  
(fig. 4.a)*



*Particolare emulatore a soffio  
(Integramouse)  
(fig. 4.b)*

<sup>12</sup> <http://www.traxsys.com/> - Sito Web del produttore Traxsys

<sup>13</sup> <http://www.lifetool.at> - Sito Web del produttrice Lifetool

## Emulatori head-mouse

Sono dispositivi ottici che permettono di controllare i movimenti del cursore mediante lo spostamento della testa. Funzionano tramite un piccolo bersaglio senza fili che, posizionato sulla fronte o sugli occhiali dell'utilizzatore, segue e di conseguenza traccia i movimenti. Sostituisce il mouse per chi è impossibilitato ad usare le mani. I "click" del mouse sono simulati da prestabiliti movimenti persistenti oppure trasmessi mediante l'uso congiunto di altri tipi di *sensori*. Questo dispositivo abbinato all'uso di una *tastiera virtuale* può sostituire completamente l'utilizzo dei canonici dispositivi di input.



*Emulatore head-mouse (Tracker Pro)  
Particolare con sensore per occhiali  
(fig. 5.a)*

Un modello più avanzato di emulatore head-mouse, infatti non richiede l'ausilio di sensori:



*Emulatore head-mouse  
(Headmouse Extreme<sup>14</sup>)  
(fig. 5.b)*



*Particolare emulatore head-mouse  
(Headmouse Extreme)  
(fig. 5.c)*

Esistono versioni più avanzate che funzionano mediante il solo **movimento della pupilla**. Conseguentemente l'uso di un sensore è assolutamente obsoleto oltre che in alcuni casi invasivo. Con questo genere di dispositivi il "click" del mouse può essere attivato mediante il battito delle palpebre.

---

<sup>14</sup> <http://www.orin.com/access/headmouse/> - pagina di riferimento della emulatore head-mouse Tracker Pro

## **Touch-screen**

Sono tipologie di monitor con integrato un sistema che rende l'area del video sensibile al tocco, trasformando così il computer in un sistema a tocco integrato. Permettono di emulare le funzioni del mouse e gestire i programmi attraverso il semplice tocco di un dito o una stilografica sullo schermo. Questa modalità di puntamento è molto intuitiva e diretta ed è adatta per gli utenti che hanno difficoltà ad interfacciarsi con i tradizionali sistemi di puntamento. I monitor touch-screen consentono di effettuare tutte le operazioni possibili con un mouse tradizionale come il doppio click, il click destro e il trascinamento. Possono inoltre essere utilizzati come tavoletta grafica oppure anche come tastiera a schermo. Inoltre l'innovativa tecnologia detta "multi-touch" che consente, come dice la parola stessa, la gestione di più tocchi contemporaneamente, apre tutto un nuovo modo di pensare questi touch-screen, aumentando di gran lunga le azioni possibili.

## **Switch-Mouse**

Sono dispositivi che consentono di collegare tra loro *joystick* e *sensori*, combinando i quali si possono sostituire i normali dispositivi di input. Consente di usare diversi sistemi di input alternativo contemporaneamente, ad esempio è possibile regolare il movimento del cursore mediante l'utilizzo di soli tasti (fig. 6.a) oppure con *joystick* (come mostrato in fig. 6.b).



*Switch-Mouse (Mouse Mover)  
con sensori collegati  
(fig. 6.a)*



*Switch-Mouse (Mouse Mover)  
con joystick collegato  
(fig. 6.b)*

### 2.2.1.3 Sensori

Sono dispositivi di tipo on/off che consentono di catturare in modo efficace e al tempo stesso più semplice possibile i movimenti volontari di un disabile motorio, anche grave. Funzionano come interruttori e, se configurati adeguatamente, consentono di controllare dispositivi hardware e/o software in modo tale da poter essere una valida alternativa ai canonici dispositivi di input.

#### Sensori a pressione

Sono i più semplici e comuni, la funzione di input si basa sulla pressione appunto della superficie del sensore. Possono essercene di vari tipi in base alle esigenze ed ai problemi degli utenti che devono utilizzarli. Oltre al modello base (*fig. 1.a*), caratterizzato da una superficie piana solitamente attivabile grazie alla pressione della mano, si possono trovare *sensori da piede* (*fig. 1.b*), caratterizzati da una notevole robustezza; sensori da viso, al contrario di quelli da piede sono caratterizzati da una superficie morbida e un alto livello di sensibilità; sensori da mano (*fig. 1.c*), molto piccoli e caratterizzati da la presenza di più tasti da selezionare con l'uso delle dita.

Soltanmente questo tipo di sensori ha la possibilità di regolare la forza da applicare per l'attivazione. Inoltre, in alcuni modelli, è presente un sistema di feedback che può essere sia tattile che sonoro.



*Sensore a pressione generico  
(Sensore Square Pad)  
(fig. 1.a)*



*Sensore a pedale  
(fig. 1.b)*



*Sensore da mano  
(Sensore Penta)  
(fig. 1.c)*

#### Sensori Leaf (a petalo)

I sensori a petalo per la loro struttura e per il loro funzionamento sono i più adatti per essere utilizzati mediante la testa. Sono particolarmente adatti per essere posizionati sia dietro la testa che a lato della guancia. Questo tipo di sensori è composto da un asta rigida, solitamente di materiale metallico, alla quale estremità è fissato un disco di spugna, che ne aumenta la sensibilità. Esercitando una leggera forza il sensore si sposta leggermente e di conseguenza ne attiva la funzione. Data la notevole sensibilità questi dispositivi sono indicati a persone che dispongono di una fine mobilità della testa, anche perché il supporto rigido può risultare dannoso per gli utenti con problemi a dosare la propria forza.



*Sensore LEAF (a petalo)  
(fig. 2)*

### **Sensori muscolari**

Questo tipo di dispositivi è in grado di percepire anche lievi movimenti muscolari e di trasformarli in un segnale di output. Il livello di precisione è chiaramente personalizzabile, per impedire che movimenti involontari siano interpretati come comandi. Sono costituiti da una fascia, all'interno della quale è contenuto appunto il sensore, si posiziona generalmente sulle braccia, ma può anche funzionare in diverse parti del corpo, quali la fronte o le gambe.



*Sensore muscolare  
(Muscular Switch)  
(fig. 3)*

### **Sensori pneumatici**

Come per gli emulatori di mouse a soffio, questo genere di sensori funziona mediante le capacità respiratorie dell'utente. Sono attivabili sia soffiando che aspirando.

### **Sensore a trazione**

Sensori estremamente sensibili e generalmente di piccole dimensioni. Per il loro funzionamento è sufficiente tirare una corda legata alla loro estremità, lo scorrere della corda all'interno del sensore ne permette la sua attivazione.

#### 2.2.1.4 Display Braille<sup>15</sup>

Sono dispositivi elettromeccanici che permettono di trasformare l'output di un computer in testo Braille leggibile da persone non vedenti. Il funzionamento è molto semplice, vengono sollevati ed abbassati sequenze di punti corrispondenti alle combinazioni puntiformi, in modo da formare i caratteri Braille corrispondenti.

Solitamente queste barre Braille sono composte da 40 celle, che corrispondono al numero di caratteri che generalmente si può trovare su ogni pagina dei libri stampati in braille su carta, anche se ne possono trovare anche da 80. Ognuna di queste celle è a sua volta formata da 8 punti, che in questo modo permettono di rappresentare 256 combinazioni, corrispondenti alla codifica ASCII<sup>16</sup> estesa. Solitamente una cella *Braille* è composta da 6 punti, il passaggio a 8 punti non comporta nessuna sostanziale variazione al modo classico di rappresentare le lettere, e di conseguenza di leggerle. L'aggiunta di 2 nuovi punti avviene sotto i 6 punti normali, in modo che non modifica il *Braille* classico, ma permettendo al tempo stesso la codifica di tutti i caratteri ASCII. Questa aggiunta è stata fatta con fine di arricchire il linguaggio *Braille* e non modificarlo.



Display Braille a 40 celle  
PAC Mate 40 Portable Braille Display  
(fig. 1.a)



Display Braille - particolare  
(fig. 1.b)

<sup>15</sup> **Braille** - per un approfondimento sul linguaggio Braille si rimanda all'appendice ad esso dedicata.

<sup>16</sup> **ASCII** - *American Standard Code for Information Interchange* (*Codice Standard Americano per lo Scambio di Informazioni* in italiano) è un sistema di codifica dei caratteri comunemente utilizzato nei calcolatori

## 2.2.2 Software

Esistono varie tipologie di software AT, anche in questo caso differenziati in base alle varie esigenze e ai vari problemi dell'utenza. Si possono infatti trovare software ausiliari che, coadiuvando software già presenti all'interno del computer, hanno il compito di estenderne o semplificarne le funzionalità; software specifici, cioè creati appositamente con fine didattico e/o terapeutico; infine estensioni o plug-in mirati, sono molto simili ai software ausiliari ma sono parte integrante del programma nel quale risiedono.

Di seguito saranno elencate alcune soluzioni software, le principali, con le loro funzionalità e possibili applicazioni.

### 2.2.2.1 Screen magnifier

Uno screen magnifier è un software che, interfacciandosi allo schermo di un computer permette di ingrandire porzioni desiderate di schermo senza l'ausilio di nessuna apparecchiatura hardware esterna, infatti consente di potenziare il normale output grafico di un computer, estendendone le funzionalità.

Gli screen magnifier sono principalmente pensati per persone con difetti visivi non troppo gravi, in qual caso sarà necessario uno screen reader.

Il funzionamento è molto semplice: viene ingrandita una porzione di testo, tramite la selezione di opzioni a video oppure tramite la pressione di una sequenza di tasti, dopo di che la zona ingrandita viene controllata tramite il movimento del mouse.

#### Funzionalità

- *Ingrandimento*: permette appunto di ingrandire porzioni di schermo, è la funzione base presente in ogni tipo di screen reader. Solitamente il range di zoom oscilla da 1- a 16 volte. Maggiore è il livello di zoom e minore sarà la quantità dello schermo originale visibile. Solitamente si decide di usare la minor quantità di zoom utile.
- *Inversione dei colori*: permette di cambiare la colorazione dello schermo, ad esempio invertendo i colori di un testo da nero su bianco a bianco su nero. Questa funzione può ridurre il bagliore dello schermo, che in alcuni può risultare fastidioso o stancante per la lettura.
- *Smoothing*: permette di “smussare” gli angoli o le forme degli elementi ingranditi presenti sullo schermo, questo per rendere il tutto più facile da riconoscere.
- *Customizzazione del cursore*: permette di modificare l'aspetto del cursore, che solitamente viene utilizzato per direzionare l'ingrandimento, e di conseguenza renderlo più facilmente individuabile all'interno dello schermo.
- *Differenti visualizzazioni*: alcuni screen magnifier permettono di poter personalizzare in che modo apparirà la parte ingrandita. Solitamente può occupare l'intero schermo, oppure una zona delimitata (esempio un angolo), o seguire il movimento del mouse come una lente di ingrandimento.
- *Screen reader*: come funzione avanzata di alcuni screen magnifier il testo selezionato, o semplicemente puntato, può essere letto da uno screen reader.

## **Screen Magnifier integrati in sistemi operativi**

- In *Microsoft<sup>17</sup> Windows<sup>18</sup>* fin da *Windows 98* è stato inserito uno screen magnifier base chiamato *Microsoft Magnifier<sup>19</sup>*.
- In *Apple<sup>20</sup> Mac OS X<sup>21</sup>* è possibile usare la funzione integrata di *screen magnification* semplicemente tenendo premuto il tasto *control* e controllando con la rotella del mouse la funzione di *zoom-in* e *zoom-out*.
- Molte distribuzioni *Linux<sup>22</sup>* che utilizzano come windows manager *Compiz-Fusion<sup>23</sup>* possono contare sulla validità di un ottimo plug-in che gestisce l'ingrandimento dello schermo chiamato *Enhanced Zoom Desktop<sup>24</sup>*.

### **2.2.2.2 Screen reader**

Uno screen reader è un software che permette di identificare ed interpretare ciò che è presente a schermo e ripresentarlo agli utenti in forme alternative. Queste forme alternative possono essere programmi di tipo *text-to-speech* oppure dispositivi di output *Braille*, entrambi verranno successivamente trattati in maggior dettaglio.

Gli screen reader sono pensati per persone non vedenti, con difficoltà visive, analfabeto o con difficoltà di apprendimento.

#### **Funzionamento**

Il funzionamento di uno screen reader è molto semplice e segue una determinata logica nel selezionare di volta in volta quale testo vocalizzare. Se ad esempio si apre il menu Start, lo screen reader legge la prima voce del menu. Se ci si sposta con i tasti freccia, la sintesi leggerà le varie voci del menu e darà informazioni aggiuntive, ad esempio indicando se una determinata voce ha un sottomenu.

Lo screen reader fornisce anche messaggi che aiutano ad orientarsi, ad esempio avverte se si è aperta una finestra di dialogo.

Durante l'esplorazione dello schermo lo screen reader può leggere ogni tipo di testo, ma non può far nulla se si trova davanti a delle scritte presenti in immagini. Per questo motivo è sempre utile inserire, al momento della creazione di pagine web, gli opportuni tag che consentono allo screen reader di leggere del testo alternativo.

---

<sup>17</sup> <http://www.microsoft.com/> - pagina ufficiale di Microsoft Corporation

<sup>18</sup> <http://www.microsoft.com/WINDOWS/> - pagina del sistema operativo Microsoft Windows

<sup>19</sup> <http://www.microsoft.com/windowsxp/using/accessibility/magnifierturnon.mspx> - pagina relativa allo screen magnifier di Microsoft Windows

<sup>20</sup> <http://www.apple.com/> - pagina ufficiale di Apple Inc.

<sup>21</sup> <http://www.apple.com/macosx/> - pagina del sistema operativo Apple Mac OS X

<sup>22</sup> <http://www.linux.org/> - pagina ufficiale del progetto Linux di Linus Torvalds

<sup>23</sup> <http://www.compiz.org/> - pagina ufficiale del progetto Compiz, windows manager per Linux

<sup>24</sup> <http://wiki.compiz.org/Plugins/Ezoom> - pagina ufficiale del progetto EZoom per Compiz

## Funzionalità

- *Verbosity*: con questo termine si intendono le caratteristiche di lettura di qualsiasi screen reader. Possono esserci due tipi di *verbosity*:
  - *Speech verbosity*: si tratta della frequenza con la quale viene letto il testo a schermo, può essere regolata in base alle esigenze di ogni singolo utente, regolando questa funzione è possibile rendere maggiormente comprensibile il testo anche a chi ha difficoltà di attenzione.
  - *Language verbosity*: chiaramente gli screen reader possono leggere testi in diverse lingue. Alcuni consentono addirittura di decifrare la lingua del testo in lettura ed adattarne la pronuncia. Questa caratteristica è chiaramente presente solamente per gli screen reader che utilizzano un output di tipo *text-to-speech*.
- *Emulazione*: alcuni screen reader consentono l'emulazione del mouse attraverso la pressione di determinate combinazioni di tasti. Questo permette di semplificare notevolmente la navigazione a schermo.
- *Personalizzazione*: le vari funzionalità sono personalizzabili per adattarsi meglio alle esigenze degli utenti.

## Tipi di screen reader

### *Command Line Interface (CLI)*

Gli screen reader, come qualsiasi dispositivo informatico, seguono direttamente lo sviluppo dei sistemi sui quali vengono utilizzati. Di conseguenza i primi screen reader, nati negli anni ottanta presso il *Research Centre for the Education of the Visually Handicapped*<sup>25</sup> presso l'*Università di Birmingham*<sup>26</sup>, funzionavano leggendo singolarmente linee di comando. Chiaramente questo tipo di screen reader non può funzionare in sistemi che utilizzano GUIs (Graphical User Interface). Questo tipologia, con l'avanzare dei tempi, sta passando in disuso e viene ormai solo utilizzata in particolari sistemi che prevedono ancora l'utilizzo della linea di comando.

### *Graphical User Interface (GUI)*

Con il passaggio da sistemi costituiti da solo testo a sistemi grafici il lavoro degli Screen Reader ha subito un ulteriore complicazione. Un sistema GUI è formato sia da parti testuali che da parti grafiche (come ad esempio bottoni, menu e tab) e sta allo screen reader, in alcuni casi anche con la collaborazione del sistema operativo, interpretarli correttamente.

Uno screen reader che opera su delle GUI può funzionare principalmente in due modi.

- *Off-screen model*: lo screen reader sta in “ascolto” dei messaggi riguardanti la grafica provenienti dal sistema operativo e crea, basandosi su di essi, una sorta di rappresentazione fittizia dello schermo, la quale sarà interamente navigabile dall’utente mediante l’uso di determinate funzioni, che solitamente corrispondono con l’uso di sensori.
- *APIs<sup>27</sup> accessibili*: grazie all’utilizzo di API è possibile sostituire l’utilizzo del modello *off-screen* semplificando notevolmente il “lavoro” dello screen reader, che in questo caso comunica direttamente con il sistema operativo, mediante queste procedure. Tutti i principali sistemi operativi mettono a disposizione questo genere di API.

---

<sup>25</sup> <http://www.education.bham.ac.uk/research/victar/> - link al Visual Impairment Centre for Teaching and Research dell’Università di Birmingham

<sup>26</sup> <http://www.bham.ac.uk/> - sito ufficiale dell’Università di Birmingham

<sup>27</sup> **API** - Application programming interface (*Interfaccia di Programmazione di un’Applicazione* in italiano) è un insieme di procedure che permette di astrarre e di conseguenza semplificare il funzionamento di un determinato programma, per permettere ad altro software di interagire con esso.

Sfortunatamente non tutto il software supporta l'utilizzo delle *API* e di conseguenza l'uso del *off-screen model* deve sempre essere reso disponibile.

### *Applicazioni self-voicing*

Sono programmi che integrano la funzione di lettura o di produzione di suoni di avviso, essi consentono alle persone con problemi visivi di interagire direttamente senza l'uso di applicazioni screen reader esterne. Di questa categoria fanno parte i *talking web browser*, ovvero browser con la funzione di *screen reader* integrata al loro interno. Questi browser però sono ormai diventati obsoleti, in favore di add-on mirate a browser più conosciuti e funzionali quali *Mozilla Firefox*<sup>28</sup> oppure *Opera*<sup>29</sup>. Grazie a *Fire Vox*<sup>30</sup>, estensione per *Mozilla Firefox*, viene reso disponibile un ottimo browser con funzionalità di *self-voicing* su i tre principali sistemi operativi (*Linux*, *Mac OS* e *Windows*).

### *Web-based*

Una relativamente nuova tipologia di screen reader è quella *web-based*. Vi sono principalmente due software che compiono questo genere di "lavoro", uno è *Spoken Web*<sup>31</sup> disponibile solo per *Microsoft Internet Explorer*<sup>32</sup> e di conseguenza solo per il sistema operativo *Microsoft Windows*, l'altro è *BrowseAloud*<sup>33</sup>, disponibile sia per *Windows* che per *Mac OS X*. In questo genere di screen reader sono presenti solo le più elementari funzioni e se ne consiglia l'uso solamente in casi eccezionali, ad esempio per la navigazione da un computer pubblico nel quale non si ha libertà di accesso software.

### **Screen reader integrati in SO**

- In *Apple Mac OS X* si può trovare il programma *VoiceOver*<sup>34</sup>, esso consente di ampliare le normali funzioni di accesso, ad esempio aggiungendo la possibilità di impartire comandi vocali, verso i computer *Macintosh*. Esiste anche una versione per dispositivi portatili *Apple* forniti di sistema operativo *Iphone OS*<sup>35</sup>.
- In *Linux*, più precisamente per l'ambiente grafico *GNOME*<sup>36</sup> vi è un interessante progetto open source chiamato *Linux Screen Reader*<sup>37</sup> in costante evoluzione e sviluppo.
- In *Microsoft Windows* è presente *Microsoft Narrator*<sup>38</sup>, si tratta di uno screen reader estremamente basilare, infatti è presente solamente la funzione di text-to-speech. Esistono comunque ottime alternative, molto più performanti, che permettono di colmare

---

<sup>28</sup> <http://www.mozilla.com/firefox/> - pagina ufficiale del browser Mozilla Firefox

<sup>29</sup> <http://www.opera.com/> - pagina ufficiale del browser Opera

<sup>30</sup> <http://firevox.clcworld.net/> - pagina ufficiale del progetto, sviluppato da Charles L. Chen

<sup>31</sup> <http://www.spoken-web.com/> - pagina ufficiale del progetto Spoken Web

<sup>32</sup> <http://www.microsoft.com/nz/ie/> - pagina ufficiale del browser Microsoft Internet Explorer

<sup>33</sup> <http://www.browsealoud.com/> - pagina ufficiale di TextHelp System, sviluppatori di BrowseAloud

<sup>34</sup> <http://www.apple.com/accessibility/voiceover/> - sezione sul sito Apple dedicata al software VoiceOver

<sup>35</sup> <http://www.apple.com/iphone/softwareupdate/> - pagina ufficiale del sistema operativo per Apple Iphone

<sup>36</sup> <http://www.gnome.org/> - GNOME (GNU Network Object Model Environment), Linux Desktop Environment

<sup>37</sup> <http://live.gnome.org/LSR> - pagina ufficiale del progetto Linux Screen Reader per GNOME

<sup>38</sup> <http://www.microsoft.com/enable/training/windowsvista/narrator.aspx> - pagina relativa allo screen reader Microsoft Narrator

questa lacuna. Un esempio su tutti può essere *NonVisual Desktop Access*<sup>39</sup> (NVDS), software open source e freeware, che permette di estendere l'accessibilità del sistema operativo mediante l'utilizzo di apparecchiature che supportano l'utilizzo del Braille.

### 2.2.2.3 Text-to-speech

I software di text-to-speech, in italiano chiamati di *sintesi vocale*, come già detto nel capitolo precedente hanno il compito di trasformare del testo scritto in parlato mediante una riproduzione artificiale della voce umana.

I software di sintesi vocale rivestono un ruolo di fondamentale importanza per l'accessibilità informatica. Combinati con *screen reader* consentono agli utenti non vedenti di poter interagire con i computer ed assistono anche persone con problemi visivi nella lettura dei testi. Inoltre questa tipologia di software è ottima per aiutare persone affette da dislessia ad ascoltare e di conseguenza a comprendere documenti di testo con maggior facilità.

#### Funzionamento

Il lavoro di un software di tipo text-to-speech si divide in due parti.

*Analisi del testo*: si occupa della conversione del testo in simboli fonetici. Per prima cosa vengono convertiti tutti i caratteri non testuali (come ad esempio i numeri) in testo, questa procedura è definita normalizzazione. Dopo di che il tutto viene convertito in simboli fonetici veri e propri, viene anche fatta un'analisi linguistica del testo rielaborato, per consentire ad esempio la corretta interpretazione dei segni di punteggiatura, questa procedura invece è definita da grafema a fonema.

*Generazione dei suoni*: una volta "decodificato" il testo, in trascrizione fonetica ed elementi di struttura delle frasi, tali informazioni sono utilizzate dal generatore di suoni per produrre la voce artificiale.

La sintesi vocale vera è propria si può realizzare mediante l'incatenazione di parti vocali memorizzati in un database, ad ogni fonema codificato corrisponde una di queste registrazioni. Un sistema costituito dal campionamento di singoli fonemi o coppie di essi avrà un database relativamente piccolo ma in compenso ne perderà la qualità della registrazione, un sistema che invece ricorre alla registrazione di parole intere o addirittura di intere frasi avrà una qualità migliore a discapito questa volta della velocità e dello spazio occupato dal database.

Un'alternativa a questo sistema è predisporre il sintetizzatore con dei modelli di tratti vocali in modo da cercare di simulare la voce umana nel modo più fedele possibile, questa tecnica però è molto più complicata della precedente.

Il livello di comprensione della voce prodotta da un computer è molto più importante che il livello di qualità della voce stessa, soprattutto se il software in questione è usato da persone non vedenti, le quali affidato alla sola voce digitale la comprensione di interi testi.

### 2.2.2.4 Speech-to-text

I software di *speech-to-text*, in italiano chiamati di *riconoscimento vocale*, funzionano esattamente all'opposto di quelli di *text-to-speech* appena trattati. Si occupano di riconoscere il linguaggio umano e di trasformarlo in un input valido ad un computer. Sono

<sup>39</sup> <http://www.nvda-project.org/> - pagina ufficiale del progetto NVDA (NonVisual Desktop Access)

utilizzati sia per riconoscere testi, e quindi trascriverli direttamente senza dovere digitare alcuna parola, oppure per controllare diversi sistemi solamente con l'uso della voce. Com'è facile intuire questo tipo di software è estremamente utile per chi ha problemi di mobilità degli arti e quindi l'interazione con la macchina potrebbe risultare alquanto complicata, ma non solo anche i normali utenti posso trarre giovamento da questa tipologia di applicazioni.

### **Funzionamento**

Un software di riconoscimento vocale funziona mediante un algoritmo di tipo statistico inferenziale, ricostruendo il linguaggio in base alla frequenza delle associazioni fra parole. Il riconoscimento a basso livello si basa anche su un vocabolario fonetico che contiene le pronunce base di numerose parole nella lingua impostata.

Esiste un'altra tipologia di riconoscimento vocale, più incentrata sul riconoscimento delle singole parole. Questo tipo di software necessita di una maggiore interazione dell'utente in quanto in questo caso il riconoscimento della pronuncia ricopre un ruolo fondamentale. Il programma si limita a trascrivere quanto detto dall'utente senza addentrarsi in ulteriori analisi della frasi o calcoli di abbinamento tra parole.

In entrambi i casi l'uso di un vocabolario non solo di singole parole, ma anche di loro associazioni tipiche, sia esso inserito ed aggiornato dal programmatore, dall'utente o per mezzi statistici, è di notevole aiuto a migliorare la qualità del riconoscimento.

Come già accennato, solitamente prima dell'uso di un software di riconoscimento vocale è necessario configurarlo, leggendo per esempio un brano davanti al microfono, per "addestrare" il programma alla voce del lettore. Questa procedura consente al software di ridurre drasticamente il numero di errori nei quali può incappare durante il riconoscimento vocale. Al momento dell'addestramento viene costruita una libreria di file vocali specifica per l'utente, questo viene fatto registrando campioni di voce, analizzandoli e campionandoli. L'addestramento può proseguire quando il programma non riconosce la pronuncia di una parola, digitando il testo e registrando la pronuncia corrispondente in modo che il programma crei un file vocale, creando di fatto un abbinamento testuale/vocale. Alla parola può essere abbinata, in alternativa, una sequenza di simboli dell'alfabeto fonetico.

### **2.2.2.5 Optical Character Recognition (OCR)**

I software OCR (*Riconoscimento Ottico dei Caratteri* in italiano) hanno lo scopo di trasformare il testo, sia esso stampato o scritto a mano, contenuto in un'immagine, solitamente prodotta mediante l'utilizzo di uno scanner, in testo digitalmente modificabile per mezzo di un editor.

Sono utilizzati per digitalizzare grandi quantità di testi disponibili solamente in formato cartaceo, come ad esempio vecchi libri. Questo, oltre a consentire una notevole editabilità e riusabilità dei testi, permette anche una maggior accessibilità alla cultura e alle informazioni da parte di persone con difficoltà visive, mediante l'utilizzo di software *text-to-speech* o *screen reader*, e soprattutto non dovendo necessariamente passare per decodificazioni più scomode e dispendiose che possono essere rappresentate ad esempio dai libri scritti in *Braille*.

### **Funzionamento**

Il funzionamento di questa tipologia di software ha notevoli analogie con quello dei sistemi *speech-to-text*. Il riconoscimento del testo si basa sia sul riconoscimento dei singoli

caratteri sia a come essi si trovano scritti all'interno di una parola e più in generale all'interno di una frase.

Data la grande varietà in cui si può presentare un testo scritto un software di riconoscimento ottico dei caratteri ha bisogno di un “addestramento” per migliorare notevolmente il suo funzionamento.

Le ultime versioni di questo tipo di software posseggono funzioni di riconoscimento ed interpretazione dei contorni del testo, in questo modo consentono di mantenere l'impaginazione dei documenti elaborati.

### 2.2.2.6 Spell Checker

Sono software o parti di esso che si occupano del controllo degli errori ortografici all'interno di un testo computerizzato. Possono essere programmi autonomi capaci di interagire con qualunque altro programma nel quale sia possibile introdurre un testo, oppure parte di altre applicazioni, come ad esempio nei *word processor*, in questo caso il loro scopo è di integrare le funzioni offerte da tale applicativo.

Le prime versioni di *spell checker* si limitavano alla sola messa in evidenza degli errori, ora la maggior parte di essi consente di suggerire possibili correzioni e in alcuni casi si possono addirittura impostare sostituzioni automatiche, sia di parole errate o di errori di capitolazione o di forma.

#### Funzionamento

Soltanamente il lavoro di uno *spell checker* si divide in due parti:

- Scansione del testo alla ricerca di errori e segnalazione, ad esempio sottolineandoli in rosso. La ricerca degli errori viene fatta confrontando ogni singola parola con quelle contenute all'interno di un dizionario prestabilito. In alcune versioni è possibile aggiungere o modificare la lista delle parole contenute nel dizionario.
- Analisi delle parole trovate errate e confronto con un lista di parole correlate per suggerire eventuali correzioni. Anche per operazione lo *spell checker* si avvale della funzione del dizionario.

Esistono tipologie di *spell checker* che non si limitano al solo controllo ortografico della singola parola ma ampliano il controllo, e di conseguenza i vari suggerimenti dati, sia alla sintassi che alla grammatica.

### 3. Accessibilità Web

#### 3.1 Definizione

Per definire l'Accessibilità Web mi avvalgo da quanto specificato dal *World Wide Web Consortium*<sup>40</sup>, abbreviato *W3C*, una comunità internazionale creata con lo scopo di migliorare ed assicurare la costante crescita del Web.

*Per accessibilità web si indica la capacità di un sito web di essere acceduto efficacemente (alla sua interfaccia e al suo contenuto) da utenti diversi in differenti contesti. Rendere un sito web accessibile significa permettere l'accesso all'informazione contenuta nel sito anche a persone con disabilità fisiche di diverso tipo e a chi dispone di strumenti hardware e software limitati. (W3C)*

Come già accennato in precedenza, si può notare da questa definizione, che il campo di interesse questa volta è relegato solamente al Web, concentrandosi sull'usabilità e sulla possibilità di reperire informazione tramite esso.

A livello internazionale i punti di riferimento sono i seguenti:

- **World Wide Web Consortium (W3C)**
- **Web Accessibility Initiative (WAI)<sup>41</sup>**  
Iniziativa per l'Accessibilità del Web (a cura del W3C)
- **Web Content Accessibility Guidelines (WCAG)<sup>42</sup>**  
Linee Guida per l'Accessibilità del Contenuto del Web (a cura della WAI)

Esistono diversi modi per valutare l'accessibilità di un sito web. Queste valutazioni sono chiaramente solo di livello teorico e seguono un procedimento euristico. Solitamente l'analisi del sito viene effettuata in base ad elenchi di direttive di accessibilità web. Gli organismi competenti non hanno ancora proposto degli standard specifici per quanto concerne l'accessibilità delle interfacce web, ma è possibile usare le norme ISO per i sistemi e prodotti informatici (*ISO 9241*).

Uno buon modo per effettuare una misurazione dell'accessibilità di un sito web consiste nel focalizzare l'analisi dell'usabilità su cinque fattori:

- facilità di apprendimento
- memorizzazione
- efficienza d'uso (velocità)
- frequenza degli errori
- soddisfazione soggettiva

Indipendentemente dai vari metodi di classificazione dell'usabilità di un sito vi sono dei criteri generali che è opportuno tenere in considerazione quando si vuole ottenere un sito accessibile:

---

<sup>40</sup> <http://www.w3.org/> - pagina ufficiale del World Wide Web Consortium (W3C)

<sup>41</sup> <http://www.w3.org/WAI/> - testo completo della specifica WAI

<sup>42</sup> <http://www.w3.org/TR/WCAG/> - testo completo della specifica WCAG

### **Browser:**

La visualizzazione del sito deve essere indipendente dal browser utilizzato. E' importante testare le proprie pagine con browser diversi come: *Mozilla Firefox*, *Internet Explorer*, *Safari*, *Opera*, *Konqueror*, ecc.. affinché per ogni browser la visualizzazione sia coerente. E' da tenere presente anche che, con i nuovi sviluppi tecnologici, la navigazione non riguarda più solamente i computer ma anche dispositivi portatili, come ad esempio gli smarthphone.

### **Risoluzione:**

Ogni sito dovrebbe essere realizzato in modo da essere indipendente dalla grandezza del monitor a disposizione dell'utente finale, che può spaziare da un televisore a ventiquattro pollici ad uno schermo di cinque pollici di un telefono cellulare. In questo caso e' importante capire quanto sia rilevante rendere accessibili le pagine web in questione anche da dispositivi portatili.

### **Tempi di caricamento:**

Data l'importanza dei tempi di caricamento dei diversi siti internet, un utilizzo razionale dell'HTML favorisce la presentazione di siti che garantiscono tempi di caricamento ottimali. La presenza di contenuti *embed*, ovvero contenuti multimediali integrati nelle pagine Web, può gravare molto sulla velocità di caricamento. Anche la tecnologia con la quale si accede al web può influenzare la necessità di avere pagine agili da caricare.

### **Architettura informativa del sito:**

Ogni sito internet deve essere creato seguendo una logica lineare e ben delineata. In nessuna occasione ci si deve trovare difronte a vicoli ciechi, ovvero pagine che non portano da nessuna altra parte. Collegandosi all'indirizzo principale del sito web si deve accedere sempre direttamente all'Home Page e da lì si deve essere in grado di raggiungere ogni sottosezione principale nella quale è diviso il sito internet. In alcuni casi può risultare utile la creazione di una mappa del sito, soprattutto per quanto riguarda pagine di enti o che più in generale offrono servizi agli utenti.

### **Home Page:**

Le pagine web devono essere sempre ben visibili e chiare nei loro contenuti fin dal primo sguardo. Sono da evitare i colori particolarmente accesi, possono infastidire la lettura, o eccessivamente piccoli, in questo caso possono affaticare la vista. Se si tratta di un sito istituzionale il nome, unitamente al logo, deve essere sempre posto in modo visibile.

L'orientamento all'interno del sito deve essere impostata in maniera logica già dalla Home Page, una barra o menu di navigazione può aiutare in questo scopo.

Le immagini utilizzate devono sempre contenere testi alternativi, in caso si accedesse con dispositivi di lettura a video o anche per eventuali stampe particolare. Si deve sempre cercare di visualizzare tutti i contenuti principali nella prima schermata visibile, tentando il meno possibile di costringere l'utente ad usare la *scroll bar*.

### **Grafica e leggibilità del sito:**

Il contrasto sfondo/testo deve essere sempre abbastanza nitido da rendere visibile e leggibile il testo. I titoli sono generalmente giustificati a sinistra e comunque resi distinguibili dal resto del testo. Per fornire un livello di accessibilità maggiore è consigliabile l'aggiunta di una *Access Bar* che consenta di modificare dinamicamente la grandezza, il formato ed il colore del testo unitamente al colore dello sfondo.

## 3.2 Il World Wide Web Consortium (W3C)

*"To lead the Web to its full potential / Portare il Web al suo massimo potenziale"*  
*(Motto del W3C)*



Il W3C è stato creato nell'Ottobre del 1994 con lo scopo di portare il Web al suo massimo potenziale, mediante lo sviluppo di protocolli comuni che promuovano la sua evoluzione ed assicurino la sua interoperabilità. In altre parole ha il compito di sviluppare standard per assicurare la costante crescita del Web. Oggi il W3C conta oltre 450 Membri ed uno staff di circa 70 persone provenienti da tutto il mondo.

Per capire meglio come opera questa organizzazione è opportuno prendere in esame i sette punti sopra cui si basa il suo lavoro:

### 1. Accesso Universale

*Il W3C definisce il Web come l'universo delle informazioni accessibili in rete (disponibili attraverso il computer, il telefono, la televisione, o il frigorifero telematico...). Oggigiorno questo universo permette alla società di fruire di nuove forme di comunicazione umana e offre nuove opportunità di condividere la conoscenza. Uno degli scopi principali del W3C è quello di rendere queste opportunità fruibili a tutti, indipendentemente da eventuali limitazioni determinate da hardware, software, supporto di rete a disposizione, lingua madre, cultura, collocazione geografica, capacità fisiche e mentali. L'impegno del Consorzio per l'accesso universale è dimostrato da varie attività: Internationalization Activity, Device Independence Activity, Voice Browser Activity, e Web Accessibility Initiative.*

### 2. Web Semantico

*La gente oggi condivide la conoscenza sul Web utilizzando un linguaggio umano, comprensibile dalle altre persone. Sul Semantic Web<sup>43</sup> ("semantico" significa "che ha a che fare con il significato"), le persone saranno capaci di esprimersi in termini tali che i nostri computer potranno interpretare e scambiare informazioni. In questo modo si darà la possibilità ai computer di risolvere i problemi che troviamo noiosi da affrontare, e di aiutarci a trovare velocemente quello che stiamo cercando: informazioni mediche, una recensione di un film, l'ordine di un libro attraverso un catalogo, etc. Il linguaggi del W3C RDF<sup>44</sup>, XML<sup>45</sup>, XML Schema, e XML signatures sono gli elementi di base su cui poggia il Web Semantico.*

### 3. Fiducia

*Il Web è un mezzo di collaborazione, non un supporto di sola lettura come una rivista. Di fatto il primo browser Web era anche un editor, sebbene molti oggi pensino al browsing essenzialmente in termini di visualizzazione dell'informazione, senza interazione. Per promuovere un ambiente più collaborativo il Consorzio sta progettando un "Web of Trust" capace di offrire riservatezza, che ispiri gradualmente fiducia, e che faccia in modo che ci si prenda la responsabilità di ciò che viene pubblicato su Web. Questi principi motivano*

<sup>43</sup> <http://www.w3.org/2001/sw/> - W3C Semantic Web Activity

<sup>44</sup> <http://www.w3.org/RDF/> - Resource Description Framework (RDF)

<sup>45</sup> <http://www.w3.org/XML/> - Extensible Markup Language (XML)

*molto del lavoro del W3C su XML signature, meccanismi di annotazione, composizione di gruppo (group authoring), gestione delle versioni, etc.*

#### **4. Interoperabilità**

*Venti anni fa si comperavano software che funzionavano solamente con altri software dello stesso produttore. Oggi si ha molta più libertà di scelta e giustamente ci si aspetta che i componenti software possano essere interscambiabili. Ci si aspetta anche di essere in grado di vedere il contenuto del Web con l'ausilio del software che si preferisce (browser grafico, sintetizzatore vocale, display braille, telefono a bordo dell'auto...). Il W3C è un'organizzazione neutrale, che incoraggia l'interoperabilità attraverso la progettazione e la promozione di linguaggi informatici e protocolli aperti (non proprietari) che evitino la frammentazione del mercato caratteristica del passato. Tutto questo è realizzato ottenendo il consenso dell'industria e incoraggiando un forum aperto per la discussione.*

#### **5. Capacità evolutiva**

*Il W3C mira all'eccellenza tecnica ma è consapevole che quello che oggi conosciamo e di cui abbiamo bisogno potrebbe essere insufficiente per risolvere i problemi di domani. Il Consorzio si adopera perciò per costruire un Web che possa facilmente evolvere in un Web ancora migliore, senza per questo dover distruggere quello che già funziona. I principi di semplicità, modularità, compatibilità ed estensibilità guidano tutti i progetti del Consorzio.*

#### **6. Decentralizzazione**

*La decentralizzazione è un principio fondamentale dei moderni sistemi distribuiti, comprese le società umane. In un sistema centralizzato ogni messaggio o azione deve passare attraverso un'autorità centrale, causando ingorghi quando il traffico aumenta. Il progetto del Consorzio è quello di limitare il numero delle risorse Web centralizzate, al fine di ridurre la vulnerabilità del Web nel suo complesso. La flessibilità è l'indispensabile compagnia dei sistemi distribuiti, è la vita e l'anima di Internet, non solo del Web.*

#### **7. Multimedia più eccitante!**

*A chi non piacerebbe maggiore interattività e supporti multimediali più ricchi sulla rete, come immagini ridimensionabili, qualità del suono, video, effetti 3D ed animazioni? Il processo di consenso del W3C non limita la creatività dei content provider e non comporta rendere noiosa la navigazione. Grazie ai suoi membri il W3C dà ascolto ai consumatori finali e opera per fornire una solida infrastruttura di sviluppo per un Cooler Web grazie a linguaggi come SVG<sup>46</sup> e SMIL<sup>47</sup> (Synchronized Multimedia Integration Language).*

---

<sup>46</sup> <http://www.w3.org/Graphics/SVG/> - Scalable Vector Graphics (SVG)

<sup>47</sup> <http://www.w3.org/AudioVideo/> - Synchronized Multimedia Integration Language (SMIL)

### **3.3 Web Accessibility Initiative (WAI)**

Con *Web Accessibility Initiative* (*Iniziativa per il Web Accessibile* in italiano) si intende il ramo del *W3C* che si occupa di migliorare l'accessibilità del *World Wide Web*. Questa iniziativa è stata pensata appositamente per garantire e tutelare l'accessibilità Web da parte di utenti con qualsiasi tipo di disabilità e, indirettamente, migliorarne anche la fruizione da parte di tutti gli utenti.

Scopo principale del WAI è sviluppare linee guida e tecniche per semplificare e direzionare il lavoro di chi sviluppa materiale per il Web.

Il *W3C* ha stilato una breve linea guida per facilitare la creazione di siti web accessibili mediante brevi consigli, il tutto si snoda su una lista di dieci punti:

#### **1. Immagini ed animazioni**

*Utilizzare l'attributo alt per descrivere la funzione di ogni elemento grafico.*

Quando ad esempio si inserisce un immagine all'intero di una pagina web lo si fa mediante il tag *img*. Per capire meglio questo punto è opportuno specificare nel dettaglio il funzionamento del tag *img*.

Il tag *img* ha una struttura molto semplice, per mostrare un immagine sarà solamente necessario impostare l'attributo *src* con l'indirizzo appunto dell'immagine:

```

```

L'attributo *alt* ha il compito di fornire al browser un testo alternativo qualora non fosse possibile la visualizzazione dell'immagine oppure nel caso la pagina venga letta da uno screen reader.

In questo esempio sono rappresentati solo gli attributi essenziali, ve ne sono altri ma non di rilevante importanza per l'accessibilità.

#### **2. Immagini cliccабili**

*Utilizzare l'elemento map e descrivere le zone attive.*

L'elemento *map* consente di definire un'immagine con diverse aree cliccабili. Anche in questo caso, come già visto per le immagini, è opportuno aggiungere una descrizione alternativa ad ognuna delle aree cliccабili poste all'interno dell'immagine, per far questo sarà sufficiente l'uso dell'attributo *alt*.

#### **3. Multimedia**

*Fornire sottotitoli e trascrizioni per l'audio, e descrizione di filmati.*

Questo aspetto risulta fondamentale per quanto riguarda l'accessibilità dei contenuti multimediali presenti su pagine Web da parte di utenti con problemi visivi o uditivi. La presenza di trascrizioni e descrizioni consente la fruibilità di tali contenuti mediante strumenti AT quali ad esempio gli *screen reader*.

## 4. Link ipertestuali

*Utilizzare enunciati che conservino il loro senso al di fuori del contesto. Per esempio, evitare «cliccare qui».*

Link ipertestuali non propriamente definiti, come l'esempio precedente, possono creare ambiguità non indicando dove porta effettivamente il collegamento.

## 5. Organizzazione

*Utilizzare titoli, liste e una struttura coerente. Utilizzare CSS<sup>48</sup> per l'impaginazione.*

CSS - *Cascading Style Sheet* (Fogli di stile a cascata) è un semplice meccanismo per l'aggiunta dello stile (per es. caratteri, colori, spaziatura) ai documenti Web. Praticamente sono file, contenenti appunto lo stile delle pagine Web, atti a descrive l'aspetto semantico, come appare e com'è formattato, di pagine web (HTML e XHTML).

Per un analisi più accurata dei CSS si rimanda ai capitoli successivi.

## 6. Figure e diagrammi

*Descriverli all'interno della pagina o utilizzare l'attributo longdesc.*

L'attributo *longdesc* consente di specificare un URL di una pagina che contiene una lunga descrizione di un'immagine. Svolge un compito analogo ad *alt* con la sola differenza che ne amplia le capacità descrittive, consentendo appunto di collegare una pagina o un documento appositi per la descrizione dell'immagine.

```

```

In questo esempio si può notare che è consigliabile definire l'attributo *img* con una brevissima descrizione dell'immagine, solitamente alcune parole sono sufficienti, e delegare all'attributo *longdesc* una descrizione più accurata.

## 7. Script, applet e plug-in

*Fornire una pagina alternativa quando tali funzionalità sono inaccessibili o non supportati.*

Non tutti i browser supportano *script*, *applet* o *plug-in* vari e, in alcuni casi, l'utente per svariati motivi può disabilitare il normale funzionamento di questi. Di conseguenza è sempre opportuno fornire delle descrizioni alternative agli elementi "esterni" inseriti all'interno delle pagine Web.

L'esempio seguente illustra il funzionamento con uno *script* e relativo tag *noscript*.

```
<script type="text/javascript">
  document.write("Hello World!")
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

---

<sup>48</sup> <http://www.w3.org/Style/CSS/> - pagina del W3C riguardante i CSS

Lo *script* qui riportato ha il compito di stampare a video la frase *Hello World!*. Nel caso qualcosa non dovesse funzionare, al posto dello *script*, verrà visualizzato il testo contenuto nel tag *noscript*. Al suo interno può essere inserito qualsiasi elemento normalmente collocabile in una pagina HTML. Il contenuto del tag *noscript* verrà mostrato solamente nel qual caso gli *script* non siano supportati oppure siano stati disabilitati dall'utente.

## 8. Cornici (frames)

*Utilizzare noframes e titoli significativi.*

Il tag *noframes* ha un funzionamento analogo a *noscript*, con la sola differenza che viene usato quando un browser non supporta i *frames*.

I frames non sono altro che suddivisioni della pagina in parti a se stanti. Consentono di visualizzare differenti documenti, collegati con l'attributo *src*, contemporaneamente all'interno di una sola pagina Web.

```
<frameset cols="25%,50%,25%">
  <frame src="frame_a.htm" />
  <frame src="frame_b.htm" />
  <frame src="frame_c.htm" />
<noframes>
  Sorry, your browser does not handle frames!
</noframes>
</frameset>
```

Come mostrato nell'esempio se il browser non supporta i frame la pagina non verrà caricata e al suo posto apparirà il testo (o la pagina alternativa) contenuto nel tag *noframe*.

Chiaramente per motivi di accessibilità e portabilità di pagine Web l'uso dei frames è altamente consigliato.

## 9. Tabelle

*Facilitare la lettura linea per linea. Riassumere.*

## 10.Verificare il lavoro

*Validare. Utilizzare gli strumenti, la lista di controllo e le linee guida di WCAG.*

Sia la validazione del codice che la linea guida WCAG saranno trattati nei prossimi capitoli.

### 3.3.1 WAI - Accessible Rich Internet Applications<sup>49</sup> (WAI-ARIA)

*Web Accessibility Initiative - Accessible Rich Internet Applications* è il nome di un insieme documenti pubblicati dal W3C e, più precisamente, dal ramo della *Web Accessibility Initiative* che si occupano di accrescere l'accessibilità dei contenuti dinamici e dei componenti per l'interfaccia utente sviluppati con *Ajax*<sup>50</sup>, *HTML*, *JavaScript*<sup>51</sup> ed altre tecnologie collegate.

Per capire meglio cosa distingue questo insieme di documenti dalle normali specifiche WAI è utile spiegare cosa si intende con la dicitura ARIA.

Con il nome di *Rich Internet Applications (RIA)* si intendono tutte quelle applicazioni dinamiche presenti sul Web che non necessitano di un installazione in locale e possono comunemente funzionare mediante l'utilizzo di un browser. Queste applicazioni sono caratterizzate da una architettura di tipo distribuito, infatti la parte dell'applicazione che elabora i dati è trasferita a livello *client* e fornisce una pronta risposta all'interfaccia utente, mentre la gran parte dei dati e dell'applicazione rimane sul *server remoto*, con notevole alleggerimento per il computer dell'utente.

Principalmente l'insieme di documenti WAI-ARIA si divide in tre parti:

#### **Accessible Rich Internet Applications (WAI-ARIA) Version 1.0<sup>52</sup>**

Documento principalmente orientato agli sviluppatori di tecnologie web assistive ed altri *user agent*, sviluppatori di altre specifiche tecniche e sviluppatori di strumenti per valutare l'accessibilità.

#### **WAI-ARIA Primer<sup>53</sup>**

Introduzione tecnica alla WAI-ARIA. Descrive i problemi a cui ci indirizza WAI-ARIA, i concetti di base, l'approccio tecnico e le motivazioni di business per cui si deve adottare WAI-ARIA.

#### **WAI-ARIA Esperienze più significative<sup>54</sup>**

Describe le migliori pratiche per diffondere *rich internet applications* con WAI-ARIA: affronta temi come i primi passi per costruire *widgets*, navigazione da tastiera, relazioni, proprietà di form, supporto per copia e incolla, avvisi e finestre di dialogo, librerie componenti riutilizzabili, e strumenti di test.

---

<sup>49</sup> <http://www.w3.org/WAI/intro/aria> - pagina del W3C dedicata alla specifica WAI-ARIA.

<sup>50</sup> **AJAX** - Asynchronous JavaScript and XML (JavaScript e XLM Asincrono in italiano) si tratta di una tecnica di sviluppo per applicazioni web interattive e dinamiche. Lo sviluppo di applicazioni HTML con AJAX si basa su uno scambio di dati in background fra web browser e server, che consente l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente. Per un approfondimento si rimanda alla sezione dedicata sul sito del W3C school (<http://www.w3schools.com/Ajax/Default.Asp>).

<sup>51</sup> **JavaScript** (anche abbreviato JS) è un linguaggio di scripting orientato agli oggetti comunemente usato nei siti web. Per un approfondimento si rimanda alla sezione dedicata sul sito del W3C school (<http://www.w3schools.com/js/default.asp>).

<sup>52</sup> <http://www.w3.org/TR/wai-aria/>

<sup>53</sup> <http://www.w3.org/TR/wai-aria-primer/>

<sup>54</sup> <http://www.w3.org/TR/wai-aria-practices/>

### 3.3.2 La specifica Web Content Accessibility Guidelines (WCAG)

WCAG - *Web Content Accessibility Guidelines* (*Linea guida per l'Accessibilità dei Contenuti Web* in italiano) è appunto una linea guida, sviluppata dal W3C e relativa al ramo della *Web Accessibility Initiative*, che ha come scopo principale quello di indirizzare gli sviluppatori Web a produrre contenuti principalmente accessibili da persone disabili, senza però dimenticare i normali utenti, prendendo in considerazione anche i dispositivi più limitati per l'accesso al Web, come ad esempio cellulari e smartphone.

La specifica WCAG si snoda su *diciannove argomenti principali*, i quali sono raggruppati in quattro *Principi fondamentali*:

1. **Percepibile** (argomenti da 1 – 7)
2. **Fruibile** (argomenti da 8 – 12)
3. **Comprensibile** (argomenti da 13 – 16)
4. **Durevole** (argomenti da 17 – 19)

#### **Principio 1: PERCEPIBILE**

*I contenuti devono essere percepibili*

Il principio numero uno si riferisce alla percettibilità dei contenuti, vale a dire alla possibilità di poter accedere al contenuto almeno in modalità testuale.

I tre punti di riferimento per questo principio sono:

- **Utilizzare equivalenti testuali**
- **Separare la presentazione dal contenuto**
- **Sincronizzare gli elementi multimediali**

#### **Utilizzare equivalenti testuali**

**1.1 – Per i contenuti non testuali, fornire un testo equivalente che abbia lo stesso scopo o fornisca le stesse informazioni del contenuto non testuale, ad esclusione di quando lo scopo del contenuto non testuale è di generare un'esperienza sensoriale specifica (ad esempio, musica ed arte visiva). In questo caso è sufficiente una etichetta (text label) o una descrizione.**

**1.4 - Tutto il testo deve poter essere decodificato in parole tramite Unicode<sup>55</sup>.**

#### **Separare la presentazione dal contenuto**

**1.3 – Le informazioni, la funzionalità e la struttura sono separabili dalla presentazione.**

**1.5 – La struttura è resa percepibile attraverso la presentazione.**

#### **Sincronizzare gli elementi multimediali**

**1.2 – Fornire media equivalenti sincronizzati per le presentazioni dipendenti dal tempo.** (esempio: descrizione audio per eventi visivi, sottotitoli per dialoghi significativi, con sincronizzazione)

**1.6 – Per le presentazioni visive predefinite il contenuto principale è facilmente distinguibile dallo sfondo.**

---

<sup>55</sup> **Unicode** è un sistema di codifica che assegna un numero univoco ad ogni carattere usato per la scrittura di testi, in maniera indipendente dalla lingua, dalla piattaforma informatica e dal programma utilizzati.

**1.7 – Per le presentazioni uditive predefinite il contenuto principale è facilmente distinguibile dallo sfondo.**

## **Principio 2: FRUIBILE**

*Accertarsi che gli elementi dell’interfaccia nel contenuto siano operabili da qualsiasi utente.*

Il principio numero due si riferisce alla fruibilità dei contenuti, vale a dire alla possibilità di poter interagire con i contenuti e con le eventuali interfacce personalizzate.

*I tre punti di riferimento per questo principio sono:*

- **Fornire interfacce utilizzabili almeno con tastiera**
- **Controllare i tempi di esecuzione**
- **Evitare interruzioni nello schermo**

### **Fornire interfacce utilizzabili almeno con tastiera**

**2.1 – Tutte le funzionalità sono fruibili almeno attraverso tastiera o emulatore di tastiera.**

**2.4 – Aggiungere funzionalità che consentano di facilitare l’orientamento e lo spostamento tra i contenuti.**

(esempi: Struttura gerarchica del mark up, mappa del sito, più meccanismi di navigazione, possibilità di saltare blocchi di contenuto, utilizzare corrispondenze per i riferimenti incrociati nelle tabelle di dati. Assicurare che le relazioni fra le intestazioni di tabella e i dati rimangano trasparenti a chi utilizza uno *screen reader* o la *barra Braille*.)

**2.5 – Fornire funzionalità che non inducano in errore o altrimenti consentire la possibilità di rimediare.**

(esempi: scrittura nelle *form*, scelta da lista opzioni, *spell checker*, diverse tipologie di ricerca nei *search engine*.)

### **Controllare i tempi di esecuzione**

**2.2 – Gli utenti possono controllare in qualsiasi momento i tempi di lettura, risposta o interazione a meno che non sia possibile in caso di eventi in tempo reale.**

(esempi: controllo del tempo negli eventi (redirection, testo scorrevole, dialoghi che scompaiono, aggiornamenti con refresh, animazioni, filmati), possibilità di disattivare animazioni e movimenti.)

### **Evitare interruzioni nello schermo**

**2.3 – Consentire agli utenti di evitare le interruzioni dello schermo.**

Questo punto è stato espressamente creato per gli utenti che soffrono di epilessia fotosensibile e difficoltà di concentrazione.

## **Principio 3: COMPRENSIBILE**

*Rendere il contenuto e i comandi comprensibili al maggior numero di utenti possibile.*

Il principio numero tre si riferisce alla comprensibilità dei contenuti, vale a dire alla capacità di rendere chiari e semplici i contenuti.

I tre punti di riferimento per questo principio sono:

- **Identificare la lingua utilizzata**
- **Utilizzare un linguaggio chiaro e semplice**
- **Creare uno stile coerente**

### **Identificare la lingua utilizzata**

**3.1 – Rendere identificabile al programma utente la lingua utilizzata nei contenuti.**

### **Utilizzare un linguaggio chiaro e semplice**

**3.2 – La definizione delle abbreviazioni e degli acronimi è determinabile senza ambiguità.**

**3.3 – I contenuti sono comprensibili e/o è fornita una versione alternativa con linguaggio chiaro e semplice.**

### **Creare uno stile coerente**

**3.4 – La disposizione e la presentazione del contenuto sono coerenti o prevedibili, ma non identiche.**

(esempi: le barre di navigazioni sono in posizioni costanti, le scelte lessicali per le etichette e le intestazioni sono coerenti, la grafica delle sezioni è riconoscibile, le convenzioni usate sono familiari, diverse presentazioni del layout sono selezionabili dall'utente, spiegazioni testuali per avvisare di comportamenti inusuali.)

## **Principio 4: DUREVOLE**

*Utilizzare le tecnologie Web che accrescono l'operatività del contenuto con le tecnologie utente e assistive attuali e future.*

Il principio numero quattro si riferisce all'evolvibilità delle tecnologie, vale a dire alla possibilità di poter interagire con i contenuti e con le eventuali interfacce personalizzate attuali e future.

I tre punti di riferimento per questo principio sono:

- **Rispettare le specifiche**
- **Utilizzare tecnologie accessibili**
- **Dichiarare le tecnologie utilizzate**

### **Rispettare le specifiche**

**4.1 – Utilizzare le tecnologie secondo le proprie specifiche.**

(esempi: il markup è valido rispetto allo schema DTD (*Document Type Definition*), elementi e attributi vengono usati secondo le specifiche, non in modo presentazionale.)

### **Utilizzare tecnologie accessibili**

**4.2 – Le interfacce dei programmi sono accessibili o è prevista una versione alternativa accessibile.**

### **Dichiarare le tecnologie utilizzate**

**4.3 – Le tecnologie utilizzate all'interno dei contenuti sono dichiarate e largamente diffuse.**

### 3.3.2.1 Conformance Level<sup>56</sup>

Con *conformance level* si indica come una pagina Web rispetta tutti i punti focali della specifica WCAG e di conseguenza da quanto stilato dal WAI.

Ogni punto focale (*checkpoint*) della guida è contraddistinto da un livello di priorità, questo livello è proporzionale all'impatto che l'insoddisfazione di tale punto potrebbe avere nei confronti dell'accessibilità generale della pagina.

#### Priorità 1

Uno sviluppatore di pagine Web dove deve soddisfare questo checkpoint. Altrimenti, per una o più categorie di disabilità sarà impossibile accedere alle informazioni presenti nel documento. La soddisfazione di questo checkpoint è un requisito base per consentire l'accesso ad alcune categorie a documenti Web.

#### Priorità 2

Uno sviluppatore di pagine Web dovrebbe soddisfare questo checkpoint. Altrimenti, per una o più categorie di disabilità sarà difficile accedere alle informazioni presenti nel documento. La soddisfazione di questo checkpoint rimuoverebbe importanti barriere per l'accesso a documenti Web.

#### Priorità 3

Uno sviluppatore di pagine Web può soddisfare questo checkpoint. Altrimenti, per una o più categorie di disabilità potrebbero incontrare qualche difficoltà ad accedere alle informazioni presenti nel documento.

Alcuni checkpoint possono specificare un livello di priorità che potrebbe cambiare in determinate condizioni, queste condizioni sono indicate all'interno della specifica.

#### Livello di adempimento:

Livello "A": tutti i checkpoint con priorità 1 sono soddisfatti.



```
<a href="http://www.w3.org/WAI/WCAG2A-Conformance"
    title="Explanation of WCAG 2.0 Level A Conformance">
    
</a>
```

Livello "AA" o "Double-A": tutti i checkpoint con priorità 1 e 2 sono soddisfatti.



```
<a href="http://www.w3.org/WAI/WCAG2AA-Conformance"
    title="Explanation of WCAG 2.0 Level Double-A Conformance">
    
</a>
```

<sup>56</sup> <http://www.w3.org/WAI/WCAG2-Conformance.html> - pagina relativa al conformance level del W3C

Livello "AAA" o "Triple-A": tutti i checkpoint con priorità 1, 2 e 3 sono soddisfatti.

```
<a href="http://www.w3.org/WAI/WCAG2AAA-Conformance"  
    title="Explanation of WCAG 2.0 Level Triple-A Conformance">  
      
</a>
```



### 3.3.3 Ulteriori documentazione rilasciate dal WAI

- **Authoring Tools Accessibility Guidelines<sup>57</sup> (ATAG)**

*La specifica ATAG (Linea guida per l'Accessibilità degli Strumenti per Autori - di contenuti Web si intende) fornisce la linea guida per gli sviluppatori di strumenti per la creazione di contenuti Web.*

*Il suo scopo è assistere gli sviluppatori nella progettazione di strumenti di sviluppo, come ad esempio Web editors, in grado di generare contenuti accessibili per il Web ed in secondo luogo indirizzarli nella creazione di interfacce sempre più accessibili agli utenti.*

*Gli strumenti di "authoring" hanno come fine principale quello di consentire agli sviluppatori di creare contenuti Web accessibili. Essi forniscono ausili quali sistemi di test e correzione (un esempio possono essere i validatori), prompts, avvertimenti, aiuti in linea e automatismi.*

*Se da una parte è importante che tutti possano accedere ai contenuti Web è altresì importante che tutti possano essere in grado di generarli. Di conseguenza lo strumento che si utilizza per creare tali contenuti deve essere ugualmente accessibile.*

- **User Agent Accessibility Guidelines<sup>58</sup> (UAAG)**

*La specifica UAAG (Linea guida per l'Accessibilità degli User Agent) contiene una linea guida per lo sviluppo degli user agents al fine di diminuire le barriere per l'accessibilità del web per persone con qualsiasi tipo di disabilità.*

*Con User Agent si intende qualsiasi strumento che rende possibile all'utente l'accesso a contenuti Web. Di conseguenza con user agent si intendono tutti quei software quale Web browsers, media players, plug-in ed altri programmi che possono aiutare l'utente ad accedere e rendere fruibili i contenuti presenti sul Web.*

- **XML Accessibility Guidelines<sup>59</sup> (XAG)**

*La specifica XAG (Linea guida per l'Accessibilità del XML) spiega come includere "features" in documenti XML promuovendone la loro accessibilità, chiaramente rispettando le norme della specifica XML.*

*Lo sviluppo di questa guida è stato fermato nel 2002, al momento è disponibile solo in versione non definitiva.*

---

<sup>57</sup> <http://www.w3.org/TR/ATAG20/> - testo completo della specifica ATAG

<sup>58</sup> <http://www.w3.org/TR/UAAG20/> - testo completo della specifica UAAG

<sup>59</sup> <http://www.w3.org/TR/xag> - testo completo della specifica XAG

## 3.4 Migliorare ulteriormente l'accessibilità

### 3.4.1 Validazione del codice

La validazione del codice, mediante gli strumenti messi a disposizione dal W3C, è di cruciale importanza se si vogliono creare pagine ad alta accessibilità. La conformità semantica e sintattica del codice è essenziale per passare questa verifica ed oltre a consentire un buon primo passo verso l'accessibilità della pagina consente di ottenere anche una buona affidabilità per quanto riguarda il *cross-browsing*<sup>60</sup>.

Se il codice è valido significa che è stato scritto rispettando i parametri e gli standard imposti dal W3C.

Esistono tre tipi di validatori automatici del codice, tutti forniti dal W3C:

- Validatore del codice (X)HTML<sup>61</sup>
- Validatore del codice CSS<sup>62</sup>
- Validatore semantico del codice<sup>63</sup>

Se il codice risulta valido si potrà inserire all'interno della pagine Web queste immagini che lo certificano:

*Codice XHTML valido*



```
<p>
    <a href="http://validator.w3.org/check?uri=referer">
        
    </a>
</p>
```

*Codice CSS valido*



```
<p>
    <a href="http://jigsaw.w3.org/css-validator/check/referer">
        
    </a>
</p>
```

---

<sup>60</sup> **Cross-browsing** è la capacità di un sito Web, un'applicazione Web, del codice HTML o di uno script di essere in grado di funzionare con qualsiasi web browser.

<sup>61</sup> <http://validator.w3.org/> - pagina del W3C che si occupa della validazione del codice (X)HTML.

<sup>62</sup> <http://jigsaw.w3.org/css-validator/> - pagina del W3C che si occupa della validazione del codice CSS.

<sup>63</sup> <http://www.w3.org/2003/12/semantic-extractor.html> - pagina del W3C che si occupa della validazione della semantica delle pagine Web.

### 3.4.2 Access Key

Un *access key* (*tasto di accesso* in italiano) è una combinazione di tasti che una volta premuta in un web browser permette di "saltare" immediatamente ad un sezione della pagina o più in generale del sito Web. Questi collegamenti permettono un agile navigazione da tastiera e consentono di migliorare l'accessibilità delle pagine Web che li adottano.

Sono stati introdotti per la prima volta nel 1999 ed hanno raggiunto in breve tempo il supporto di tutti i principali Web browser.

#### Specificare un access key

Specificare un access key è molto semplice, basta infatti utilizzare l'attributo apposito *accesskey* all'interno di un collegamento alla pagina a cui esso farà riferimento.

In questo esempio è mostrato un possibile collegamento alla home page:

```
<a href="index.html" accesskey="h">Home</a>
```

In linea di massima, e per una maggiore immediatezza, è opportuno evidenziare in qualche modo il carattere che contraddistingue l'access key appunto, mediante l'utilizzo del tag *<em>* il quale ha il compito di enfatizzare parti di testo, oppure tramite la sottolineatura del carattere, questa volta con il tag *<u>*.

```
<a href="index.html" accesskey="h"><em>H</em>ome</a>
```

Un altro ottimo modo, forse il più corretto ed intuitivo è mediante l'utilizzo dei CSS. Infatti basterà inserire questa riga di codice all'interno del proprio foglio di stile:

```
* [accesskey]:after {content: ' [' attr(accesskey) '] '}
```

per ottenere il risultato seguente:

Home [h]

#### Mappatura standard di access key

Nell'anno 2004 il governo del Regno Unito ha raccomandato l'uso dei seguenti access key, con lo scopo di creare una sorta di linea guida da seguire al momento dell'impostazione di queste "scorciatoie" all'interno dei propri siti Web.

- S - Skip navigation
- 1 - Home page
- 2 - Novità
- 3 - Mappa del sito
- 4 - Ricerca
- 5 - Domande Frequenti - Frequently Asked Questions (FAQ)
- 6 - Aiuto
- 7 - Presentazione reclami
- 8 - Termini e condizioni
- 9 - Feedback form
- 0 - Dettagli sugli Access key

Chiaramente questa vuole essere solo una linea guida e soprattutto non implica l'inserimento di tutti questi access key all'interno di ogni singolo sito Web. C'è da tener presente però che ad esempio la scorciatoia *skip navigation* può risultare parecchio utile se la navigazione della pagina è fatta mediante uno *screen reader*, in questo caso, tramite una veloce combinazione di tasti, è possibile passare alla navigazione della sezione successiva, senza dover attendere che la lettura di quella corrente sia necessariamente terminata.

### **Access nei principali browser**

Soltamente il funzionamento degli access key è dato mediante la combinazione di un tasto di accesso (*alt* computer che utilizzano *Windows* e *ctrl* invece su macchine *Macintosh*) più il tasto corrispondete alla "scorciatoia".

Qui di seguito sono riportate le combinazioni di tasti necessarie per l'attivazione degli access key per i principali Web browser (Google Chrome<sup>64</sup>, Internet Explorer<sup>65</sup>, Mozilla Firefox<sup>66</sup>, Opera<sup>67</sup> e Safari<sup>68</sup>):

| Browser           | Combinazione |                  | Note   |
|-------------------|--------------|------------------|--|
| Google Chrome     | alt (win)    | alt + ctrl (mac) | Compatibile dalla versione 3 o successive  |
| Internet Explorer | alt          |                  | Per confermare necessita la pressione di enter                                     |
| Mozilla Firefox   | alt + shift  |                  |  |
| Opera             | shift + esc  |                  | I tasti devono essere rilasciati prima della selezione dell'access key desiderato. |
| Safari            | alt (win)    | alt + ctrl (mac) | Sequenza di tasti relativa alla versione 4   |

---

<sup>64</sup> <http://www.google.com/chrome> - pagina ufficiale del browser Google Chrome

<sup>65</sup> <http://www.microsoft.com/nz/ie/> - pagina ufficiale del browser Internet Explorer

<sup>66</sup> <http://www.mozilla.com/firefox/> - pagina ufficiale del browser Mozilla Firefox

<sup>67</sup> <http://www.opera.com/> - pagina ufficiale del browser Opera

<sup>68</sup> <http://www.apple.com/safari/> - pagina ufficiale del browser Safari

### 3.4.3 Access Bar

L'access bar è un validissimo strumento per aumentare l'accessibilità del proprio sito internet. Essa infatti permette di modificare, nel modo più semplice ed intuitivo possibile, l'aspetto di una pagina Web per consentire una miglior accessibilità a persone con svariate disabilità. Per ottenere una maggior efficacia va posizionata in un area ad alta visibilità, l'header o una barra laterale sono ottime soluzioni, deve avere una grafica lineare e le funzioni devono essere mostrate in modo chiaro.

#### Funzionalità base

Soltamente questa classe di funzioni è presente in tutte le access bar e consiste in un primo passo base verso l'accessibilità.

##### *Colore<sup>69</sup>*

Cambiando la colorazione rispettivamente del *testo* e dello *sfondo* è possibile creare combinazioni e contrasti adatti a creare una maggior leggibilità del testo da parte di utenti con problemi visivi. Oltre ai classici colori per evidenziare il contrasto, giallo su nero può forse essere l'esempio più lampante, è opportuno prestare attenzione anche a problemi visivi quali i *scotopic sensitive*, persone per le quali è possibile distinguere differenze di brillantezza ma non di cromaticità, di conseguenza la loro lettura è agevolata da testi con sfondi dalle colorazioni tenui.

##### *Dimensione*

La possibilità di variare la dimensione del testo riveste un ruolo fondamentale per garantire una visione ottimale del testo, consentendo un suo ingrandimento e rimpicciolimento a piacere.

Esiste una funzione analoga presente nativamente nella maggior parte dei browser Web chiamata *zoom-in/zoom-out*, ma può risultare macchinosa e poco intuitiva specialmente per chi non è estremamente pratico con i computer. Inoltre questa funzione solitamente ha il compito di aumentare (o diminuire) proporzionalmente le dimensioni di qualsiasi elemento presente nella pagina e questo tende a snaturare le normali proporzioni del layout. Nel capitolo successivo si vedrà come attivare questa modifica solo relativamente alle dimensioni del testo.

##### *Stile del testo*

Questa funzione riveste un ruolo più marginale rispetto alle due precedenti in quanto solitamente si sceglie di cambiare lo stile del testo (*font style*) solamente per una questione di abitudine o di maggior gradimento di un font rispetto ad un altro. C'è però da aggiungere che certe categorie di disabilità trovano giovamento nella lettura di un testo con font della famiglia *sans-serif* (letteralmente senza grazie), in quanto la rappresentazione del carattere risulta più semplice ed il suo riconoscimento risulta più immediato.

#### Funzionalità avanzate

Queste funzionalità sono per la maggior parte dei casi un "surplus" per la navigabilità della pagina Web, esse infatti, oltre ad essere più complicate da sviluppare, e solitamente tendono ad appesantire la struttura delle pagine in quanto necessitano di contenuti multimediali ad esse collegati.

---

<sup>69</sup> <http://www2.comune.venezia.it/letturagevolata/files/questionedileggibilita.pdf> - per un approfondimento legato alla leggibilità e la scelta dei colori si rimanda alla lettura del documento "Questione di leggibilità" rilasciato dal comune di Venezia.

### *Screen reader*

In alcune access bar è presente una funzione di screen reader, naturalmente questa può portare un incremento dell'accessibilità non solo nei confronti degli utenti non vedenti ma anche da quelli affetti da dislessia.

### *Download multimediali*

Soltanente collegati alla funzione di screen reader, permettono lo scaricamento sul proprio computer del testo letto dallo screen reader. In alcuni casi è possibile anche trovare la funzione di download dell'intero testo della pagina, anche se questo caso non ricopre un ruolo centrale per quanto riguarda l'accessibilità Web.

### **Alcuni esempi di access bar:**



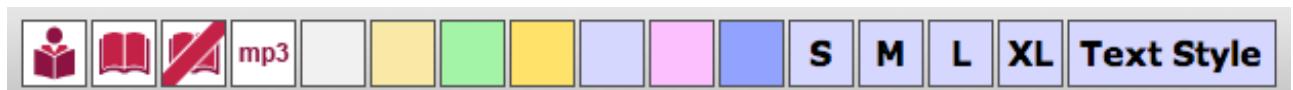
Access Bar minimale, consente di modificare la grandezza del testo ed è possibile scegliere tra due fogli di stile, il classico e quello ad alto contrasto.

*Access bar del sito ufficiale di Assistive Solutions Ltd<sup>70</sup>.*



Access Bar che consente la modifica della grandezza, dello stile e del colore del testo, del colore dello sfondo ed è inoltre disponibile una funzione di *reset* alle impostazioni standard.

*Access bar del sito ufficiale della British Dyslexia Association<sup>71</sup> (BDA).*



Access bar con funzioni avanzate, oltre alla possibilità di cambio del colore di sfondo, grandezza e stile del testo è presente la funzione di screen reader e di download del testo letto in formato mp3.

*Access bar della pagina ufficiale dell'Association of Dislexia Specialists in Higher Education<sup>72</sup> (Adshe).*

<sup>70</sup> <http://www.assistiveitsolutions.com/> - pagina ufficiale della compagnia Assistive Solutions Ltd.

<sup>71</sup> <http://www.bdadyslexia.org.uk/> - pagina ufficiale della British Dyslexia Association

<sup>72</sup> <http://adshe.org.uk/> - pagina ufficiale della Association of Dyslexia Specialists in Higher Education (ADSHE)

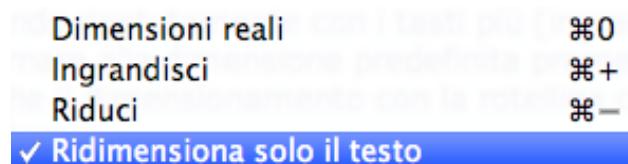
### 3.4.4 Accessibilità via browser

Anche i normali browser Web permettono di modificare i metodi di visualizzazione delle pagine Web, senza necessariamente aver bisogno di strumenti creati appositamente dagli sviluppatori di tali pagine.

(Nota: In questa sezione verranno presi in considerazioni solo i browser Internet Explorer, Mozilla Firefox e Safari.)

#### Ridimensionare il carattere

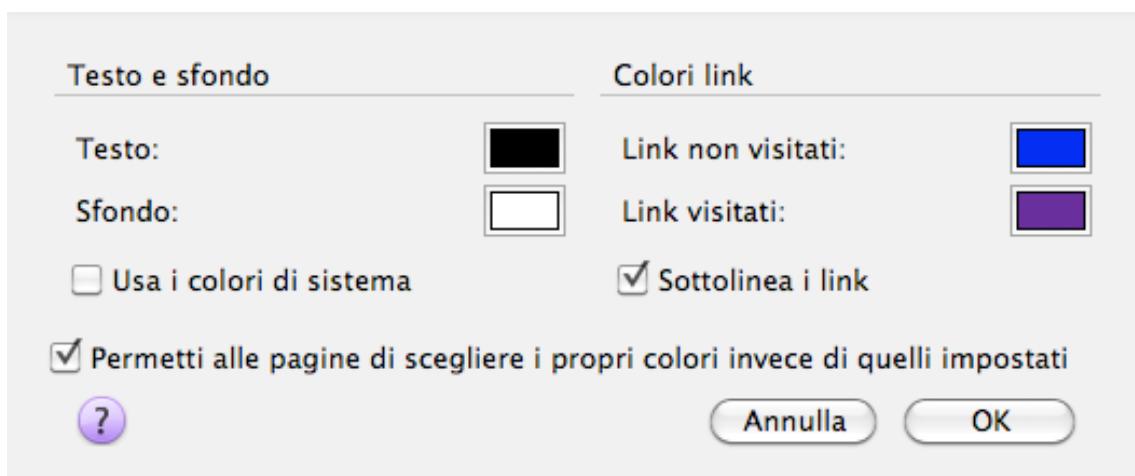
Anche da un normale browser è possibile ridimensionare il carattere. L'unica accortezza che bisogna avere è quella di essere sicuri di aver spuntato l'opzione di ridimensionare solo testo (come si può notare nell'immagine di esempio che fa riferimento al Web Browser *Safari*), dopo di che sarà possibile con le apposite funzioni *ingrandisci* e *riduci* modificare solamente la grandezza appunto del testo. Solitamente, oltre che dalle normali voci di menu sarà possibile effettuare questo ridimensionamento mediante tasti rapidi (*command + e command -* su Mac OS e *ctrl + e ctrl -* su Windows) oppure mediante la rotellina del mouse (sempre con tasto *command* su Mac OS e *ctrl* su Windows).



#### Cambiare i colori di default

Oltre a cambiare le dimensioni del carattere è possibile modificare i colori di *default* all'interno delle pagine Web visitabili. Come ad esempio il colore del testo, dello sfondo e dei link (già visitati e non ancora).

Nel esempio, che fa riferimento al browser *Mozilla Firefox* è possibile vedere mostrate la schermata appositamente dedicata a questa funzione, raggiungibile dal menu *preferenze*, sezione *contenuto*. Qui è anche possibile cambiare il font di *default*.



#### Impostare un proprio fogli di stile

Per una modifica più libera e sostanziale alla struttura delle pagine che si possono visitare è possibile impostare dei propri fogli di stile durante la navigazione. In *Internet Explorer* e

*Safari* questa operazione non necessita nulla di particolarmente complicato, è sufficiente andare nella sezione apposita nelle preferenze del browser e selezionare il proprio foglio di stile desiderato, invece per quanto riguarda *Mozilla Firefox* questa operazione risulta un po' più complicata in quanto si dovrà modificare manualmente il foglio di stile di *default* del browser.

Nell'esempio è possibile notare la finestra di *accesso facilitato* del browser *internet Explorer*, in questo caso selezionando *Formatta i documenti utilizzando il foglio di stile dell'utente* e successivamente impostando tale file, sarà possibile modificare il normale aspetto delle pagine che saranno visualizzate tramite il browser.



## Toolbar di verifica dell'accessibilità

Le due principali *toolbar* che consentono di verificare l'accessibilità di una pagina Web sono principalmente una per *Internet Explorer* ed è rilasciata dal *Web Accessibility Tools Consortium*<sup>73</sup> e l'altra per *Mozilla Firefox* ed è presente come *add-on*<sup>74</sup> dal nome *Firefox Accessibility Extension*<sup>75</sup> sviluppata dal *Illinois Center for Information Technology and Web Accessibility*<sup>76</sup>.



Entrambe le toolbar dell'accessibilità sono state create con lo scopo di supportare la verifica manuale di un gran numero di aspetti riguardanti l'accessibilità di una pagina Web. Prevedono diverse funzionalità, riassumibili nelle seguenti aree:

- Identificare le componenti di una pagina Web.
- Fornire l'accesso a visualizzazioni alternative del contenuto di una pagina.
- Facilitare l'uso di applicazioni online di altri sviluppatori.

<sup>73</sup> <http://www.wat-c.org/> - pagina ufficiale del Web Accessibility Tools Consortium

<sup>74</sup> <https://addons.mozilla.org/en-US/firefox/addon/5809> - pagina di riferimento per il download dell'estensione

<sup>75</sup> <http://firefox.cita.uiuc.edu/> - pagina di riferimento del progetto Firefox Accessibility Extension

<sup>76</sup> <http://www.cita.uiuc.edu/> - pagina ufficiale del Illinois Center for Information Technology and Web Accessibility

### **3.5 Accessibilità Web - Conclusione**

Riassumendo, per creare Web accessibile, è sufficiente rispettare delle semplici regole, sintetizzabili nei seguenti punti:

- (X)HTML valido (secondo gli standard del W3C) per il contenuto della pagina.
- CSS valido (secondo gli standard del W3C) per il layout della pagina.
- Almeno un livello di adempimento WAI-AA (preferibilmente AAA) in accordo alla specifica WCAG del WAI.
- Contenuti alternativi per ogni contenuto multimediale utilizzato all'interno del sito (video, flash, audio, etc).
- Una versione ad alto contrasto del testo del sito per consentire la lettura a persone con problemi di vista.
- Access keys predisposti per il controllo della pagina.
- Web Markup semantici.
- Rispetto di tutte le norme legali vigenti sul territorio per il quale il sito Web è destinato.

Una buona accessibilità permette una maggior fruibilità dei contenuti e di conseguenza maggior capillarità nella diffusione delle informazioni presenti nelle proprie pagine Web. L'importanza dell'Accessibilità Web, e dell'Assistive Technology in generale, consiste nella sua capacità di "abbattere" le barriere informatiche, permettendo un notevole sviluppo sia del Web che dell'informatizzazione in generale e, come diretta conseguenza, uno sviluppo della società dato dal massimo accesso alle informazioni da parte di tutti indistintamente.

### 3.6 Legislazione Italiana

L’Italia seguendo le indicazioni dell’Unione Europea e basandosi sia sulle direttive del W3C, presenti nella specifica WCAG, e sulla *Section 508*<sup>77</sup> del Governo degli Stati Uniti, ha creato la *Legge 04/2004*, detta anche *Legge Stanca*<sup>78</sup>.

Questa legge si applica a tutte le pubbliche amministrazioni e a tutte quelle aziende che hanno una prevalente capitalizzazione pubblica o che sono concessionarie di servizi pubblici.

La legge Stanca purtroppo presenta tre grossi limiti:

- Il *campo di attuazione*: essa infatti si applica solamente al settore pubblico, non entrando in merito per quanto concerne il privato.
- L'*applicabilità*: in quanto per essere applicata necessita la presenza di un contratto, tra soggetto erogatore e usufruente del servizio. Conseguentemente tale contratto, qualora la legge non sia rispettata, potrà considerarsi nullo.
- Il *controllo*: ovvero la mancanza di un vero e proprio organo di controllo atto a verificare, e di conseguenza far rispettare, l’insieme di tecniche e metodologie preso in esame da questa legge.

Il *Decreto Ministeriale dell’8 Luglio 2005* rende a tutti gli effetti operativa la Legge Stanca. Questo documento si focalizza in modo particolare sull’accessibilità informatica, suddividendola in tre grandi aree tematiche (*Web, hardware e software*) e delinea una serie di requisiti tecnici che dovrebbero essere sempre rispettati:

- **Requisiti Web**: riguardanti l’accessibilità delle applicazioni basate su tecnologie internet.
- **Requisiti Hardware**: riguardanti l’accessibilità per i personal computer di tipo desktop e portatili.
- **Requisiti Software**: riguardanti l’accessibilità per l’ambiente operativo, le applicazioni e i prodotti a scaffale.

In aggiunta si trovano specificati anche i criteri essenziali su cui basare la verifica soggettiva dei siti Web e delle applicazioni realizzate con tecnologie Internet.

#### 3.6.1 Requisiti Web

*Requisiti tecnici di accessibilità delle applicazioni basate su tecnologie internet*<sup>79</sup>

*Requisito n. 1*

Realizzare le pagine e gli oggetti al loro interno utilizzando tecnologie definite da grammatiche formali pubblicate nelle versioni più recenti disponibili quando sono supportate dai programmi utente. Utilizzare elementi ed attributi in modo conforme alle specifiche, rispettandone l’aspetto semantico. In particolare, per i linguaggi a marcatori

<sup>77</sup> <http://www.section508.gov/> - sezione 508 del Governo degli Stati Uniti

<sup>78</sup> Per il testo integrale della legge si rimanda alla lettura dell’Appendice C ad essa dedicata.

<sup>79</sup> <http://www.pubbliaccesso.gov.it/normative/DM080705-a.htm> - link al D.M. 8 luglio 2005 - Allegato A

HTML (Hypertext Markup Language) e XHTML (eXtensible HyperText Markup Language):

- a) per tutti i siti di nuova realizzazione utilizzare almeno la versione 4.01 dell'HTML o preferibilmente la versione 1.0 dell'XHTML, in ogni caso con DTD (Document Type Definition - Definizione del Tipo di Documento) di tipo Strict;
- b) per i siti esistenti, in sede di prima applicazione, nel caso in cui non sia possibile ottemperare al punto a) è consentito utilizzare la versione dei linguaggi sopra indicati con DTD Transitional, ma con le seguenti avvertenze:
  - 1) evitare di utilizzare, all'interno del linguaggio a marcatori con il quale la pagina è realizzata, elementi ed attributi per definirne le caratteristiche di presentazione della pagina (per esempio, caratteristiche dei caratteri del testo, colori del testo stesso e dello sfondo, ecc.), ricorrendo invece ai Fogli di Stile CSS (Cascading Style Sheets) per ottenere lo stesso effetto grafico;
  - 2) evitare la generazione di nuove finestre; ove ciò non fosse possibile, avvisare esplicitamente l'utente del cambiamento del focus;
  - 3) pianificare la transizione dell'intero sito alla versione con DTD Strict del linguaggio utilizzato, dandone comunicazione alla Presidenza del Consiglio dei Ministri – Dipartimento per l'innovazione e le tecnologie e al Centro nazionale per l'informatica nella pubblica amministrazione.

#### *Requisito n. 2*

Non è consentito l'uso dei frame nella realizzazione di nuovi siti. In sede di prima applicazione, per i siti Web esistenti già realizzati con frame è consentito l'uso di HTML 4.01 o XHTML 1.0 con DTD frameset, ma con le seguenti avvertenze:

- a) evitare di utilizzare, all'interno del linguaggio a marcatori con il quale la pagina è realizzata, elementi ed attributi per definirne le caratteristiche di presentazione della pagina (per esempio, caratteristiche dei caratteri del testo, colori del testo stesso e dello sfondo, ecc.), ricorrendo invece ai Fogli di Stile CSS (Cascading Style Sheets) per ottenere lo stesso effetto grafico;
- b) fare in modo che ogni frame abbia un titolo significativo per facilitarne l'identificazione e la navigazione; se necessario, descrivere anche lo scopo dei frame e la loro relazione;
- c) pianificare la transizione a XHTML almeno nella versione 1.0 con DTD Strict dell'intero sito dandone comunicazione alla Presidenza del Consiglio dei Ministri – Presidenza del Consiglio dei Ministri – Dipartimento per l'innovazione e le tecnologie e al Centro nazionale per l'informatica nella pubblica amministrazione.

#### *Requisito n. 3*

Fornire una alternativa testuale equivalente per ogni oggetto non di testo presente in una pagina e garantire che quando il contenuto non testuale di un oggetto cambia dinamicamente vengano aggiornati anche i relativi contenuti equivalenti predisposti; l'alternativa testuale equivalente di un oggetto non testuale deve essere commisurata alla funzione esercitata dall'oggetto originale nello specifico contesto.

#### *Requisito n. 4*

Garantire che tutti gli elementi informativi e tutte le funzionalità siano disponibili anche in assenza del particolare colore utilizzato per presentarli nella pagina.

#### *Requisito n. 5*

Evitare oggetti e scritte lampeggianti o in movimento le cui frequenze di intermittenza possano provocare disturbi da epilessia fotosensibile o disturbi della concentrazione, ovvero possano causare il malfunzionamento delle tecnologie assistive utilizzate; qualora

esigenze informative richiedano comunque il loro utilizzo, avvertire l'utente del possibile rischio prima di presentarli e predisporre metodi che consentano di evitare tali elementi.

*Requisito n. 6*

Garantire che siano sempre distinguibili il contenuto informativo (foreground) e lo sfondo (background), ricorrendo a un sufficiente contrasto (nel caso del testo) o a differenti livelli sonori (in caso di parlato con sottofondo musicale); evitare di presentare testi in forma di immagini; ove non sia possibile, ricorrere agli stessi criteri di distinguibilità indicati in precedenza.

*Requisito n. 7*

Utilizzare mappe immagine sensibili di tipo lato client piuttosto che lato server, salvo il caso in cui le zone sensibili non possano essere definite con una delle forme geometriche predefinite indicate nella DTD adottata.

*Requisito n. 8*

In caso di utilizzo di mappe immagine lato server, fornire i collegamenti di testo alternativi necessari per ottenere tutte le informazioni o i servizi raggiungibili interagendo direttamente con la mappa.

*Requisito n. 9*

Per le tabelle dati usare gli elementi (marcatori) e gli attributi previsti dalla DTD adottata per descrivere i contenuti e identificare le intestazioni di righe e colonne.

*Requisito n. 10*

Per le tabelle dati usare gli elementi (marcatori) e gli attributi previsti nella DTD adottata per associare le celle di dati e le celle di intestazione che hanno due o più livelli logici di intestazione di righe o colonne.

*Requisito n. 11*

Usare i fogli di stile per controllare la presentazione dei contenuti e organizzare le pagine in modo che possano essere lette anche quando i fogli di stile siano disabilitati o non supportati.

*Requisito n. 12*

La presentazione e i contenuti testuali di una pagina devono potersi adattare alle dimensioni della finestra del browser utilizzata dall'utente senza sovrapposizione degli oggetti presenti o perdita di informazioni tali da rendere incomprensibile il contenuto, anche in caso di ridimensionamento, ingrandimento o riduzione dell'area di visualizzazione o dei caratteri rispetto ai valori predefiniti di tali parametri.

*Requisito n. 13*

In caso di utilizzo di tabelle a scopo di impaginazione, garantire che il contenuto della tabella sia comprensibile anche quando questa viene letta in modo linearizzato e utilizzare gli elementi e gli attributi di una tabella rispettandone il valore semantico definito nella specifica del linguaggio a marcatori utilizzato.

*Requisito n. 14*

Nei moduli (form), associare in maniera esplicita le etichette ai rispettivi controlli, posizionandole in modo che sia agevolata la compilazione dei campi da parte di chi utilizza le tecnologie assistive.

*Requisito n. 15*

Garantire che le pagine siano utilizzabili quando script, applet, o altri oggetti di programmazione sono disabilitati oppure non supportati; ove ciò non sia possibile fornire una spiegazione testuale della funzionalità svolta e garantire una alternativa testuale equivalente, in modo analogo a quanto indicato nel requisito n. 3.

*Requisito n. 16*

Garantire che i gestori di eventi che attivano script, applet o altri oggetti di programmazione o che possiedono una propria specifica interfaccia, siano indipendenti da uno specifico dispositivo di input.

*Requisito n. 17*

Garantire che le funzionalità e le informazioni veicolate per mezzo di oggetti di programmazione, oggetti che utilizzano tecnologie non definite da grammatiche formali pubblicate, script e applet siano direttamente accessibili.

*Requisito n. 18*

Nel caso in cui un filmato o una presentazione multimediale siano indispensabili per la completezza dell'informazione fornita o del servizio erogato, predisporre una alternativa testuale equivalente, sincronizzata in forma di sotto-titolazione o di descrizione vocale, oppure fornire un riassunto o una semplice etichetta per ciascun elemento video o multimediale tenendo conto del livello di importanza e delle difficoltà di realizzazione nel caso di trasmissioni in tempo reale.

*Requisito n. 19*

Rendere chiara la destinazione di ciascun collegamento ipertestuale (link) con testi significativi anche se letti indipendentemente dal proprio contesto oppure associare ai collegamenti testi alternativi che possiedano analoghe caratteristiche esplicative, nonché prevedere meccanismi che consentano di evitare la lettura ripetitiva di sequenze di collegamenti comuni a più pagine.

*Requisito n. 20*

Nel caso che per la fruizione del servizio erogato in una pagina è previsto un intervallo di tempo predefinito entro il quale eseguire determinate azioni, è necessario avvisare esplicitamente l'utente, indicando il tempo massimo consentito e le alternative per fruire del servizio stesso.

*Requisito n. 21*

Rendere selezionabili e attivabili tramite comandi da tastiere o tecnologie in emulazione di tastiera o tramite sistemi di puntamento diversi dal mouse i collegamenti presenti in una pagina; per facilitare la selezione e l'attivazione dei collegamenti presenti in una pagina è necessario garantire che la distanza verticale di liste di link e la spaziatura orizzontale tra link consecutivi sia di almeno 0,5 em, le distanze orizzontale e verticale tra i pulsanti di un modulo sia di almeno 0,5 em e che le dimensioni dei pulsanti in un modulo siano tali da rendere chiaramente leggibile l'etichetta in essi contenuta.

*Requisito n. 22*

Per le pagine di siti esistenti che non possano rispettare i su elencati requisiti (pagine non accessibili), in sede di prima applicazione, fornire il collegamento a una pagina conforme a tali requisiti, recante informazioni e funzionalità equivalenti a quelle della pagina non accessibile ed aggiornata con la stessa frequenza, evitando la creazione di pagine di solo

testo; il collegamento alla pagina conforme deve essere proposto in modo evidente all'inizio della pagina non accessibile.

### **3.6.2 Requisiti Hardware**

*Requisiti tecnici di accessibilità per i personal computer di tipo desktop e portatili<sup>80</sup>*

*Requisito n. 1*

Il computer deve potersi collegare mediante canali standard a sistemi di accensione remota.

*Requisito n. 2*

I tasti e i pulsanti devono essere raggiungibili ed operabili con minima abilità e con una forza massima di 2,3 Kg (pari a circa 22,2 N).

*Requisito n. 3*

I tasti e i pulsanti devono essere tattilmente percepibili, senza necessità di attivarli.

*Requisito n. 4*

In presenza della funzionalità di ripetizione dei tasti, l'intervallo di tempo sia per la prima ripetizione che per le ripetizioni successive, deve essere configurabile in almeno 2 secondi.

*Requisito n. 5*

Il diverso stato di attivazione dei tasti selezionati o bloccati deve essere percepibile, oltre che visivamente, anche attraverso il tatto o l'udito.

*Requisito n. 6*

Deve essere presente almeno una porta di comunicazione conforme agli standard industriali.

*Requisito n. 7*

Qualora venga utilizzata una forma di identificazione biometrica, deve essere fornita una forma alternativa di identificazione.

### **3.6.3 Requisiti Software**

*Requisiti tecnici di accessibilità per l'ambiente operativo, le applicazioni e i prodotti a scaffale<sup>81</sup>*

*Requisito n. 1*

Le funzioni previste dall'interfaccia utente devono poter essere attivate anche attraverso comandi da tastiera nei casi in cui possa essere fornita una descrizione della funzione stessa o del risultato della sua esecuzione.

---

<sup>80</sup> <http://www.pubbliaccesso.gov.it/normative/DM080705-C.htm> - link al D.M. 8 luglio 2005 - Allegato C

<sup>81</sup> <http://www.pubbliaccesso.gov.it/normative/DM080705-D.htm> - link al D.M. 8 luglio 2005 - Allegato D

*Requisito n. 2*

Comandi e funzionalità dell'interfaccia utente non devono limitare o disabilitare le caratteristiche e le funzionalità di accessibilità dell'ambiente operativo documentate e rese disponibili dal produttore dell'ambiente stesso.

*Requisito n. 3*

L'applicazione deve rendere disponibili sufficienti informazioni, quali gli elementi identificativi, le operazioni possibili e lo stato, sugli oggetti contenuti nell'interfaccia utente affinché le tecnologie assistive possano identificarli interpretandone le funzionalità.

*Requisito n. 4*

Nel caso di simboli grafici utilizzati per identificare controlli, indicatori di stato o altri elementi di programma, il significato assegnato a tali simboli deve essere coerente nell'ambito dell'intera applicazione, ivi compresa l'interfaccia utente.

*Requisito n. 5*

Le informazioni di tipo testuale devono essere fornite utilizzando le funzionalità dell'ambiente operativo previste per la visualizzazione del testo; in particolare devono essere disponibili il contenuto testuale, la locazione del punto di inserimento e gli attributi del testo.

*Requisito n. 6*

L'applicazione che utilizza segnalazioni audio deve prevedere una funzionalità equivalente di tipo visivo, seguendo le eventuali convenzioni dell'ambiente operativo.

*Requisito n. 7*

Per fornire informazioni, per indicare o per richiedere azioni non devono essere utilizzati unicamente animazioni, elementi grafici o sonori e differenze di colori.

*Requisito n. 8*

Le applicazioni non devono sovrapporsi alle scelte effettuate dall'utente riguardo a livelli di contrasto, colori ed altri attributi di visualizzazione.

*Requisito n. 9*

L'interfaccia utente non deve contenere elementi di testo, oggetti o altri elementi lampeggianti aventi una frequenza di intermittenza maggiore di 2Hz e minore di 55Hz.

*Requisito n. 10*

L'elemento attivo "focus" di una interfaccia utente deve essere chiaramente identificabile; la identificazione e la variazione del focus devono essere segnalate a livello di interfaccia di programmazione (API) affinché le tecnologie assistive possano gestirle; vanno altresì adeguatamente segnalati gli elementi che richiedono obbligatoriamente un'azione da parte dell'utente.

*Requisito n. 11*

La documentazione di supporto al prodotto e le caratteristiche di accessibilità devono essere rese disponibili anche in formato elettronico accessibile.

### **3.6.4 Verifica soggettiva**

*Criteri di valutazione per la verifica soggettiva dell'accessibilità delle applicazioni basate su tecnologie internet<sup>82</sup>*

Il decreto ministeriale prevede, in aggiunta ai già citati requisiti, una serie di criteri essenziali su cui basare una verifica soggettiva dei siti Web e delle applicazioni realizzate con tecnologie Internet.

#### *Percezione*

Informazioni e comandi necessari per l'esecuzione dell'attività devono essere sempre disponibili e percettibili;

#### *Comprendibilità*

Informazioni e comandi necessari per l'esecuzione delle attività devono essere facili da capire e da usare;

#### *Operabilità*

Informazioni e comandi devono consentire una scelta immediata della azione adeguata per raggiungere l'obiettivo voluto;

#### *Coerenza*

Simboli, messaggi e azioni devono avere lo stesso significato in tutto l'ambiente; salvaguardia della salute (safety): l'ambiente deve possedere caratteristiche idonee a salvaguardare il benessere psicofisico dell'utente;

#### *Sicurezza*

L'ambiente deve possedere caratteristiche idonee a fornire transazioni e dati affidabili, gestiti con adeguati livelli di sicurezza;

#### *Trasparenza*

L'ambiente deve comunicare all'utente lo stato, gli effetti delle azioni compiute e le informazioni necessarie per la corretta valutazione della dinamica dell'ambiente stesso;

#### *Apprendibilità*

L'ambiente deve possedere caratteristiche di utilizzo di facile e rapido apprendimento; aiuto e documentazione: funzioni di aiuto, quali le guide in linea, e documentazione relativa al funzionamento dell'ambiente devono essere di facili reperimento e connesse al compito svolto dall'utente;

#### *Tolleranza agli errori*

L'ambiente, pur configurandosi in modo da prevenire gli errori, ove questi, comunque, si manifestino, deve fornire appropriati messaggi che individuino chiaramente l'errore occorso e le azioni necessarie per superarlo;

#### *Gradevolezza*

L'ambiente deve possedere caratteristiche idonee a favorire e mantenere l'interesse dell'utente;

---

<sup>82</sup> <http://www.pubbliaccesso.gov.it/normative/DM080705-B.htm> - link al D.M. 8 luglio 2005 - Allegato B

## *Flessibilità*

L'ambiente deve tener conto delle preferenze individuali e dei contesti.

Questa analisi soggettiva si sviluppa su tre fasi:

- Un incipit portato avanti da un “esperto”.
- Un test diretto con l’utenza.
- L’elaborazione dei dati raccolti con la stesura di un rapporto conclusivo.

### **3.6.5 Il logo**

*Logo di accessibilità dei siti Web e delle applicazioni realizzate con tecnologie Internet<sup>83</sup>*

Con il Decreto Ministeriale è stato creato anche un logo che, come nel caso di quelli messi a disposizione dal W3C, attesta il grado di *accessibilità dei siti Web e delle applicazioni realizzate con tecnologie Internet*.

Il logo consiste nella sagoma di un personal computer unito a tre figure umane stilizzate le quali fuoriescono dallo schermo a braccia levate. Questo logo risponde ai vari livelli di accessibilità.

#### *Logo senza asterischi*

Il logo senza asterischi risponde al primo livello di accessibilità, legato alla conformità ai requisiti previsti per la verifica tecnica, ovvero vengono rispettati tutti i requisiti Web. Il logo utilizzabile per questo livello di accessibilità è il seguente:



#### *Logo con asterischi*

Consiste nello stesso disegno sopra descritto con l’aggiunta di asterischi. Questa tipologia di logo garantisce la conformità ai requisiti della verifica tecnica e l’ulteriore livello di qualità raggiunto dal sito a seguito dell’esito positivo della verifica soggettiva.

Tale livello di qualità è indicato da uno, due o tre asterischi riportati nella parte del logo raffigurante la tastiera del personal computer. Chiaramente un maggior numero di asterischi indica una maggior qualità della pagina Web. Di seguito è mostrato il logo con tre asterischi che attesta il raggiungimento del massimo livello di accessibilità Web.



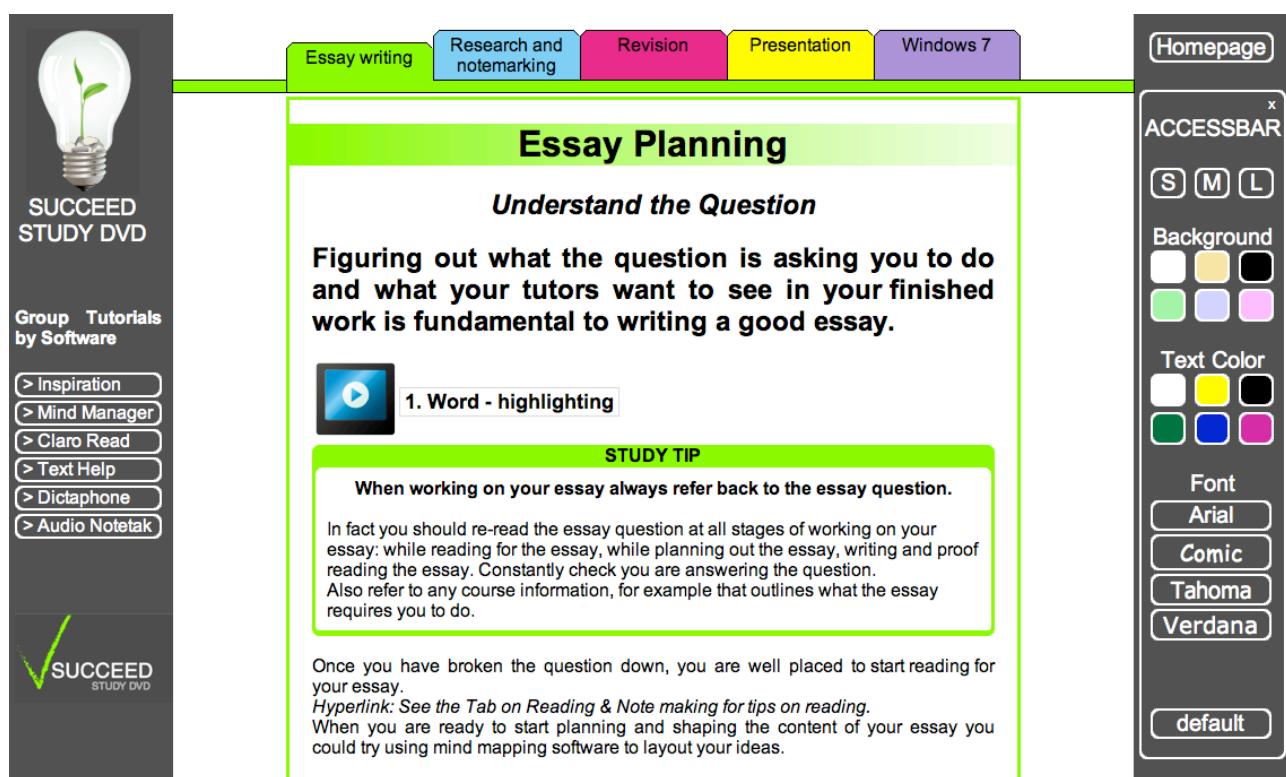
---

<sup>83</sup> <http://www.pubbliaccesso.gov.it/normative/DM080705-E.htm> - link al D.M. 8 luglio 2005 - Allegato E

## 4. Succeed

### 4.1 Introduzione

Il Software **Succeed** sviluppato durante la mia esperienza di stage presso la compagnia *Assistive Technology Ltd.*<sup>84</sup> è stato creato con la funzione di rappresentare un valido strumento per aiutare persone con difficoltà di apprendimento, una su tutte la dislessia, a studiare in modo semplice e al tempo stesso proficuo, tentando di assistere nell'insegnamento di un valido metodo di studio. Per far questo ci si è avvalsi di spiegazioni teoriche, spesso coadiuvate da registrazioni audio/video, della creazione di tools appositi e di software dedicato di terze parti.



(Nell'immagine uno screenshot di una schermata tipo del software)

Prima di addentrarsi nei particolari di cosa fa effettivamente il software Succeed e soprattutto di come è fatto è opportuno però dare una definizione di dislessia e di disturbi specifici di apprendimento.

<sup>84</sup> <http://www.assistiveitsolutions.com/> - pagina ufficiale della compagnia Assistive Solutions Ltd.

#### 4.1.1 Definizione di Dislessia<sup>8586</sup>

La *Dislessia* è un *Disturbo Specifico dell'Apprendimento (DSA)*. Con questo termine ci si riferisce ai soli disturbi delle abilità scolastiche ed in particolare a: *dislessia, disortografia, disgrafia e discalculia*.

La Dislessia Evolutiva è un disturbo che si manifesta nella difficoltà di apprendere a leggere nonostante un'istruzione idonea, un'intelligenza adeguata, un'integrità neurosensoriale e un ambiente socioculturale favorevole. Essa dipende da disabilità cognitive di base che sono frequentemente di origine costituzionale (world Federation of Neurology, cit. in Ellis, 1984).

La perdita della capacità di leggere in seguito a eventi patologici che comportano danni cerebrali viene invece definita *dislessia acquisita*. Mentre la *dislessia evolutiva* è un disturbo specifico dell'apprendimento e, come tale, si manifesta in bambini normodotati dal punto di vista intellettuivo.

La principale caratteristica di questa categoria è le sue specificità, ovvero il disturbo interessa uno specifico dominio di abilità (lettura, scrittura, calcolo) lasciando intatto il funzionamento intellettivo generale. Ciò significa che per avere una diagnosi di dislessia, il soggetto NON deve presentare: deficit di intelligenza, problemi ambientali o psicologici, deficit sensoriali o neurologici. Tale disturbo è determinato da un'alterazione neurobiologica che caratterizza i DSA (disfunzione nel funzionamento di alcuni gruppi di cellule deputate al riconoscimento delle lettere-parole e il loro significato).

La dislessia è una difficoltà che riguarda la capacità di leggere e scrivere in modo corretto e fluente. Leggere e scrivere sono considerati atti così semplici e automatici che risulta difficile comprendere la fatica di un bambino dislessico.

Purtroppo in Italia la dislessia è poco conosciuta, benché si calcoli che riguardi il 3-4% della popolazione scolastica (fascia della Scuola Primaria e Secondaria di primo grado). La dislessia non è causata da un deficit di intelligenza né da problemi ambientali o psicologici o da deficit sensoriali o neurologici.

Il soggetto dislessico può leggere e scrivere, ma riesce a farlo solo impegnando al massimo le sue capacità e le sue energie, poiché non può farlo in maniera automatica e perciò si stanca rapidamente, commette errori, rimane indietro, non impara. La dislessia si presenta in quasi costante associazione ad altri disturbi; questo fatto determina la marcata eterogeneità dei profili e l'espressività con cui i DSA si manifestano. La difficoltà di lettura può essere più o meno grave e spesso si accompagna a problemi nella scrittura: disortografia (cioè una difficoltà di tipo ortografico, nel 60% dei casi) e disgrafia (difficoltà nel movimento fino-motorio della scrittura, cioè una cattiva resa formale, nel 43% dei casi), nel calcolo (44% dei casi) e, talvolta, anche in altre attività mentali. Tuttavia questi soggetti sono intelligenti e, solitamente, vivaci e creativi.

---

<sup>85</sup> <http://www.aiditalia.org/> - riferimento all'Associazione Italiana Dislessia

<sup>86</sup> <http://www.dislessia.org/> - riferimento all'Associazione Italiana Dislessia - Sezione di Roma

#### **4.1.2 Struttura generale del software**

Il software è caratterizzato da cinque aree didattiche, a loro volta divise in sottosezioni:

- **Essay writing - scrittura di un riassunto**
  - Essay planning - progettazione del riassunto
  - Writing essay in Word - scrittura di un riassunto in un word-processor
  - Proofreading - revisione della bozza
  - Customising Word - customizzazione del testo
- **Research and notemarking - ricerca e appunti**
  - Reading - lettura
  - Online research - ricerca Online
  - Digital notetaking - appunti digitali
  - Notetaking in lectures - appunti nelle letture
- **Revision - revisione**
  - Time management - gestione del tempo
  - Visual revision - revisione visiva
  - Auditory revision - revisione audio
- **Presentation - presentazione**
  - Creating a presentation - creare una presentazione
  - Mind maps - mappe concettuale
  - Presentation techniques - tecniche di presentazione
- **Windows 7**
  - Keyboard shortcuts - scorciatoie da tastiera
  - Navigation - navigazione
  - Filing - archiviazione

Le cinque aree tematiche, con dei nomi molto eloquenti, delineano in linea di massima tutti gli aspetti ai quali ci si può trovare davanti al momento dello studio, ovvero la *creazione di elaborati, la lettura e la ricerca di materiale, la revisione del proprio lavoro, la sua presentazione ed infine una parte su come migliorare l'approccio all'area di lavoro*, nel nostro caso Microsoft Windows 7<sup>87</sup>.

All'interno del software ogni area tematica è contraddistinta da un colore, in modo da facilitarne il riconoscimento immediato.

I colori prescelti sono:

- Essay writing
- Research and notemarking
- Revision
- Presentation
- Windows 7

---

<sup>87</sup> <http://www.microsoft.com/windows/windows-7/> - pagina ufficiale sul sito Microsoft del SO Windows 7

All'interno del software è inoltre presente un'ulteriore sezione creata appositamente per contenere tutorial sui software, di terze parti, utilizzati durante le spiegazioni.

I programmi in questione sono:

- *Inspiration*<sup>88</sup>

Inspiration sviluppato da *Inspiration Software Inc.*<sup>89</sup> è un software creato per l'apprendimento e il ragionamento visivo, è anche popolare per la creazione di grafici organizzativi, come mappe concettuali, mappe mentali, diagrammi e creazione di *flow chart*.

- *MindManager*<sup>90</sup>

MindManager sviluppato da *Minjet Corporation* è un software per la creazione di mappe mentali e concettuali. Grazie al suo avanzato sistema di API può comunicare con parecchi programmi integrati all'interno del sistema operativo, quali word processor, foglio di calcolo e Web browser.

- *ClaroRead*<sup>91</sup>

ClaroRead sviluppato da *Claro Software* è un programma di *text-to-speech* che consente la lettura del testo presente a video.

- *TextHelp*<sup>92</sup>

Si tratta di un software con le caratteristiche di uno *Screen Reader*, è disponibile anche una versione per Web browser che consente di abilitare la funzione di *text-to-speech* per la lettura di pagine Web.

- *Dictaphone*

Questa sezione si occupa di quella categoria di software che permettono la registrazione di tracce audio. Infatti in questa sezione viene illustrato come registrare ed archiviare la propria voce per prendere appunti. Viene inoltre illustrato come successivamente utilizzare la funzione di riascolto per fini di studio o revisione.

- *Audio Notetaker*<sup>93</sup>

Audio Notetaker è un software che aiuta lo studente con la registrazione, la navigazione, l'annotazione e l'organizzazione dei propri appunti registrati digitalmente.

---

<sup>88</sup> <http://www.inspiration.com/Inspiration> - pagina ufficiale del software Inspiration

<sup>89</sup> <http://www.inspiration.com/> - pagina ufficiale dello sviluppatore Inspiration Software Inc.

<sup>90</sup> <http://www.mindjet.com/> - pagina ufficiale dello sviluppatore del software MindManager

<sup>91</sup> <http://www.clarosoftware.com/> - pagina ufficiale dello sviluppatore del software ClaroRead

<sup>92</sup> <http://www.texthelp.com/> - pagina ufficiale del programma TextHelp

<sup>93</sup> <http://www.audionotetaker.com/> - pagina ufficiale del software AudioNotetaker

## 4.2 Caratteristiche del software

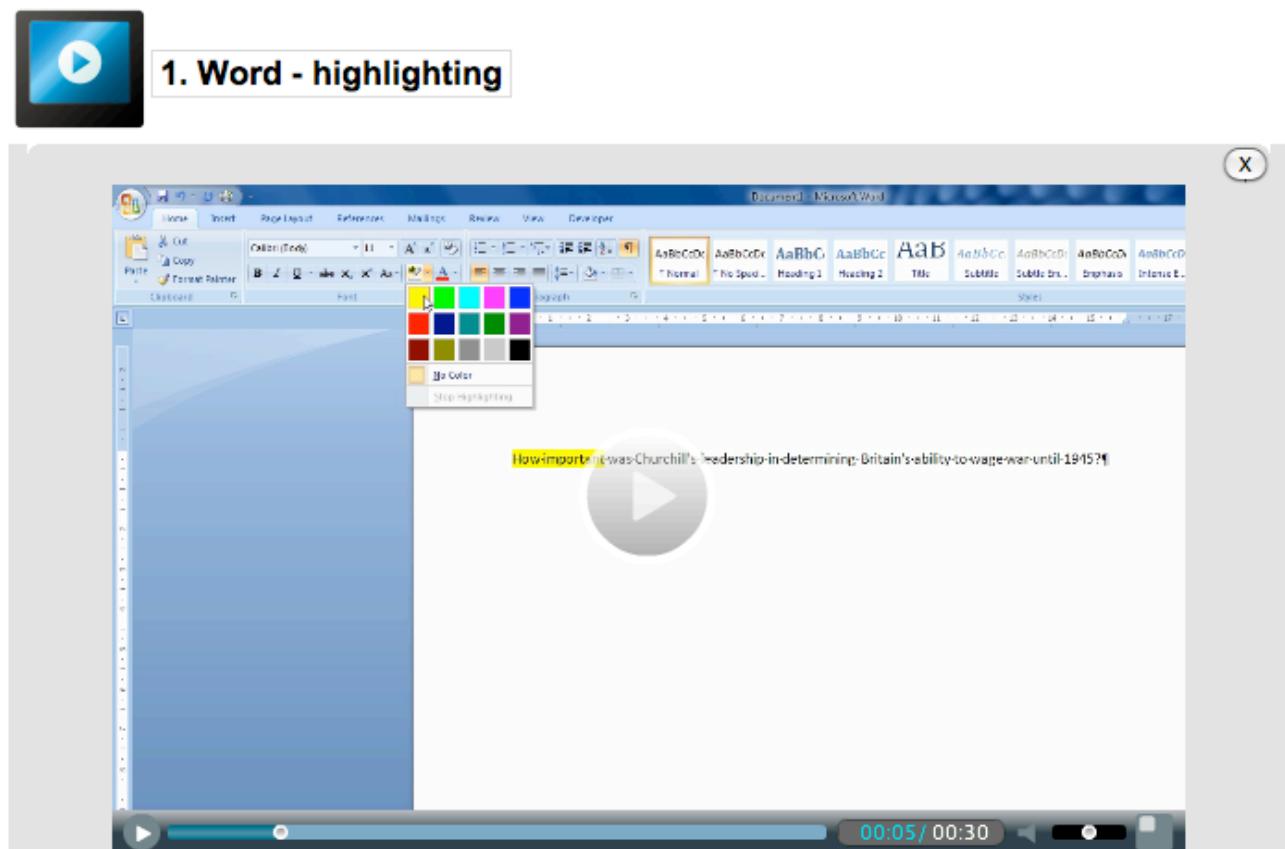
Il software si presenta allo studente come un normale sito Web, all'interno del quale si possono trovare:

- **Contenuti testuali**

Solitamente sempre brevi e concisi, dato che si tratta di soggetti con difficoltà di concentrazione e attenzione non è buona norma utilizzare lunghe perifrasi o giri di parole. Le spiegazioni teorico/testuali sono sempre corredate da dei video illustrativi.

- **Video illustrativi**

Spiegano passo a passo come svolgere i passaggi descritti nei contenuti testuali. Sono sempre corredate da una traccia audio che spiega come maggior accuratezza le procedure mostrate a schermo e da una trascrizione testuale di tale spiegazione. Questo per consentire la massima *accessibilità*. Per quanto riguarda i video è possibile vederli in modalità *full screen*.



V/O: Many people write essays that don't answer the question. So, start by highlighting the key terms in your question. Use a highlighter pen or you can do it in Word, like this: type in the question. On the Home Tab click on the highlighter tool, choose a colour and highlight the key terms.

Nell'esempio qui sopra è possibile vedere un contenuto video, corredata di intestazione con icona esplicativa e trascrizione testuale dell'audio.

- **Study Tip** - consigli per lo studio

Ovvero piccoli suggerimenti su come facilitare lo studio. Sono divisi dal resto del testo ed inseriti in piccoli box, che richiamano il colore della sezione in cui sono, per assicurargli una maggior visibilità a colpo d'occhio.

#### STUDY TIP

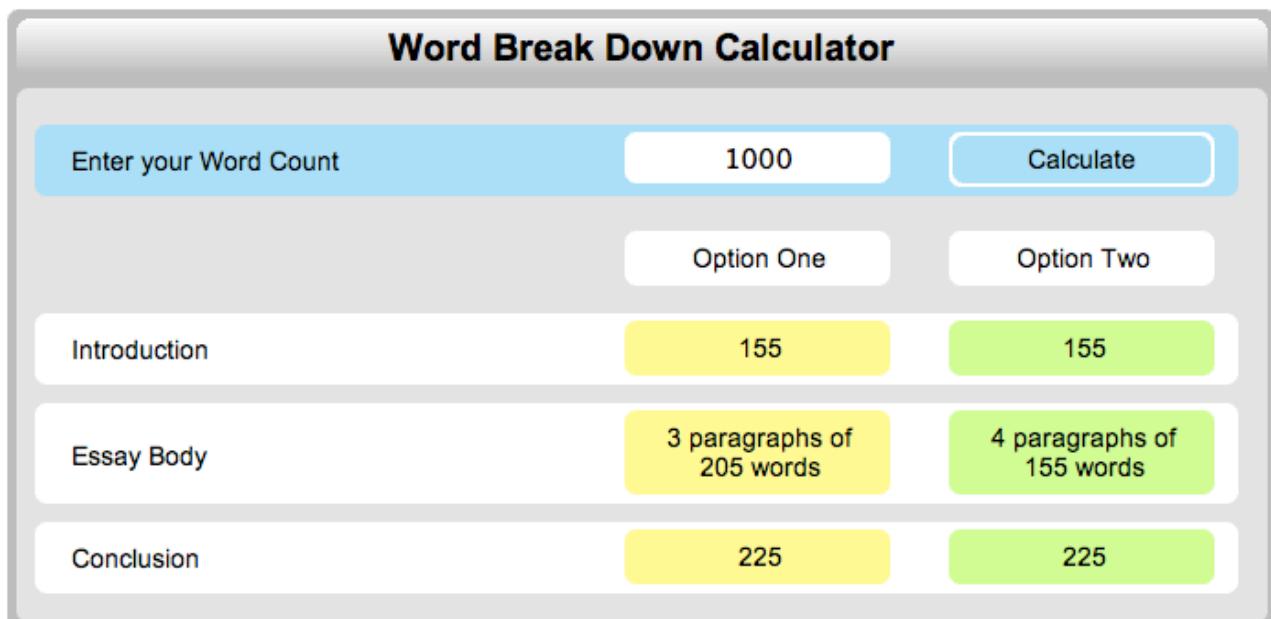
##### When working on your essay always refer back to the essay question.

In fact you should re-read the essay question at all stages of working on your essay: while reading for the essay, while planning out the essay, writing and proof reading the essay. Constantly check you are answering the question.

Also refer to any course information, for example that outlines what the essay requires you to do.

- **Tools**

All'interno delle singole sezioni si possono trovare *tools dinamici*. Questi strumenti hanno il compito di assistere lo studente nell'organizzazione del suo studio. Ad esempio si possono trovare *tools* che suggeriscono come suddividere le varie parti di un testo in maniera equilibrata oppure su come gestire il tempo in preparazione ad un test.



Nell'esempio è illustrato un *tool* che permette di calcolare come suddividere un testo in varie parti in base al numero di parole da cui dovrà essere composto.

Questi strumenti dinamici aiutano lo studente ad avvicinarsi e a familiarizzare maggiormente con il software stesso.

## 4.2.1 Caratteristiche tecniche del software

### 4.2.1.1 Tecnologie utilizzate

Le tecnologie utilizzate per la creazione di questo software sono principalmente quelle che si possono trovare in qualsiasi pagina Web. Si è deciso di sviluppare questo software seguendo la logica di una creazione di un sito Web per i seguenti motivi, che rispecchiano le *qualità interne* di un programma secondo l'*ingegneria del software*:

- **Maggior portabilità**

Ogni computer ha a disposizione un Web browser, questo consente al software Succeed di essere indipendente da ogni sistema operativo in uso sulla macchina.

- **Maggior usabilità**

Utilizzando la logica di un sito Web non vi è nessuna difficoltà a capire il funzionamento del software, in quanto sembrerà di navigare normalmente sul Web, questo consente una maggior immediatezza nell'uso del programma.

- **Maggior manutenibilità, riparabilità e evolvibilità**

La revisione e l'integrazione del software saranno semplicissime da attuare, in quanto non vi è nulla di complicato e criptico nella modifica di una pagina Web. Questo porta ad una notevole apertura a future modifiche e/o migliorie.

- **Maggior verificabilità**

Grazie alle metodologie, abbondantemente discusse nei capitoli precedenti, è possibile utilizzare tutte le tecniche per la verifica del codice di una pagina Web per controllare l'effettiva qualità funzionale del software.

Le tecnologie utilizzate sono le seguenti:

- **XHTML**

Il contenuto delle pagine è stato scritto in XHTML seguendo le specifiche delineate dal W3C.

- **CSS**

Lo stile del layout grafico del software è stato creato sotto forma di foglio di stile a cascata (CSS), in modo tale da avere un'indipendenza dal contenuto vero e proprio inserito in pagine XHTML e quindi una buona portabilità, anche in previsione di revisioni e modifiche future.

- **JavaScripts**

Gli scripts in JavaScript sono stati utilizzati principalmente per gestire gli effetti dinamici del software eseguendo le seguenti funzioni:

- Funzionamento dei *tools*.
- Modifica dell'aspetto delle varie pagine mediante la *access bar*.
- Effetti grafici di selezione nei vari menu.
- Effetti grafici quali l'arrotondamento degli angoli.
- Gestione degli effetti di comparsa/scomparsa.

- **Adobe<sup>94</sup> Flash<sup>95</sup>**

La tecnologia Flash è stata utilizzata per la gestione dei file video. Ogni file video è stato codificato in formato *FLV*, anche il player stesso è in formato Flash ma questa volta *Shockwave Flash (SWF)*.

Si è scelto di utilizzare Adobe Flash per la visualizzazione dei video in quanto si è ritenuto il software con miglior rapporto quantità/qualità, intendendo con quantità l'effettivo spazio occupato da ogni singolo video e con qualità la risoluzione e l'obiettiva qualità del video.

#### 4.2.1.2 Software esterni

Per creare *Succeed* ci siamo avvalsi dell'uso di software creato da terze parti, come ad esempio per la creazione dei video, inoltre si sono utilizzati anche script esterni per la gestione del video player e di alcuni aspetti grafici.

Nel dettaglio ci siamo avvalsi delle seguenti tecnologie esterne:

- **Camtasia Studio<sup>96</sup>**

*Screen Recorder*, ovvero software che consente la registrazione di ciò che succede a schermo, sviluppato da TechSmith<sup>97</sup>.

Questo software è stato utilizzato per registrare i video didattici, per editarli ed infine codificarli in formato Flash.

- **Flow Player<sup>98</sup>**

Flash Player creato da Flowplayer Ltd. che consente l'embed di video in formato Flash all'interno di pagine Web, il suo funzionamento e il suo aspetto sono altamente configurabili grazie ad un ottimo sistema di API.

- **Nifty Corners Cube<sup>99</sup>**

Script che consente l'arrotondamento degli angoli degli elementi HTML all'interno di una pagina Web e sviluppato da Alessandro Fulciniti<sup>100</sup>.

L'arrotondamento di parti quali menu, box e bottoni è stato scelto per conferire un aspetto più armonioso all'intero layout. La scelta è ricaduta su questo script in quanto consente di ridurre notevolmente le linee di codice necessarie per la creazione di questo tipo di effetto, conferendo al codice una maggior linearità e semplicità. Per motivi di *crossbrowsing* però non tutti gli "arrotondamenti" sono stati creati mediante l'utilizzo di questo script. Si discuterà di questo più approfonditamente nei prossimi capitoli.

---

<sup>94</sup> <http://www.adobe.com/> - pagina ufficiale di Adobe System

<sup>95</sup> <http://www.adobe.com/products/flash/> - sezione dedicata a Flash sul sito di Adobe.

Nota: si ricorda che per la corretta visualizzazione di materiale Flash sul proprio computer è necessario installare l'apposito Flash Player per Web browser, disponibile sul sito ufficiale Adobe al seguente link: <http://get.adobe.com/it/flashplayer/>

<sup>96</sup> <http://www.techsmith.com/camtasia.asp> - pagina ufficiale del software Camtasia Studio

<sup>97</sup> <http://www.techsmith.com/> - pagina ufficiale della compagnia TechSmith

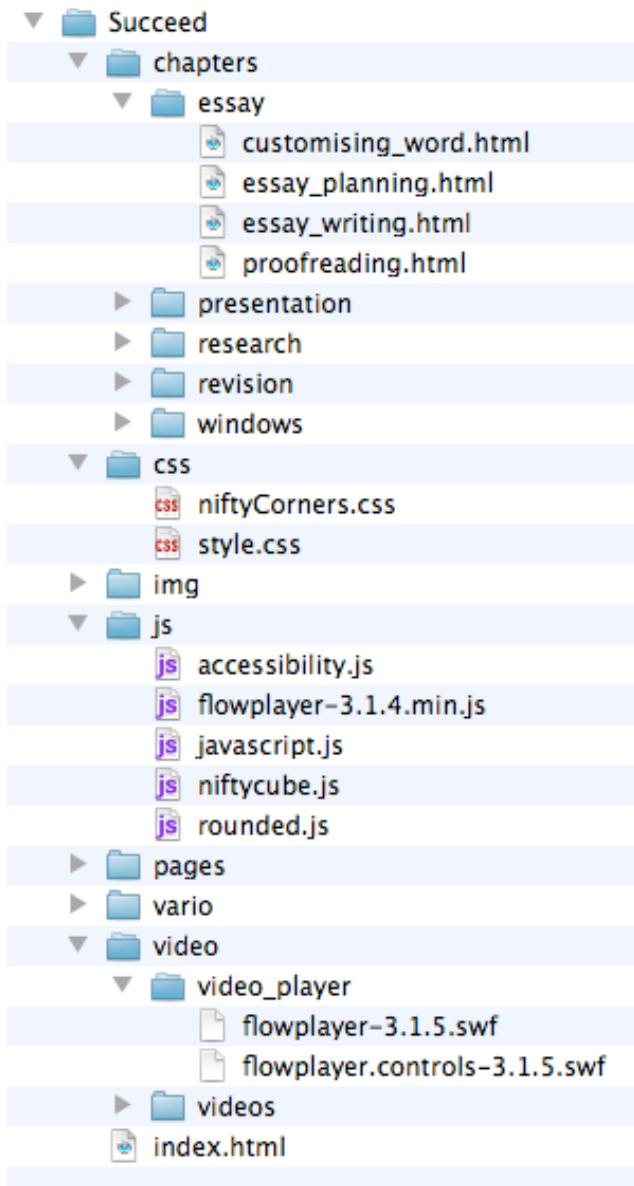
<sup>98</sup> <http://flowplayer.org/> - pagina ufficiale del Flash player flowplayer

<sup>99</sup> <http://www.html.it/articoli/niftycube/index.html> - pagina di riferimento per il software Nifty Corners Cube

<sup>100</sup> [a.fulciniti@html.it](mailto:a.fulciniti@html.it) - indirizzo email di Alessandro Fulciniti, sviluppatore dello script Nifty Corners Cube

## 4.3 Struttura interna del software

Il software *Succeed* ha una struttura interna molto lineare. Come si può notare in figura, dove viene mostrata la struttura gerarchica del programma, vi è una suddivisione logica secondo la tipologia di ogni singolo file. Infatti si possono trovare le seguenti sezioni:



funzionamento dello script *Nifty Corners Cube* necessario all’arrotondamento grafico degli angoli.

### • **img**

La cartella *img* ha il compito di contenere tutte le immagini che si trovano all’interno del software. Si va dalle immagini delle icone a le immagini relative ai campioni di gradienti di colore.

### • **js**

La cartella *js* contiene tutti i file *JavaScript* utilizzati all’interno del programma. Gli script presenti sono i seguenti:

#### - **accessibility.js**

### • **Pagine HTML**

Alle pagine HTML sono dedicate tre sezioni:

#### - **Chapters**

Contiene le cartelle relative ad ogni singola parte dell’area didattica. Ogni capitolo ha una sua cartella, all’interno della quale è presente una pagina per ogni sotto sezione.

#### - **Pages**

Contiene tutte le pagine generiche che non fanno parte di nessuna delle aree tematiche. Qui si trovano le pagine di help e quelle dedicate ad ogni singolo tutorial software.

#### - **Index.html**

Non è una vera e propria sezione, infatti si tratta di una singola pagina HTML che risiede nella cartella principale (*root*) del programma, è stato scelto di posizionarla qua per aumentarne la visibilità e di conseguenza la sua accessibilità.

### • **css**

Questa cartella contiene tutti i fogli di stile (*Cascading Style Sheet*).

#### - **Style.css**

Si tratta del foglio di stile principale, tutte le regole necessarie alla corretta visualizzazione del layout si trovano al suo interno.

#### - **niftyCorners.css**

Contiene le regole di stile necessarie al

Contiene tutte le funzioni che consentono il funzionamento della barra dell'accessibilità (*access bar*).

- ***flowplayer-3.1.4.min.js***

Contiene tutte le funzioni *JavaScript* necessarie al corretto funzionamento del Flash Player *Flowplayer*.

- ***javascript.js***

File che contiene tutti i principali script utilizzati all'interno del programma. Si possono trovare infatti gli script necessari alla visualizzazione e alla scomparsa di elementi della pagina, all'evidenziazione delle voci dei vari menu e alla gestione dei funzionamenti dei *tools*.

- ***niftycube.js***

Contiene tutte le funzioni necessarie al funzionamento dello script *Nifty Corners Cube*.

- ***rounded.js***

Contiene invece l'elenco degli elementi presenti in forma "arrotondata" all'interno delle pagine Web. Inoltre contiene la funzione che richiama, al caricamento di ogni singola pagina, il settaggio dell'aspetto in base all'accessibilità.

• **vario**

Questa cartella ha il fine di contenere tutti quei file superflui al corretto funzionamento del programma ma comunque che possono sempre essere utili. Infatti al suo interno si possono trovare i file della codifica del colore, versioni alternative del codice e backup vari.

• **video**

La cartella video contiene tutto il materiale relativo ai video didattici. Contiene due sottosezioni:

- ***video player***

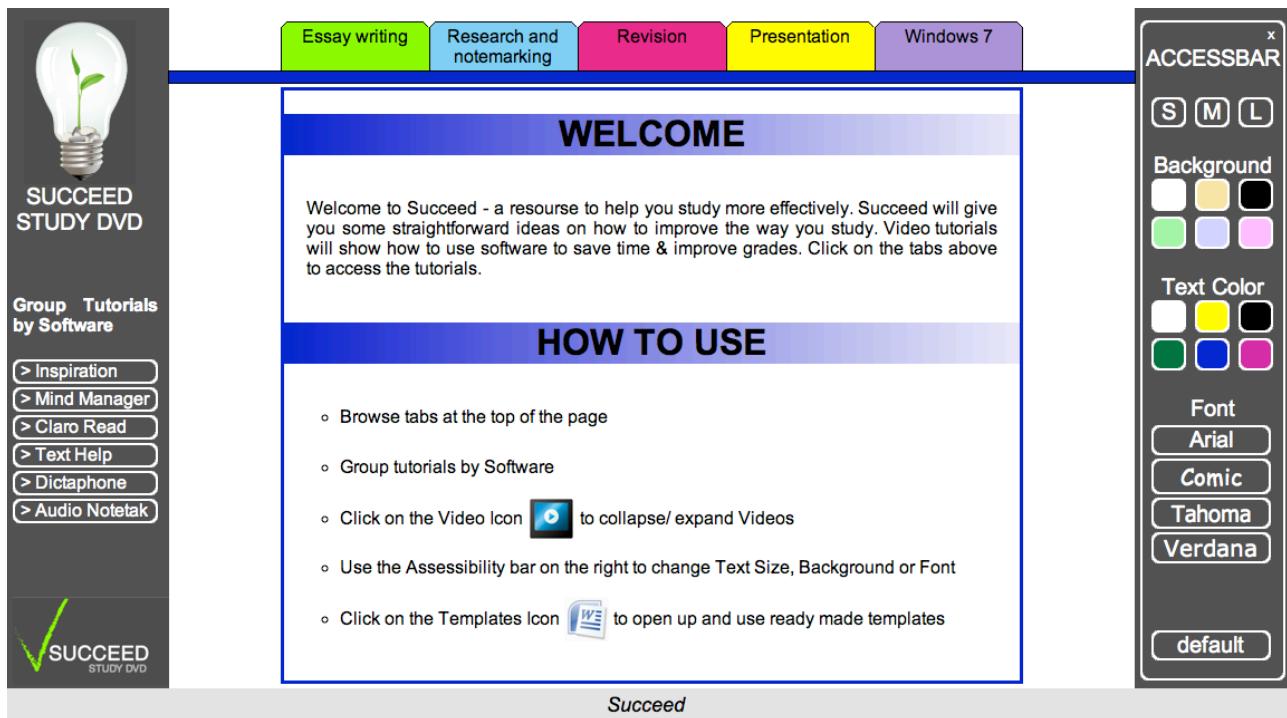
Al suo interno sono contenuti i due file Flash principali che costituiscono il video player vero è proprio. Uno (*flowplayer-3.1.5.swf*) si occupa della visualizzazione e l'altro (*flowplayer.controls-3.1.5.swf*) della gestione dei controlli sul flusso video.

- ***videos***

In questa cartella sono contenuti tutti i video utilizzati all'interno del software *Succeed*.

## 4.4 Il layout

Il layout rappresenta l'impaginazione e la struttura grafica di una pagina Web e di conseguenza anche del software *Succeed*.



L'immagine qui riportata rappresenta l'*Home Page* di *Succeed*, in essa è visibile come il layout appare per intero.

### 4.4.1 La struttura

Il layout adottato dal software *Succeed* è liberamente ispirato al layout *Holy Grail*<sup>101</sup>.

La struttura principale si può riassumere in questi punti:

- *colonna centrale fluida (#center)*, di dimensione variabile, ciò significa che si adatta in modo dinamico alla grandezza della finestra del Web browser. Ha anche una dimensione minima per consentire ai tab menu di apparire sempre per intero.
- *colonna sinistra fissa (#left)*, di dimensione non variabile, ha il compito di contenere i loghi del software e i collegamenti ai tutorial software.
- *colonna destra fissa (#right)*, di dimensione non variabile, ha il compito di contenere il link all'home page, la access bar e altri link di servizio (quali ad esempio le funzioni di download dei contenuti).
- *footer (#footer)*, che conterrà informazioni "burocratiche" quali informazioni legali, copyrights e brevi informazioni sulla compagnia.

<sup>101</sup> <http://www.alistapart.com/articles/holygrail> - pagina di riferimento per il layout Holy Grail

## *Modello strutturale*



In figura è mostrata la struttura generale del layout

*Nota: Nella figura di esempio è rappresentata la struttura del layout in linea di massima, in quanto per motivi di maggior chiarezza è stato preferito scegliere la maggior comprensibilità invece che il rispetto delle proporzioni originali.*

#### 4.4.2 Il codice

##### Struttura HTML

Ogni pagina del software *Succeed* che utilizza il layout a tre colonne precedentemente illustrato deve rispettare la seguente struttura del codice HTML:

```
[...]
<body>

    <div id="container">

        <!--Center Column-->
        <div id="center" class="column"></div>

        <!--Left Column-->
        <div id="left" class="column"></div>

        <!--Right Column-->
        <div id="right" class="column"></div>

    </div>

    <!--Footer-->
    <div id="footer"></div>

</body>
[...]
```

##### CSS - i Selettori<sup>102</sup>

Per comprendere meglio come abbinare le proprietà di stile presenti nei file CSS alle pagine HTML è opportuno capire come vengono applicate queste regole mediante l'uso dei selettori quali *classi* ed *identificatori*.

##### Classi

Le classi applicano la regola a tutti gli elementi della pagina HTML che presentano la proprietà `class="nome_classe"`. La sintassi CSS è la seguente:

```
.nome_classe {
    [...]
}
```

##### Identificatori

Gli identificatori (comunemente ID) applicano la regola a quell'elemento della pagina che presenta la proprietà `id="nome_identificatore"`. Gli ID contraddistinguono elementi unici. La sintassi CSS è la seguente:

```
#nome_identificatore {
    [...]
}
```

---

<sup>102</sup> [http://www.w3schools.com/Css/css\\_id\\_class.asp](http://www.w3schools.com/Css/css_id_class.asp) - per un approfondimento sui selettori CSS

## Il codice CSS

Qui di seguito è riportato tutto il codice CSS necessario per la corretta impostazione del layout descritto nella sezione precedente. Dal seguente codice sono state tolte tutte le parti superflue alla descrizione del layout, quali ad esempio quelle necessarie alla formattazione del testo, per cercare di renderlo il più semplice possibile.

### style.css

Il codice che si occupa dello stile è contenuto nel file CSS.

```
body {  
    min-width: 1140px;  
    margin: 0;  
    padding: 0;  
}  
  
#container {  
    padding-left: 150px;  
    padding-right: 340px;  
    overflow: hidden;  
    background: #414141;  
}  
  
#container .column {  
    position: relative;  
    float: left;  
    padding-top: 1em;  
}  
  
#center {  
    padding: 15px 100px;  
    width: 100%;  
    background: #FFF;  
}  
  
#left {  
    width: 130px;  
    padding: 0 10px;  
    right: 350px;  
    margin-left: -100%;  
}  
  
#right {  
    width: 130px;  
    padding: 0 10px;  
    margin-right: -100%;  
}  
  
#footer {  
    padding: 0.3em 0;  
    background: #DDD;  
}
```

## Il codice CSS nel dettaglio

### Body

```
body {
```

Con body si intendono tutti i contenuti visualizzati del browser.

Praticamente rappresenta il “corpo” vero e proprio della pagina web.

```
min-width: 1140px;
```

**min-width:** rappresenta la larghezza minima che può assumere un dato elemento.

La larghezza minima dell’intera pagina è impostata a 1140 px, ciò significa che riducendo le dimensioni della finestra del browser, una volta scesi al di sotto di questa larghezza, la parte centrale del layout non verrà più ridimensionata e la pagina, nella sua larghezza, non sarà più visualizzata per intero e sarà navigabile mediante la barra di scorrimento orizzontale.

```
margin: 0; padding: 0;
```

**margin<sup>103</sup>** e **padding<sup>104</sup>**: definiscono lo spazio circostante gli elementi. La prima lo spazio esterno ai bordi, la seconda quello interno.

L’impostazione sia del *margin* che del *padding* a 0 è fondamentale per consentire al layout di “occupare” tutto lo spazio disponibile del browser.

```
}
```

### Container - il contenitore

```
#container {
```

Il *conteiner* ha lo scopo di contenere le tre colonne principali (centrale, destra e sinistra). Esso si posiziona al di sopra del *footer* e di fatto conterrà le parti essenziali del programma.

```
padding-left: 150px;
```

La larghezza di *padding sinistro* del *container*, per una corretta visualizzazione del layout in tre colonne, deve corrispondere esattamente alla larghezza della colonna sinistra.

```
padding-right: 340px;
```

La larghezza di *padding destro* del *container* deve invece essere pari alla larghezza della colonna destra più la larghezza di *padding* della colonna centrale.

```
overflow: hidden;
```

**overflow<sup>105</sup>**: specifica cosa succede nel caso un contenuto dovesse sovrapporsi alla struttura di un altro elemento.

In questo caso questo sovrapporsi verrà nascosto, è necessario per un corretto posizionamento tra *container* e *footer*.

---

<sup>103</sup> [http://www.w3schools.com/css/css\\_margin.asp](http://www.w3schools.com/css/css_margin.asp) - approfondimento sul margin

<sup>104</sup> [http://www.w3schools.com/css/css\\_padding.asp](http://www.w3schools.com/css/css_padding.asp) - approfondimento sul padding

<sup>105</sup> [http://www.w3schools.com/Css/tryit.asp?filename=trycss\\_overflow](http://www.w3schools.com/Css/tryit.asp?filename=trycss_overflow) - esempio sull’uso dell’overflow

```
background: #414141;
```

**background**: serve ad impostare il colore di sfondo di un elemento.

Impostando il colore di *background* del *container* si riesce ad “aggirare” il problema di *wrapping* delle due colonne laterali. Con *wrapping* si intende il problema di dare un’altezza omogenea ad entrambe le colonne, colorando lo sfondo dell’elemento nel quale sono contenute non sarà necessario che esse abbiano la stessa dimensione, in quanto la differenza sarà “mimetizzata” dal colore di sfondo del container e quindi non visibile.

```
}
```

## Column - le colonne

```
#container .column {
```

Questa *classe* (.column) di regole serve a contenere tutte le proprietà legate alle tre colonne contenute, e che di conseguenza costituiscono, il *container*.

```
position: relative;
```

**position**<sup>106</sup>: la proprietà *position* (*posizione*) consente di specificare in quale modo un elemento deve posizionarsi all’interno della pagina. In questo caso in modo *relativo*, cioè non dipendente dalla posizione degli altri elementi.

```
float: left;
```

**float**<sup>107</sup>: serve a indicare in quale modo gli elementi devono posizionarsi all’interno della finestra, in questo caso a *sinistra*. La proprietà *float* si può vedere come una sorta di allineamento.

```
padding-top: 1em;
```

*Padding superiore* di 1em, serve per dare un po’ di “respiro” al layout.

```
}
```

## Center - colonna centrale

```
#center {
```

L’identificatore *center* è utilizzato per la colonna centrale del layout, quella che all’interno conterrà i testi principali delle sezioni.

```
padding: 15px 100px;
```

La proprietà *padding*, come nelle classi precedenti, descrive la distanza del testo dal bordo dell’elemento. La distanza scelta in questo caso ha scopi puramente strutturali.

```
width: 100%;
```

La *larghezza* (width appunto) impostata a 100% indica che la larghezza della colonna centrale andrà a sfruttare tutto lo spazio a sua disposizione. Questo significa che una volta che si andrà a ridimensionare il browser questa larghezza cambierà di conseguenza in modo dinamico.

---

<sup>106</sup> [http://www.w3schools.com/Css/css\\_positioning.asp](http://www.w3schools.com/Css/css_positioning.asp) - approfondimento sulla proprietà position

<sup>107</sup> [http://www.w3schools.com/Css/css\\_float.asp](http://www.w3schools.com/Css/css_float.asp) - approfondimento sulla proprietà float

```
background: #FFF;
```

Con questa proprietà si descrive che la colonna centrale del layout utilizza uno sfondo di colore bianco.

```
}
```

### **left, right e footer - colonna destra, sinistra e footer**

Le proprietà contenute nei seguenti identificatori sono già state prese in esame per le altre parti del layout. Di conseguenza l'analisi di questo codice è tralasciata per questioni di ridondanza.

## 4.5 Access Bar - la barra dell'Accessibilità

La barra dell'accessibilità ha lo scopo di rendere più accessibile il software *Succeed* offrendo le seguenti funzionalità:

- Modifica della *dimensione del testo*.
- Modifica del *colore dello sfondo*.
- Modifica del *colore del testo*.
- Modifica del *tipo di font*.
- Ritorno ai parametri di *default*.

Tutte le modifiche apportate dalla Access Bar avranno effetto solamente nella parte centrale del layout, quella che conterrà il testo.

La barra dell'accessibilità basa il suo funzionamento sui *Javascript* contenuti all'interno del file **Accessibility.js**.

All'interno di questo file, oltre alle procedure di impostazione delle varie modifiche, sono contenute anche funzioni specifiche per il salvataggio e la lettura dei *cookie*<sup>108</sup>, che consentono di mantenere le impostazioni selezionate anche cambiando pagina oppure riaprendo il programma, e per il mantenimento di un armonia cromatica, ad esempio se lo sfondo è impostato di colore bianco verrà cambiato automaticamente una volta scelto il colore del testo nero.

### Impostazioni iniziali:

- dimensione del testo: *S* (piccolo)
- colore di sfondo: *bianco*
- colore del testo: *nero*
- tipo di font: *Arial*



### Piccola specificazione preliminare:

Per consentire un ridimensionamento della grandezza del testo esso dovrà essere indicato sotto forma percentuale all'interno del file CSS.

```
#text{ font-size: 100%; }
```

<sup>108</sup> I **cookie HTTP**, chiamati anche Web cookies, sono brevi frammenti di testo che possono essere salvati sul computer quando un utente visita un determinato sito Web. Il server invia i dati che sono salvati in locale e all'occorrenza queste informazioni possono essere rilette quando l'utente torna a visitare tale server, in questo modo un sito Web può adattarsi automaticamente all'utente. I cookie HTTP sono usati per eseguire autenticazioni e tracking di sessioni e memorizzare informazioni specifiche riguardanti gli accessi alla pagina. Nel caso del software Succeed il server è rappresentato dalla stessa macchina sul quale girerà il programma.

#### 4.5.1 Le funzioni JavaScript

Qui di seguito è riportato l'elenco delle funzioni *JavaScript* presenti all'interno del file *accessibility.js* corredate da una breve descrizione dei compiti principali svolti da ognuna.

```
function setFontSize(size) {...}
```

Imposta la dimensione del testo.

```
function setBackgroundColor(color,check) {...}
```

Imposta il colore di sfondo.

```
function setTextColor(color,check) {...}
```

Imposta il colore del testo.

```
function checkColors(id,color){...}
```

Consente di determinare se i colori selezionati sono in conflitto, nel senso che impediscono la lettura del testo, e nel caso lo fossero li cambia. Ad esempio se è selezionato lo sfondo e l'utente seleziona il testo di colore bianco, lo sfondo verrà impostato automaticamente come nero.

```
function setTextStyle(font){...}
```

Imposta il tipo di font.

```
function setDefault(){...}
```

Reimposta le impostazioni iniziali.

```
function createCookie(name,value,days){...}
```

Crea e salva i *cookies* relativi alle varie modifiche apportate dall'utente mediante la Access Bar.

```
function readCookie(name){...}
```

Carica e legge i *cookies* relativi alle modifiche apportate nelle sessioni o nelle pagine precedenti.

```
function setAccessibility(){...}
```

Imposta le caratteristiche precedentemente selezionate e contenute all'interno dei *cookies* al caricamento di ogni pagina.

Per un corretto funzionamento di questa funzione sono necessarie le seguenti righe di codice:

```
window.onload=function() {  
    setAccessibility();  
}
```

Esse servono ad indicare che ogni qualvolta viene caricata una pagina deve essere chiamata la funzione *setAccessibility* che consente di impostare l'aspetto precedentemente selezionato, oppure quello di default.

Nel codice HTML per essere chiamate queste funzioni saranno indicate come dei link.

Nell'esempio è indicato il codice necessario alla chiamata di *setFontSize*.

```
<a href="javascript: setFontSize(1)">
```

## Le funzioni JavaScript nel dettaglio

Ora verrà analizzato il funzionamento delle singole funzioni.

(Nota: non tutte le linee di codice saranno commentate, se delle istruzioni sono già state esaminate per una funzione non saranno più esaminate nelle successive, questo per evitare ridondanze ed un'eccessiva pesantezza esplicativa.)

**setFontSize** - per impostare la grandezza del testo

```
function setFontSize(size) {  
    var testo = document.getElementById('text');  
    var percentage = "100%"; //default S  
    if(size == 1) percentage = "100%"; //S  
    if(size == 2) percentage = "120%"; //M  
    if(size == 3) percentage = "150%"; //L  
    testo.style.fontSize = percentage;  
    createCookie('TextFontSize',size,365);  
}
```

```
var testo = document.getElementById('text');
```

Per prima cosa si imposta l'elemento sul quale verranno apportate le modifiche, in questo caso `text`, che sta ad indicare la parte di pagina Web contenente il testo principale di ogni singola pagina.

```
var percentage = "100%"; //default S
```

Di *default* la dimensione del testo, espressa in percentuale, è pari al 100% e corrisponde ad **S** nell'Access Bar.

```
if(size == 1) percentage = "100%"; //S  
if(size == 2) percentage = "120%"; //M  
if(size == 3) percentage = "150%"; //L
```

A questo punto, in base al valore di `size`, viene impostata la percentuale di grandezza che dovrà assumere ora il testo. Seguendo questi abbinamenti:

| size | percentuale | grandezza |
|------|-------------|-----------|
| 1    | 100%        | S         |
| 2    | 120%        | M         |
| 3    | 150%        | L         |

```
testo.style.fontSize = percentage;
```

Questa linea di codice consente di impostare la grandezza del testo selezionata.

Praticamente viene assegnato il valore scelto al campo `fontSize` dell'elemento `text`.

```
createCookie('TextFontSize',size,365);
```

Il varole appena impostato viene salvato in un *cookie*.

In seguito verrà illustrato più nel dettaglio come vengono gestiti i *cookie*.

**setBackgroundColor** - per impostare il colore dello sfondo

```
function setBackgroundColor(color, check) {
    var testo = document.getElementById('text');
    var backgroundColor = "#FFF"; //default white
    if (color == 1) backgroundColor = "#FFF"; // white
    if (color == 2) backgroundColor = "#F6E596"; //light yellow
    if (color == 3) backgroundColor = "#000"; //black
    if (color == 4) backgroundColor = "#96F698"; //light green
    if (color == 5) backgroundColor = "#CCCCFF"; //light blue
    if (color == 6) backgroundColor = "#FBAEFF"; //light pink
    if (check) {
        checkColors('b', color);
    }
    testo.style.backgroundColor = backgroundColor;
    createCookie('BackgroundColor', color, 365);
    return;
}
```

Il funzionamento della funzione `setBackgroundColor` segue lo stesso principio utilizzato da `setFontSize`, viene identificata l'area sulla quale si vuole apportare la modifica, viene identificato il colore, viene impostato il colore scelto ed infine viene creato il *cookie* corrispondente.

La selezione del colore segue i seguenti abbinamenti:

| color | codifica esadecimale | colore corrispondente |
|-------|----------------------|-----------------------|
| 1     | #FFF                 | bianco                |
| 2     | #F6E596              | giallo chiaro         |
| 3     | #000                 | nero                  |
| 4     | #96F698              | verde chiaro          |
| 5     | #CCCCFF              | blu chiaro            |
| 6     | #FBAEFF              | rosa chiaro           |

```
if (check) {
    checkColors('b', color);
}
```

Queste linee di comando indicano che in questo punto viene richiesto il controllo del colore, per evitare problemi di leggibilità. Per capire meglio questo particolare si rimanda alla sezione dedicata alla funzione `checkColors`, per il momento è importante capire che tale funzione è richiamata mandandole il colore selezionato e un selettore, in questo caso *b* (che sta ad indicare che si tratta del colore di sfondo e quindi di *background*).

A differenza delle altre funzioni, nelle quali è presente un solo parametro di *input*, che serve appunto ad indicare le proprietà del parametro di modifica, qui compare una *variabile* chiamata *check* e di conseguenza un controllo su di essa. Questa *variabile* ha lo scopo di indicare se il controllo sul colore, rappresentato dalla funzione *checkColors*, è già stato eseguito. All'interno della variabile *check* sarà possibile trovare due valori *true* e *false*, *true* significa che il controllo deve essere eseguito, chiamando la funzione *checkColors*, mentre se il valore è impostato come *false* il controllo verrà saltato.

**setTextColor** - per impostare il colore del testo

```
function setColor(color,check) {
    var testo = document.getElementById('text');
    var textColor = "#000"; // default black
    if (color == 1) textColor = "#FFF"; //white
    if (color == 2) textColor = "#FF0"; //yellow
    if (color == 3) textColor = "#000"; //black
    if (color == 4) textColor = "#006633"; //green
    if (color == 5) textColor = "#0000CC"; //blue
    if (color == 6) textColor = "#CC0099"; //pink
    if (check) {
        checkColors('t',color);
    }
    testo.style.color = textColor;
    createCookie('TextColor',color,365);
}
```

Il funzionamento di *setBackgroundColor* è analogo a *setBackgroundColor*, in quanto segue il medesimo procedimento, con le sole varianti nella selezione dei colori e nel richiamo della funzione di controllo, dove questa volta verrà passata la lettera *t*.

L'assegnazione dei colori segue questo schema:

| color | codifica esadecimale | colore corrispondente |
|-------|----------------------|-----------------------|
| 1     | #FFF                 | bianco                |
| 2     | #FF0                 | giallo                |
| 3     | #000                 | nero                  |
| 4     | #006633              | verde                 |
| 5     | #0000CC              | blu                   |
| 6     | #CC0099              | rosa                  |

### **checkColors - il controllo sui colori**

```
function checkColors(id,color) {
    var testo = document.getElementById('text');

    //background-color
    if(id=='b'){
        switch(readCookie('TextColor')){
            case '1': //white
                if(color!=3)
                    setTextColor(3,false);
                break;
            case '2': //yellow
                if(color!=3)
                    setTextColor(3,false);
                break;
            case '3': //black
                if(color==3)
                    setTextColor(1,false);
                break;
        }
    }

    //text-color
    if(id=='t'){
        switch(color){
            case 1: //white
                if(readCookie('BackgroundColor') !=3)
                    setBackgroundColor(3,false);
                break;
            case 2: //yellow
                if(readCookie('BackgroundColor') !=3)
                    setBackgroundColor(3,false);
                break;
            case 3: //black
                if(readCookie('BackgroundColor') ==3)
                    setBackgroundColor(1,false);
                break;
        }
    }

    return;
}
```

La funzione `checkColors` principalmente è divisa in due parti, una relativa al controllo del colore dello sfondo e l'altra al colore del testo.  
Controlla che i colori scelti siano abbinabili, in caso contrario attiva in automatico il cambio di uno dei due.

### *Controllo del colore del background*

```
if(id=='b') {
```

Se l'identificatore (`id`) dell'elemento è `b` significa che il colore da controllare è relativo allo sfondo (*background*).

```
switch(readCookie('TextColor')) {
```

Viene letto il *cookie* relativo al colore del testo, per confrontarlo con quello dello sfondo. Si utilizza un'istruzione di `switch` che consente una scelta multipla, in questo caso in base al colore letto verranno eseguite differenti righe di codice. I `case`, cioè i differenti valori che possono essere assunti da `TextColor`, fanno riferimento agli abbinamenti di colore visti nella funzione `setTextColor`.

```
case '1': //white
```

Caso 1, significa che il testo del documento è impostato di colore bianco.

```
if(color!=3)
```

Se si è scelto un colore di sfondo differente da nero.

```
setTextColor(3, false);
```

Viene automaticamente impostato il colore del testo come nero, ricordando di passare `false` come condizione di controllo, in quanto è appena stato eseguito.

```
break;
```

Questa istruzione serve ad indicare che la funzione dello `switch` è terminata e quindi di "passare" al codice che si trova successivamente.

Gli altri casi utilizzano la stessa logica e sono omessi per evitare ridondanze di codice.

### *Controllo del colore del testo*

```
if(id=='t') {
```

Come già visto per lo sfondo `id` indica di quale elemento si sta "passando" la colorazione. In questo caso dato che è `t` si tratta del colore del *testo*.

```
switch(color) {
```

Questa volta lo `switch` è fatto sul colore che dovrà assumere lo sfondo.

```
case 1: //white
```

Caso 1, il colore selezionato per il testo è bianco.

```
if(readCookie('BackgroundColor')!=3)
```

Di conseguenza se lo sfondo è diverso da nero.

```
setBackgroundColor(3, false);
```

Imposta lo sfondo di colore nero.

In entrambi i casi lo `switch` lavora sul colore del testo, questa scelta è stata fatta sia per limitare il numero di linee di codice sia per mantenere una certa semplicità dello script, in quanto, analizzando il colore del testo, si deve affrontare un numero minore di casi.

#### *Gli abbinamenti possibili*

| testo  | condizione       | sfondo |
|--------|------------------|--------|
| bianco | solo con         | nero   |
| giallo | solo con         | nero   |
| nero   | tutti tranne con | nero   |

**setTextStyle** - per impostare il font (lo stile) del testo

```
function setTextStyle(font) {
    var testo = document.getElementById('text');
    var family = "arial"; // default
    if (font == 1) family = "arial";
    if (font == 2) family = "Comic Sans MS";
    if (font == 3) family = "tahoma";
    if (font == 4) family = "verdana";
    testo.style.fontFamily = family;
    createCookie('TextFont', font, 365);
}
```

Funzione analoga a `setFontSize`. Il funzionamento è identico con la sola variante che questa volta si occupa di modificare lo stile del testo.

```
if (font == 1) family = "arial";
if (font == 2) family = "Comic Sans MS";
if (font == 3) family = "tahoma";
if (font == 4) family = "verdana";
```

Le famiglie selezionabili di font sono quattro e rispettano i seguenti assegnamenti:

| font | font-family   | esempio |
|------|---------------|---------|
| 1    | Arial         | testo   |
| 2    | Comic Sans MS | testo   |
| 3    | Tahoma        | testo   |
| 4    | Verdana       | testo   |

**setDefault** - imposta l'aspetto di *default*

```
function setDefault() {
    var testo = document.getElementById('text');
    testo.style.color = "#000";
    testo.style.backgroundColor = "#FFF";
    testo.style.fontSize = "100%";
    testo.style.fontFamily = "arial";
    createCookie('TextFont','1',365);
    createCookie('TextColor','3',365);
    createCookie('BackgroundColor','1',365);
    createCookie('TextFontSize','1',365);
}
```

In questa funzione vengono semplicemente assegnati i valori di *default* alle varie proprietà dell'elemento e il tutto viene memorizzato in dei *cookies*.

**setAccessibility** - imposta l'aspetto al caricamento della pagina

```
function setAccessibility() {
    size = readCookie('TextFontSize');
    setFontSize(size);
    font = readCookie('TextFont');
    setTextStyle(font);
    color = readCookie('TextColor');
    setTextColor(color,true);
    backgroundColor = readCookie('BackgroundColor');
    setBackgroundColor(backgroundColor,true);
    accessState = readCookie('AccessState');
    if(accessState==0)
        accessCollapse(true);
    else
        accessCollapse(false);
}
```

size = readCookie('TextFontSize');

Legge dal *cookie* le informazioni relative alla grandezza del testo e le immagazzina in una *variabile*.

setFontSize(size);

Richiama la funzione di impostazione della grandezza del testo con il valore appena letto e contenuto nella *variabile size*.

```
if(accessState==0)
    accessCollapse(true);
else
    accessCollapse(false);
```

Queste righe di codice si occupano di capire se l'Access Bar è visibile o “contratta” e di impostarla di conseguenza. Questa funzione verrà successivamente trattata con maggior dettaglio.

## I Cookies

Come già spiegato in precedenza i *cookies* sono uno strumento indispensabile per memorizzare informazioni relative allo stato di una pagina Web per le future visite. Anche il software *Succeed* si avvale di questa tecnologia per “memorizzare” l’aspetto del layout scelto dall’utente.

Per l’utilizzo dei *cookies* sono state create due funzioni apposite, una per crearli `createCookie` e un’altra per leggerli `readCookie`, e sono entrambe contenute nel file `accessibility.js`.

### `createCookie` - crea un *cookie*

```
function createCookie(name,value,days) {  
    if (days) {  
        var date = new Date();  
        date.setTime(date.getTime()+(days*24*60*60*1000));  
        var expires = ' '; expires=''+date.toGMTString();  
    }  
    else expires = '';  
    document.cookie = name+'='+value+expires+'; path=/';  
}
```

Molto brevemente questa funzione si occupa di stabilire una data di durata del *cookie* e successivamente di crearlo.

```
function createCookie(name,value,days) {
```

La funzione riceve in *input* tre parametri, `name` relativo al nome del *cookie* da creare, `value` relativo al valore che dovrà avere e `days` che indica per quanti giorni dovrà rimanere in memoria (in questo *Succeed* si è deciso di salvarlo per un anno intero, cioè 365 giorni).

```
    if (days) {  
        var date = new Date();  
        date.setTime(date.getTime()+(days*24*60*60*1000));  
        var expires = ' '; expires=''+date.toGMTString();  
    }  
    else expires = '';
```

Queste linee di codice si preoccupano solamente di stabilire la data di “scadenza” del *cookie* e non sono importanti per capire il funzionamento del software, per questo la loro spiegazione viene tralasciata.

```
    document.cookie = name+'='+value+expires+'; path=/';
```

Qui viene indicato come sarà costituito il *cookie*, secondo la sintassi (nome = valore + scadenza) e dove verrà salvato.

### **readCookie** - legge un *cookie*

```
function readCookie(name) {  
    var nameEQ = name + '=';  
    var ca = document.cookie.split(';');  
    for(var i=0;i < ca.length;i++) {  
        var c = ca[i];  
        while (c.charAt(0)==' ')  
            c = c.substring(1,c.length);  
        if (c.indexOf(nameEQ) == 0)  
            return c.substring(nameEQ.length,c.length);  
    }  
    return null;  
}
```

La funzione permette di leggere i *cookies* salvati in memoria dal Web browser. Senza addentrarci nei particolari tecnici è importante capire che in questa funzione il *cookie* viene letto ed interpretato. Se esiste ed è valido allora la funzione restituirà il suo valore, altrimenti `null` per indicare che non è presente.

Qui di seguito è riportato un esempio per illustrare come i *cookies*, relativi all'Access Bar, sono salvati e visibili all'interno del Web browser (in questo caso Safari).

| Sito web       | Nome            | Percorso | Sicuro | Scadenza     | Contenuto |
|----------------|-----------------|----------|--------|--------------|-----------|
| .^filecookies^ | AccessState     | /        |        | 9 March 2011 | 1         |
| .^filecookies^ | BackgroundColor | /        |        | 9 March 2011 | 1         |
| .^filecookies^ | TextFont        | /        |        | 9 March 2011 | 1         |
| .^filecookies^ | TextFontSize    | /        |        | 9 March 2011 | 1         |
| .^filecookies^ | TextColor       | /        |        | 9 March 2011 | 3         |

In questo caso, dall'interpretazione dei *cookies*, si può dedurre che l'aspetto salvato è quello di *default*.

## 4.5.2 Esempi

Qui di seguito sono riportati degli esempi per mostrare meglio come opera l'Access Bar.

### Default

Così è come si presenta normalmente il software *Succeed*.

The screenshot shows the main interface of the SUCCEED STUDY DVD software. On the left is a vertical sidebar with a lightbulb icon and the text "SUCCEED STUDY DVD". Below this are links for "Group Tutorials by Software" and a list of tools: Inspiration, Mind Manager, Claro Read, Text Help, Dictaphone, and Audio Notetak. At the bottom of the sidebar is a logo with a checkmark and the text "SUCCEED STUDY DVD". The main content area has a green header bar with tabs: "Essay writing" (selected), "Research and notemarking", "Revision", "Presentation", and "Windows 7". To the right of the tabs is a "Homepage" button. The main content area displays a slide titled "Essay Planning" with the sub-section "Understand the Question". The text on the slide reads: "Figuring out what the question is asking you to do and what your tutors want to see in your finished work is fundamental to writing a good essay." Below this is a video player icon with the text "1. Word - highlighting". A "STUDY TIP" box contains the instruction: "When working on your essay always refer back to the essay question." It also notes that one should re-read the question at all stages of writing. The text continues: "Once you have broken the question down, you are well placed to start reading for your essay." A hyperlink "See the Tab on Reading & Note making for tips on reading." is provided. The right side of the interface features the "ACCESSBAR" panel, which includes buttons for "S" (Small), "M" (Medium), and "L" (Large) text sizes, and color swatches for "Background" (white, yellow, black, green, blue, pink) and "Text Color" (white, yellow, black, green, blue, pink). It also lists fonts: Arial, Comic, Tahoma, Verdana, and a "default" option.

### Cambio della dimensione e del font

In questo esempio è stata impostata la dimensione del testo massima (*L*) e il font *Verdana*.

This screenshot shows the same software interface as above, but with the "L" (Large) size setting selected in the ACCESSBAR. This has enlarged the text throughout the application. The main content area now displays the "Essay Planning" slide with the "Understand the Question" section. The text is significantly larger than in the default view. The "STUDY TIP" box is also larger. The right side of the interface remains the same, showing the ACCESSBAR with its various settings.

## Cambio dello sfondo

Viene selezionato il colore dello sfondo (*background*) giallo.

The screenshot shows the 'Essay Planning' section of the software. The background is yellow. At the top, there is a navigation bar with tabs: 'Essay writing' (highlighted in blue), 'Research and notemarking', 'Revision', 'Presentation', and 'Windows 7'. On the right side, there is a vertical 'ACCESSBAR' panel with buttons for 'Homepage', 'Background' (with color swatches), 'Text Color' (with color swatches), 'Font' (with options: Arial, Comic, Tahoma, Verdana), and a 'default' button. On the left, there is a sidebar with a lightbulb icon and the text 'SUCCEED STUDY DVD'. Below the sidebar is a list of 'Group Tutorials by Software' including: > Inspiration, > Mind Manager, > Claro Read, > Text Help, > Dictaphone, and > Audio Notetak. At the bottom of the sidebar is the 'SUCCEED STUDY DVD' logo.

## Cambio del colore del carattere

### con annesso cambiamento automatico dello sfondo

Se dallo stato presentato nell'esempio precedente, cioè *sfondo giallo*, si decide di impostare il colore del *testo giallo*, lo *script* eseguirà automaticamente il cambio dello sfondo impostandolo di *colore nero*. Come mostrato qui sotto.

The screenshot shows the same 'Essay Planning' section as the previous one, but with a black background and yellow text. The text 'Figuring out what the question is asking you to do and what your tutors want to see in your finished work is fundamental to writing a good essay.' is displayed in yellow. The rest of the interface remains the same, including the navigation bar, ACCESSBAR, sidebar, and footer.

### 4.5.3 Funzioni aggiuntive

Le funzioni descritte qui di seguito sono opzionali per il corretto funzionamento dell'Access Bar.

#### Evidenziazione al passaggio del mouse

Questa funzione ha lo scopo di rendere più chiaro il puntamento delle opzioni disponibili. Il suo funzionamento è molto semplice, quando il puntatore del mouse si trova sopra un area selezionabile, che può essere ad esempio un colore assegnabile al testo, quest'area viene evidenziata colorando i bordi laterali di colore nero (come si può notare nell'immagine qui a lato).



Questa operazione è eseguita da due semplicissime funzioni

*JavaScript* contenute nel file *javascript.js*. Le funzioni sono *bordlight* che consente di colorare il bordo di *nero* e *noBordlight* che invece lo *colora di bianco*.

```
function bordlight(id) {
    document.getElementById(id).style.background='black';
}
```

Questa funzione assegna allo sfondo dell'elemento indicato da *id* il colore nero. Viene chiamata quando il mouse si trova sopra all'elemento selezionato.

```
function noBordlight(id) {
    document.getElementById(id).style.background='white';
}
```

Questa funzione invece assegna allo sfondo dell'elemento indicato da *id* il colore nero e anche lei viene richiamata al passaggio del mouse.

Nella pagina HTML l'elemento che utilizza questa funzione dovrà essere indicato nel seguente modo:

```
<div id="yellowT" class="access_label"
      onmouseover="bordlight(this.id)"
      onmouseout="noBordlight(this.id)">
    <div class="access_color yellow">
      &nbsp;
    </div>
</div>
```

Il frammento di codice qui riportato riguarda parte del "bottone" per impostare il giallo come colore del testo. Si possono due elementi di tipo *div* uno con *id yellowT* avrà sfondo *bianco* e conterrà l'altro di sfondo *giallo*.

Al passaggio del mouse su questo "bottone" l'attributo *onmouseover* chiamerà la funzione *bordlight(this.id)* passandogli come parametro l'*id* dell'elemento da evidenziare, colorandone il bordo laterale di colore *nero*, in questo caso *yellowT*. Una volta che il mouse lascerà quest'area verrà invece chiamata in modo analogo la funzione

`noBordlight(this.id)` mediante l'attributo `onmouseout` con il compito di far tornare il bordo del “bottone” di colore *bianco*.

### Riduzione dell'Access Bar (*collapse*)

La funzione di *collapse* dell'Access Bar ha un compito puramente estetico e consente di ridurre le dimensioni della barra nascondendo le varie opzioni disponibili. Qui a fianco è possibile vedere l'aspetto “ristretto” della barra. Mediante i pulsanti (+) e (x) nell'angolo superiore sarà possibile controllare questa funzione.



La funzione che si occupa di questa operazione di chiama `accessCollapse` ed è presente all'interno del file *javascript.js*. Essa sostanzialmente ha il compito di *mostrare* o *nascondere* l'Access Bar in versione contratta e non. Di conseguenza se l'Access Bar *contratta* sarà *visibile* quella *estesa* sarà *nascosta* e viceversa. Prima di analizzare il codice *JavaScript* è opportuno vedere come si presenta la *Access Bar* all'interno del file *HTML*.

```
//Access Bar contratta
<div id="accessHide">
    <a href="javascript: accessCollapse(false);">
        <div class="accessIcon">+</div>
    </a>
    ACCESSBAR<br/><br/>
</div>
//Access Bar estesa
<div id="accessShow">
    <a href="javascript: accessCollapse(true);">
        <div class="accessIcon">x</div>
    </a>
    ACCESSBAR<br/><br/>

    <!-- codice contenute all'interno dell'Access bar estesa -->
    <!-- omesso per questioni di spazio -->
    <!-- non ha effetti sul funzionamento dello script -->

</div>
```

Nel codice sono presenti contemporaneamente entrambe le *Access Bar*. *Di default* entrambe le *Access Bar* sono impostate *hide* (nascoste) all'interno del CSS, con il seguente codice:

```
#accessHide{display:none; (...)}109
#accessShow{display:none; (...)}
```

Sarà la funzione `setAccessibility`, mediante `accessCollapse` ad occuparsi di quale delle due versioni mostrare al caricamento della pagina, di *default* sarà la versione *estesa* ad essere mostrata.

---

<sup>109</sup> (...) indica che è stato omesso del codice, in quanto irrilevante ai fini dell'esempio.

Durante la normale esecuzione la funzione della funzione di *collapse* sarà invece gestita dai pulsanti (x) e (+).

```
<a href="javascript: accessCollapse(false);">
    <div class="accessIcon">+</div>
</a>
```

Il pulsante + sarà visibile quando l'*Access Bar* è *contratta*. Premento questo pulsante verrà chiamata la funzione `accessCollapse` passandole il parametro `false`.

```
<a href="javascript: accessCollapse(true);">
    <div class="accessIcon">x</div>
</a>
```

Mentre il pulsante x sarà visibile quando l'*Access Bar* è *estesa* e chiamerà la funzione `accessCollapse` passandole questa volta il parametro `true`.

```
function accessCollapse(hide) {
    if(hide) {
        document.getElementById('accessShow').style.display =
            "none";
        document.getElementById('accessHide').style.display =
            "inline-block";
        createCookie('AccessState', 0, 365);
    }
    else{
        document.getElementById('accessShow').style.display =
            "inline-block";
        document.getElementById('accessHide').style.display =
            "none";
        createCookie('AccessState', 1, 365);
    }
    return;
}
```

La funzione `accessCollapse` ha un funzionamento molto semplice, essa infatti in base al parametro `hide` ricevuto in *input*, mostra o nasconde una barra invece che l'altra e salva il tutto in un *cookie*.

```
if(hide) {
```

Se il parametro passato in *input* è uguale a `true` significa che l'*Access Bar* dovrà essere contratta.

```
document.getElementById('accessShow').style.display = "none";
Di conseguenza la parte relativa all'Access Bar estesa dovrà essere impostata come non
visible.
```

```
document.getElementById('accessHide').style.display =  
    "inline-block";
```

Mentre la parte che riguarda l'*Access Bar contratta* dovrà essere resa visibile.

```
createCookie('AccessState', 0, 365);
```

Il tutto poi dovrà essere salvato all'interno di un *cookie* di nome `AccessState`, impostato con valore 0.

```
}
```

Per quanto riguarda la parte relativa alla visualizzazione dell'*Access Bar estesa*, essa opera in modo analogo alla precedente, di conseguenza non verrà riportata.

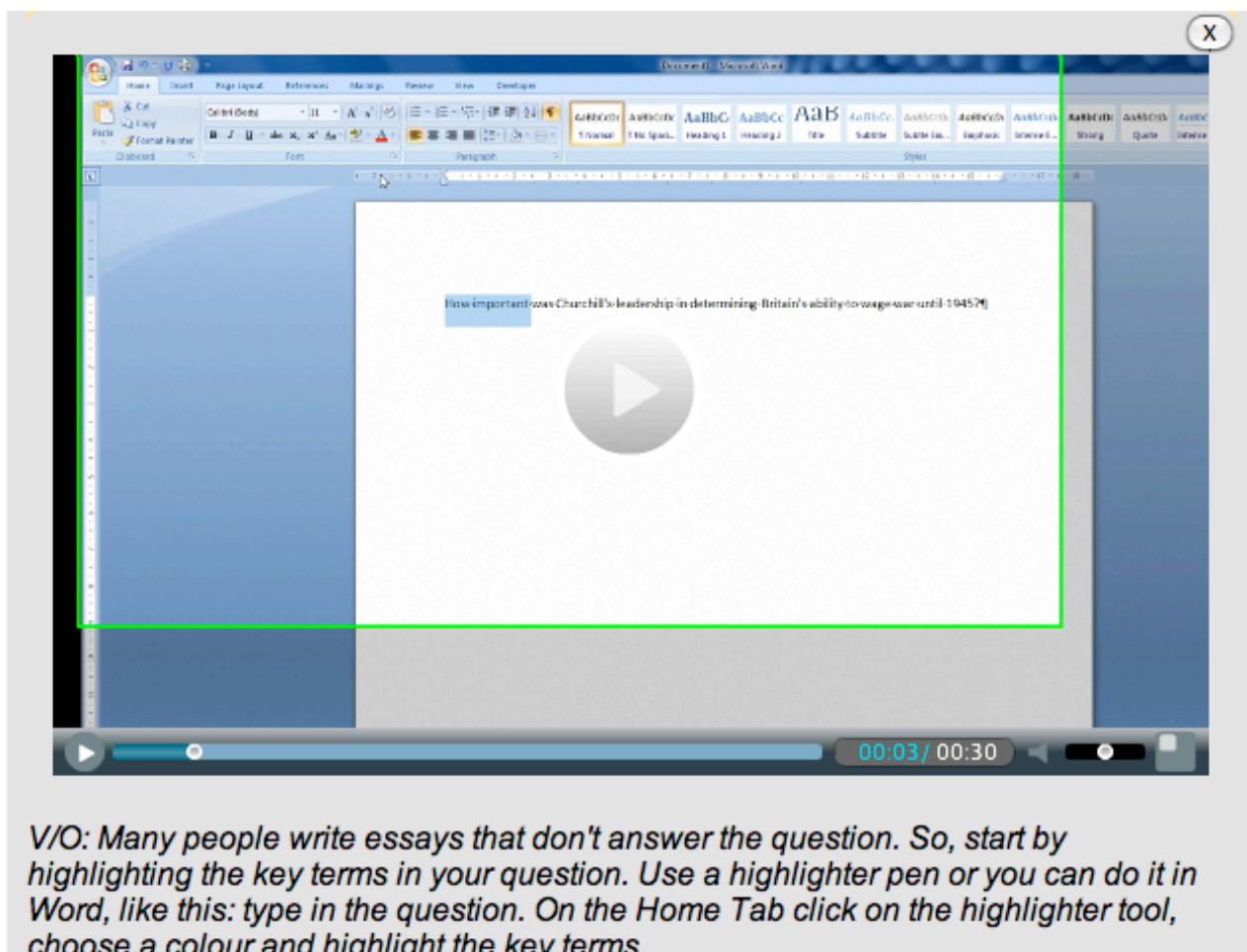
## 4.6 Embed video

Con il termine *embed*, letteralmente integrare, si intende la tecnica di inserire materiale multimediale all'interno di pagine HTML, in questo caso i contenuti saranno di tipo video.

All'interno di *Succeed* i contenuti video sono indicati nel modo seguente:



Per passare alla visualizzazione dei video sarà sufficiente fare *click* sull'icona apparirà una sezione dedicata alla visualizzazione del video:



How important was Churchill's leadership in determining Britain's ability to wage war until 1945?

V/O: Many people write essays that don't answer the question. So, start by highlighting the key terms in your question. Use a highlighter pen or you can do it in Word, like this: type in the question. On the Home Tab click on the highlighter tool, choose a colour and highlight the key terms.

In questa sezione oltre al *player* è presente la trascrizione dell'audio di commento del video, per consentire una maggior accessibilità da parte degli utenti con problemi di udito.

Una volta visionato il video sarà possibile richiudere questa sezione facendo nuovamente click sull'icona oppure sull'apposito pulsante di chiusura.



## Codice per l'embed

```
<!--video_embed-->
<a id="player1" class="video_embed" href="video/video1.flv"></a>
  <script type="text/javascript">
    flowplayer("player1",
      "flowplayer-3.1.5.swf",
      {clip: {autoPlay: false}});
  </script>
<!--video_embed-->
```

Il codice qui riportato è necessario per l'*embed* del video.

<a id="player1" class="video\_embed" href="video/video1.flv"></a>  
Questa linea di codice descrive un *link*, infatti *flowplayer* posiziona il video all'interno del *tag* `<a>`. *id* univoco (nell'esempio `player1`) servirà a consentire questo "collegamento". La *class* è utilizzata per lo stile del video player, specificato all'interno del file *style.css*. Mentre *href* indica la posizione esatta in cui si trova il video da riprodurre.

```
<script type="text/javascript">
flowplayer("player1",
  "flowplayer-3.1.5.swf",
  {clip: {autoPlay: false}});
</script>
```

Questa breve funzione *JavaScript* serve a richiamare il *player* e metterlo in collegamento con il *tag* `<a>`, atto appunto a contenere il video vero e proprio.

Vengono dati come parametri il nome dell'identificatore dell'elemento `<a>`, il video player e dei parametri aggiuntivi, in questo caso `{clip: {autoPlay: false}}` sta ad indicare che la funzione di *autoplay* del video player sarà disabilitata.

## Specifiche tecniche video

La registrazione e la codifica video è stata eseguita mediante il programma *Camtasia*.

### *Il formato: FLV*

Per consentire l'*embeded* si è dovuto optare per una conversione in formato flash FLV, in quanto il player video *flowplayer* può eseguire solo file in formato flash.

### *La risoluzione: 1280 X 720*

Dato che il *player* consente una visualizzazione *full screen* (a tutto schermo) si è dovuta scegliere un'alta risoluzione video, in questo modo anche su schermi di grandi dimensioni il video non verrà sgranato e soprattutto si avrà una maggior nitidezza dei caratteri.

### *La dimensione:*

Una risoluzione così elevata porta ad un ingrandimento dello spazio necessario ad ogni singolo video. Siccome però il software *Succeed* sarà eseguito o in locale o in supporti di formato DVD non si incorrerà né in problemi legati allo spazio né a problemi riguardanti la banda della connessione.

## 4.7 Rounded Corner - Arrotondamento degli angoli<sup>110</sup>

L'arrotondamento degli angoli non rappresenta un aspetto fondamentale per il funzionamento del software *Succeed*, è utilizzato a puro scopo grafico, per rendere l'intero layout maggiormente *look & feel*.

Per la creazione del programma sono stati utilizzati due metodi, uno (*Nifty Corners Cube*) basato su dei file *JavaScript* e l'altro creato utilizzando solo codice *HTML* e *CSS*.

Si sono dovuti utilizzare due metodi in quanto *Nifty Corners Cube* non permette l'utilizzo di bordi, per assicurare una maggior compatibilità anche con *Web browser* datati.

Il problema dei bordi è parzialmente risolvibile utilizzando degli elementi annidati, anche se questo non porta a risultati dal punto di vista grafico del tutto soddisfacenti, per questo, in alcuni casi, si è dovuto ricorrere all'utilizzo del metodo *HTML/CSS*.

### 4.5.1 *Nifty Corners Cube*

*Nifty Corners Cube* consente di creare automaticamente l'arrotondamento degli angoli per componenti grafici *HTML*. Il suo funzionamento si basa su tre file:

- *nifycube.js*  
Contiene tutto il codice *JavaScript* necessario al corretto funzionamento dello *script*.
- *rounded.js*  
Contiene la lista di quali elementi utilizzeranno la funzione di arrotondamento.
- *niftyCorners.css*  
Contiene le regole *CSS* necessarie per l'impostazione dello stile arrotondato.

La comodità di questo *script* risiede nella possibilità di “trasformare” gli angoli di qualsiasi elemento senza andare a modificare la struttura della pagina e tanto meno senza cambiare nulla nella pagina stessa.

Per impostare un elemento di tipo “arrotondato” è sufficiente inserire il suo descrittore all'interno del file *rounded.js*, il codice presente in questo file dovrà essere eseguito ogni volta che la pagina sarà caricata mediante la funzione `window.onload=function()`.

La sintassi da mantenere all'interno del fine *rounded.js* è la seguente:

```
Nifty("div#box","big");
```

Dove *Nifty* indica la funzione da richiamare, "div#box" indica il tipo dell'elemento (in questo caso un `div` con classe `box`) e poi il *tipo di arrotondamento*<sup>111</sup> (in questo caso "big").

All'interno dei *Succeed* si è utilizzato solamente l'arrotondamento di *default*, come ad esempio: `Nifty("div.rounded")`.

---

<sup>110</sup> Ho deciso di prendere in esame questa parte opzionale del software in quanto l'arrotondamento degli angoli necessita l'aggiunta di diversi file all'interno della struttura base di *Succeed*, in questo modo verranno presi in esame tutti i componenti maggiormente significativi. Inoltre, dato che nell'Access Bar viene utilizzata questa tecnica, mi è sembrato opportuno trattarla maggiormente nel dettaglio.

<sup>111</sup> Vi sono numerosi tipi di parametro assegnabili a ciascun elemento, per maggiori dettagli si rimanda alla pagina ufficiale dello script <http://www.html.it/articoli/nifycube/index.html>.

Esempio di arrotondamento con *Nifty Corners Cube*:

### STUDY TIP

Esempio di arrotondamento con bordo, utilizzando la tecnica di “annidamento” degli elementi:



Per capire meglio in cosa consiste questa tecnica si analizzerà nel dettaglio l'esempio presente qui sopra.

Di seguito è riportato il codice relativo:

*In HTML*

```
<div id="default" class="access_font_border">
    <div class="access_font">
        default
    </div>
</div>
```

Vengono creati due elementi di tipo `div` uno contenuto nell'altro a cui vengono assegnati due identificatori di tipo `class`, `access_font_border` per l'elemento *contenitore* che simulerà il bordo e `access_font` per quello *interno* che conterrà il testo vero e proprio dell'elemento.

*In CSS*

```
.access_font_border {
    background-color:#FFF;
    //altro codice non importante ai fini dell'esempio
}
```

Stile per l'elemento *contenitore*, è sufficiente indicare il colore che dovrà assumere il bordo.

```
.access_font {
    margin:2px;
    background:#414141;
    position:relative;
}
```

Stile per l'elemento *interno*, si imposta il `margin`, `2px` in questo caso, relativo all'ampiezza del bordo<sup>112</sup>, il colore dello sfondo (`background`), differente da quello del

---

<sup>112</sup> Per questa tecnica di bordo mediante “annidamento” l'ampiezza minima assumibile dal bordo è di `2px`. Diminuendo questo valore è possibile che si presentino problemi relativi ad una buona resa grafica degli angoli.

*contenitore* ed infine gli si assegna un posizionamento (*position*) relativo (*relative*) all'interno dell'elemento *contenitore*.

In rounded.js

```
Nifty("div.access_font");  
Nifty("div.access_font_border");
```

All'interno del file atto a contenere gli elementi che dovranno essere *arrotondati* sarà sufficiente indicare, come già spiegato, il tipo dell'elemento e il suo identificatore.

#### 4.5.2 Arrotondamento degli angoli con HTML/CSS

Come già detto precedentemente lo *Nifty Corners Cube* non supporta bordi lineari, di conseguenza per la realizzazione delle *tab* di selezione delle aree didattiche è stato necessario l'ausilio di un'altra tecnica. Per far questo è stata utilizzata una combinazione di HTML e CSS.

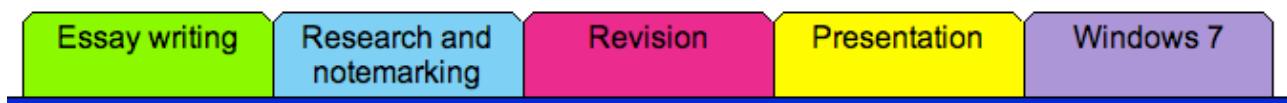
Il funzionamento è molto semplice basta creare un insieme di righe unitarie al di sopra dell'elemento desiderato. Ognuna di queste righe sarà dello stesso colore dell'elemento e di volta in volta la sua ampiezza si ridurrà di un'unità. Per il bordo sarà sufficiente assegnare un *bordo laterale* ad ognuna delle righe.

I lati negativi di questo tipo di tecnica sono l'innumerabile numero di linee di codice da inserire all'interno della pagina HTML, ogni qual volta si vorrà utilizzare l'effetto di *arrotondamento* si dovranno riutilizzare le stesse istruzioni, così facendo si avrà un inevitabile appesantimento del codice ed una maggior complessità d'uso rispetto a *Nifty Corners Cube*.

L'esempio seguente presenta un ingrandimento di un angolo, si possono vedere come le singole righe, di ampiezza diversa, formano l'effetto arrotondato.



Qui invece è riportata la *barra dei tab di navigazione* che utilizza questa tecnica per implementare l'arrotondamento degli angoli con bordo:



## **Il codice**

Il codice qui riportato fa riferimento alla prima *tab* dell'esempio precedente, quella relativa a "Essay writing".

### *In HTML*

```
<b class="rtop">
    <b class="rr0"></b>
    <b class="rr1 first"></b>
    <b class="rr2 first"></b>
    <b class="rr3 first"></b>
    <b class="rr4 first"></b>
</b>
```

La struttura del codice in HTML è molto semplice, si limita ad avere una struttura generale (identificata con `rtop`) che conterrà le relative righe, questo elemento è fondamentale per delineare il colore dello sfondo. Dopo di che si trovano le 5 righe (identificate da `rr0` a `rr4`) che consentono l'effettiva realizzazione del bordo arrotondato.

### *In CSS*

```
.rtop {
    display:block;
    background: #FFF;
}
```

Come già detto nella parte relativa al codice HTML, l'elemento caratterizzato da questo stile avrà il compito di indicare il colore dello sfondo.

```
.rtop b {
    display:block;
    height: 1px;
    overflow: hidden;
}
```

Queste sono regole generali che accomunano tutti gli elementi contenuti all'interno di `.rtop`. Principalmente sono impostati visibili (`display:block`) e di altezza 1px.

```
.rr0 {
    margin: 0 5px;
    border-bottom:solid 1px;
    background: #FFF;
}
```

Stile relativo alla prima *riga*, è di colore identico allo sfondo e il suo unico scopo è quello di fornire il bordo superiore della figura.

```

.rr1 {
    margin: 0 4px;
    border-right:solid 1px;
    border-left:solid 1px;
}

.rr2 {
    margin: 0 3px;
    border-right:solid 1px;
    border-left:solid 1px;
}

.rr3 {
    margin: 0 2px;
    border-right:solid 1px;
    border-left:solid 1px;
}

```

Queste tre istruzioni descrivono lo stile delle *righe* che si trovano nel mezzo. Come si può notare tutte hanno un bordo destro e sinistro. Ogni riga, partendo da rr1 fino ad arrivare a rr2, si allarga di un'unità per mezzo della proprietà margin.

```

.rr4 {
    margin: 0 1px;
    height: 2px;
    border-right:solid 1px;
    border-left:solid 1px;
}

```

Analogamente alle precedenti tre righe anche la quarta segue la stessa logica, con l'unica differenza che l'altezza di questo elemento sarà di 2px, questo per conferire una maggior armonia grafica.

```

.first {
    background-color: #7DFF00;
}

```

La classe ha scopo puramente grafico, essa infatti ha il compito di impostare il colore di sfondo negli elementi che costituiscono le *righe*.

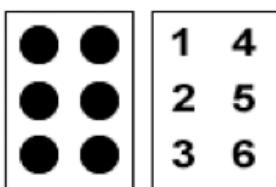
## Appendice A - Il Braille<sup>113</sup>

### Definizione

Il Braille è il codice di lettura e scrittura più utilizzato dai ciechi. È stato inventato dal francese Louis Braille nel 1829. Si tratta di un sistema di scrittura basato su sei punti in rilievo in cui la maggior parte dei simboli è universalmente riconosciuta e quindi può essere usato in molte lingue diverse. Il Braille infatti non è una lingua di per sé, ma è un mezzo di scrittura internazionale. Con esso si possono rappresentare le lettere dell'alfabeto, la punteggiatura, i numeri, i simboli matematici e quelli musicali.

### La cella Braille

I caratteri Braille sono basati su una cella formata da sei punti disposti in due colonne e tre righe.



I punti sono numerati dall'1 al 6, secondo lo schema appena illustrato. La dimensione di una cella Braille è standard, 7x4 millimetri, ed è una dimensione ottimale perché permette di percepire la sua intera superficie e allo stesso tempo di distinguere bene i singoli punti. Con i 6 punti si possono ottenere 64 combinazioni diverse che però non sono sufficienti a rappresentare tutti i caratteri. Ad esempio l'alfabeto del computer (il codice ASCII) ha 255 simboli. Per conservare la corrispondenza biunivoca tra questi segni ed i segni Braille si è pensato di arricchire il casellino o cella Braille, passando da 6 a 8 punti. Infatti, con 8 punti si possono rappresentare 255 segni.

|   |   |   |   |
|---|---|---|---|
| 1 | • | • | 4 |
| 2 | • | • | 5 |
| 3 | • | • | 6 |
| 7 | • | • | 8 |

Nonostante l'introduzione della cella a 8 punti la rappresentazione più comune e maggiormente utilizzata rimane quella a 6 punti.

### L'alfabeto

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| • a | • b | • c | • d | • e | • f | • g | • h | • i | • j |
| • k | • l | • m | • n | • o | • p | • q | • r | • s | • t |
| • u | • v | • w | • x | • y | • z |     |     |     |     |

<sup>113</sup> Fonte: Unione Italiana Ciechi, sezione di Frosinone. Sito ufficiale <http://www.uicfrosinone.it/>

## **Appendice B - Classificazione Accessive Devices (Dispositivi Assistivi)**

### **Specific ISO 9999:2007 Classification of Technical Aids**

#### **Ausili per trattamenti sanitari personali**

Ausili che servono per migliorare, monitorare o mantenere le condizioni di salute della persona. Sono esclusi gli strumenti usati esclusivamente da operatori sanitari.

#### **Ausili per l'esercizio di abilità**

Ausili finalizzati a migliorare le abilità fisiche, mentali e sociali della persona. Gli ausili che hanno una funzione primaria diversa rispetto all'addestramento, ma che possono essere usati anche per tale scopo, sono classificati nella classe che meglio caratterizza la loro funzione principale.

#### **Ortesi e protesi**

Le ortesi o dispositivi ortesici sono dispositivi applicati esternamente al corpo per modificare le caratteristiche strutturali e funzionali degli apparati neuromuscolare e scheletrico. Le protesi o dispositivi protesici sono dispositivi applicati esternamente al corpo per sostituire completamente o parzialmente una sua parte mancante o danneggiata. Sono comprese, ad esempio, le protesi funzionali, le protesi cosmetiche e le calzature ortopediche. Sono invece escluse le endoprotesi, che non sono contemplate da questo standard internazionale.

#### **Ausili per la cura e la protezione personale**

Sono compresi, ad esempio, gli ausili per vestirsi e svestirsi, per la protezione del corpo, per l'igiene personale, per la cura delle tracheostomie, delle stomie e dell'incontinenza, per la misurazione delle proprietà fisiche e fisiologiche e per le attività sessuali.

#### **Ausili per la cura della casa**

Sono compresi, ad esempio, gli ausili per mangiare e bere.

#### **Mobilia e adattamenti per la casa o per altri ambienti**

Sono compresi, ad esempio, i mobili con o senza ruote per riposare o lavorare, gli accessori per mobilia, gli ausili e le attrezzature per l'adattamento di ambienti residenziali, professionali e scolastici.

#### **Ausilio per comunicazioni, informazione e segnalazione**

Ausili che aiutano la persona a ricevere, inviare, produrre o elaborare informazioni secondo varie modalità. Sono compresi, ad esempio, i dispositivi per vedere, sentire, leggere, scrivere e telefonare, i dispositivi di segnalazione e di allarme, gli ausili informatici.

#### **Ausili per manovrare oggetti o dispositivi**

#### **Adattamenti dell'ambiente, utensili e macchine**

Dispositivi e apparecchiature che servono a migliorare l'ambiente personale nella vita quotidiana, utensili manuali e macchinari motorizzati. Sono escluse le apparecchiature finalizzate al miglioramento complessivo dell'ambiente.

#### **Ausili per le attività di tempo libero**

Dispositivi finalizzati a giochi, hobby, sport e altre attività per il tempo libero.

## **Appendice C - Legge Stanca - Legge 9 gennaio 2004, n. 4**

**Pubblicata in G.U. n. 13 del 17 gennaio 2004**

**La Camera dei deputati ed il Senato della Repubblica hanno approvato;**

### **IL PRESIDENTE DELLA REPUBBLICA Promulga**

**la seguente legge:**

#### **Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici**

##### **Art. 1 (Obiettivi e finalità)**

1. La Repubblica riconosce e tutela il diritto di ogni persona ad accedere a tutte le fonti di informazione e ai relativi servizi, ivi compresi quelli che si articolano attraverso gli strumenti informatici e telematici.
2. È tutelato e garantito, in particolare, il diritto di accesso ai servizi informatici e telematici della pubblica amministrazione e ai servizi di pubblica utilità da parte delle persone disabili, in ottemperanza al principio di uguaglianza ai sensi dell'articolo 3 della Costituzione.

##### **Art. 2 (Definizioni)**

1. Ai fini della presente legge, si intende per:

- a) «accessibilità»: la capacità dei sistemi informatici, nelle forme e nei limiti consentiti dalle conoscenze tecnologiche, di erogare servizi e fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari;
- b) «tecnologie assistive»: gli strumenti e le soluzioni tecniche, hardware e software, che permettono alla persona disabile, superando o riducendo le condizioni di svantaggio, di accedere alle informazioni e ai servizi erogati dai sistemi informatici.

##### **Art. 3 (Soggetti erogatori)**

1. La presente legge si applica alle pubbliche amministrazioni di cui al comma 2 dell'articolo 1 del decreto legislativo 30 marzo 2001, n. 165, e successive modificazioni, agli enti pubblici economici, alle aziende private concessionarie di servizi pubblici, alle aziende municipalizzate regionali, agli enti di assistenza e di riabilitazione pubblici, alle aziende di trasporto e di telecomunicazione a prevalente partecipazione di capitale pubblico e alle aziende appaltatrici di servizi informatici.

2. Le disposizioni della presente legge in ordine agli obblighi per l'accessibilità non si applicano ai sistemi informatici destinati ad essere fruiti da gruppi di utenti dei quali, per disposizione di legge, non possono fare parte persone disabili.

#### **Art. 4 (Obblighi per l'accessibilità)**

1. Nelle procedure svolte dai soggetti di cui all'articolo 3, comma 1, per l'acquisto di beni e per la fornitura di servizi informatici, i requisiti di accessibilità stabiliti con il decreto di cui all'articolo 11 costituiscono motivo di preferenza a parità di ogni altra condizione nella valutazione dell'offerta tecnica, tenuto conto della destinazione del bene o del servizio. La mancata considerazione dei requisiti di accessibilità o l'eventuale acquisizione di beni o fornitura di servizi non accessibili è adeguatamente motivata.
2. I soggetti di cui all'articolo 3, comma 1, non possono stipulare, a pena di nullità, contratti per la realizzazione e la modifica di siti INTERNET quando non è previsto che essi rispettino i requisiti di accessibilità stabiliti dal decreto di cui all'articolo 11. I contratti in essere alla data di entrata in vigore del decreto di cui all'articolo 11, in caso di rinnovo, modifica o novazione, sono adeguati, a pena di nullità, alle disposizioni della presente legge circa il rispetto dei requisiti di accessibilità, con l'obiettivo di realizzare tale adeguamento entro dodici mesi dalla data di entrata in vigore del medesimo decreto.
3. La concessione di contributi pubblici a soggetti privati per l'acquisto di beni e servizi informatici destinati all'utilizzo da parte di lavoratori disabili o del pubblico, anche per la predisposizione di postazioni di telelavoro, è subordinata alla rispondenza di tali beni e servizi ai requisiti di accessibilità stabiliti dal decreto di cui all'articolo 11.
4. I datori di lavoro pubblici e privati pongono a disposizione del dipendente disabile la strumentazione hardware e software e la tecnologia assistiva adeguata alla specifica disabilità, anche in caso di telelavoro, in relazione alle mansioni effettivamente svolte. Ai datori di lavoro privati si applica la disposizione di cui all'articolo 13, comma 1, lettera c), della legge 12 marzo 1999, n. 68.
5. I datori di lavoro pubblici provvedono all'attuazione del comma 4, nell'ambito delle disponibilità di bilancio.

#### **Art. 5 (Accessibilità degli strumenti didattici e formativi)**

1. Le disposizioni della presente legge si applicano, altresì, al materiale formativo e didattico utilizzato nelle scuole di ogni ordine e grado.
2. Le convenzioni stipulate tra il Ministero dell'istruzione, dell'università e della ricerca e le associazioni di editori per la fornitura di libri alle biblioteche scolastiche prevedono sempre la fornitura di copie su supporto digitale degli strumenti didattici fondamentali, accessibili agli alunni disabili e agli insegnanti di sostegno, nell'ambito delle disponibilità di bilancio.

**Art. 6**  
**(Verifica dell'accessibilità su richiesta)**

1. La Presidenza del Consiglio dei ministri - Dipartimento per l'innovazione e le tecnologie valuta su richiesta l'accessibilità dei siti INTERNET o del materiale informatico prodotto da soggetti diversi da quelli di cui all'articolo 3.
2. Con il regolamento di cui all'articolo 10 sono individuati:
  - a) le modalità con cui può essere richiesta la valutazione;
  - b) i criteri per la eventuale partecipazione del richiedente ai costi dell'operazione;
  - c) il marchio o logo con cui è reso manifesto il possesso del requisito dell'accessibilità;
  - d) le modalità con cui può essere verificato il permanere del requisito stesso.

**Art. 7**  
**(Compiti amministrativi)**

1. La Presidenza del Consiglio dei ministri - Dipartimento per l'innovazione e le tecnologie, anche avvalendosi del Centro nazionale per l'informatica nella pubblica amministrazione di cui all'articolo 4, comma 1, del decreto legislativo 12 febbraio 1993, n. 39, come sostituito dall'articolo 176 del decreto legislativo 30 giugno 2003, n. 196:

- a) effettua il monitoraggio dell'attuazione della presente legge;
- b) vigila sul rispetto da parte delle amministrazioni statali delle disposizioni della presente legge;
- c) indica i soggetti, pubblici o privati, che, oltre ad avere rispettato i requisiti tecnici indicati dal decreto di cui all'articolo 11, si sono anche meritamente distinti per l'impegno nel perseguire le finalità indicate dalla presente legge;
- d) promuove, di concerto con il Ministero del lavoro e delle politiche sociali, progetti, iniziative e programmi finalizzati al miglioramento e alla diffusione delle tecnologie assistive e per l'accessibilità;
- e) promuove, con le altre amministrazioni interessate, sentita la Conferenza permanente per i rapporti tra lo Stato, le regioni e le province autonome di Trento e di Bolzano, l'erogazione di finanziamenti finalizzati alla diffusione tra i disabili delle tecnologie assistive e degli strumenti informatici dotati di configurazioni particolari e al sostegno di progetti di ricerca nel campo dell'innovazione tecnologica per la vita indipendente e le pari opportunità dei disabili;
- f) favorisce, di concerto con il Ministero del lavoro e delle politiche sociali e con il Ministro per le pari opportunità, lo scambio di esperienze e di proposte fra associazioni di disabili, associazioni di sviluppatori competenti in materia di accessibilità, amministrazioni pubbliche, operatori economici e fornitori di hardware e software, anche per la proposta di nuove iniziative;
- g) promuove, di concerto con i Ministeri dell'istruzione, dell'università e della ricerca e per i beni e le attività culturali, iniziative per favorire l'accessibilità alle opere multimediali, anche attraverso specifici progetti di ricerca e sperimentazione con il coinvolgimento delle associazioni delle persone disabili; sulla base dei risultati delle sperimentazioni sono indicate, con decreto emanato di intesa dai Ministri interessati, le regole tecniche per l'accessibilità alle opere multimediali;
- h) definisce, di concerto con il Dipartimento della funzione pubblica della Presidenza del Consiglio dei ministri, gli obiettivi di accessibilità delle pubbliche amministrazioni nello sviluppo dei sistemi informatici, nonché l'introduzione delle problematiche relative all'accessibilità nei programmi di formazione del personale.

2. Le regioni, le province autonome e gli enti locali vigilano sull'attuazione da parte dei propri uffici delle disposizioni della presente legge.

**Art. 8  
(Formazione)**

1. Le amministrazioni di cui all'articolo 3, comma 1, nell'ambito delle attività di cui al comma 4 dell'articolo 7 del decreto legislativo 30 marzo 2001, n. 165, nonché dei corsi di formazione organizzati dalla Scuola superiore della pubblica amministrazione, e nell'ambito delle attività per l'alfabetizzazione informatica dei pubblici dipendenti di cui all'articolo 27, comma 8, lettera g), della legge 16 gennaio 2003, n. 3, inseriscono tra le materie di studio a carattere fondamentale le problematiche relative all'accessibilità e alle tecnologie assistive.

2. La formazione professionale di cui al comma 1 è effettuata con tecnologie accessibili.

3. Le amministrazioni di cui all'articolo 3, comma 1, nell'ambito delle disponibilità di bilancio, predispongono corsi di aggiornamento professionale sull'accessibilità.

**Art. 9  
(Responsabilità)**

1. L'inosservanza delle disposizioni della presente legge comporta responsabilità dirigenziale e responsabilità disciplinare ai sensi degli articoli 21 e 55 del decreto legislativo 30 marzo 2001, n. 165, ferme restando le eventuali responsabilità penali e civili previste dalle norme vigenti.

**Art. 10  
(Regolamento di attuazione)**

1. Entro novanta giorni dalla data di entrata in vigore della presente legge, con regolamento emanato ai sensi dell'articolo 17, comma 1, della legge 23 agosto 1988, n. 400, sono definiti:

- a) i criteri e i principi operativi e organizzativi generali per l'accessibilità;
- b) i contenuti di cui all'articolo 6, comma 2;
- c) i controlli esercitabili sugli operatori privati che hanno reso nota l'accessibilità dei propri siti e delle proprie applicazioni informatiche;
- d) i controlli esercitabili sui soggetti di cui all'articolo 3, comma 1.

2. Il regolamento di cui al comma 1 è adottato previa consultazione con le associazioni delle persone disabili maggiormente rappresentative, con le associazioni di sviluppatori competenti in materia di accessibilità e di produttori di hardware e software e previa acquisizione del parere delle competenti Commissioni parlamentari, che devono pronunciarsi entro quarantacinque giorni dalla richiesta, e d'intesa con la Conferenza unificata di cui all'articolo 8 del decreto legislativo 28 agosto 1997, n. 281.

## **Art. 11 (Requisiti tecnici)**

1. Entro centoventi giorni dalla data di entrata in vigore della presente legge il Ministro per l'innovazione e le tecnologie, consultate le associazioni delle persone disabili maggiormente rappresentative, con proprio decreto stabilisce, nel rispetto dei criteri e dei principi indicati dal regolamento di cui all'articolo 10:

- a) le linee guida recanti i requisiti tecnici e i diversi livelli per l'accessibilità;
- b) le metodologie tecniche per la verifica dell'accessibilità dei siti INTERNET, nonché i programmi di valutazione assistita utilizzabili a tale fine.

## **Art. 12 (Normative internazionali)**

1. Il regolamento di cui all'articolo 10 e il decreto di cui all'articolo 11 sono emanati osservando le linee guida indicate nelle comunicazioni, nelle raccomandazioni e nelle direttive sull'accessibilità dell'Unione europea, nonché nelle normative internazionalmente riconosciute e tenendo conto degli indirizzi forniti dagli organismi pubblici e privati, anche internazionali, operanti nel settore.

2. Il decreto di cui all'articolo 11 è periodicamente aggiornato, con la medesima procedura, per il tempestivo recepimento delle modifiche delle normative di cui al comma 1 e delle innovazioni tecnologiche nel frattempo intervenute.

La presente legge, munita del sigillo dello Stato, sarà inserita nella Raccolta ufficiale degli atti normativi della Repubblica Italiana. È fatto obbligo a chiunque spetti di osservarla e di farla osservare come legge dello Stato.

Data a Roma, addì 9 gennaio 2004

CIAMPI  
Berlusconi, Presidente del Consiglio dei Ministri  
Stanca, Ministro per l'innovazione e le tecnologie

Visto, il Guardasigilli: Castelli