

Raccolta di domande in esami/Vero o Falso/Esercizi

- Qual è il numero massimo di voci in un menù?

Ideale non più di 7, ma si può arrivare ad un max di 10. Fino a 7 c'è una certa complessità da parte dell'utente, con un numero di voci maggiore inizia a diventare troppo difficile. Inoltre, una modifica in profondità è più facile, in larghezza no perché se non ho spazio non so cosa fare. Si fanno categorie quindi molto ampie, cercando di scongiurare il problema di doverne aggiungere. Tutto ciò per evitare il sovraccarico cognitivo dell'utente, quindi avere troppe informazioni da dover gestire e conseguente disorientamento.

- Quanto può essere profonda una struttura organizzativa gerarchica?

Al massimo 5 (tra 4 e 5 generalmente), la profondità corrisponde al numero di click che servono per passare dalla home alla pagina che ci interessa. Poco profonde e "ragionevolmente" ampia sarebbe l'ideale. Per controllare il numero di clic viene fatto un albero della pagina web, si parte dalla home e poi si trovano le pagine collegabili. A mano a mano che si scende i figli rappresentano i link cliccabili in quella determinata pagina. Un albero rappresenta il numero di clic nel sito che mi servono per passare dalla home alla foglia.

- Fare degli esempi di schemi organizzativi ambigui

Schemi per argomento (topic)

Schemi orientati al compito (task)

Schemi specifici per audience

Schemi metaforici (*metaphor driven*)

Schemi ibridi

- Discutere i metodi utilizzati per rendere una tabella accessibile
- Summary (che si usa solo in XHTML) oppure uno span descrittore della tabella e un tag aria per dargli la semantica di descrizione (*aria-described-by*) per HTML5
 - o Questa deve stare entro i 70 caratteri, massimo 100 per informazioni puntuali
- gli scope (assegna intestazione a celle) oppure gli headers (assegna celle a intestazione); o si usa il primo o il secondo.
- abbr (attributo ove presente e non tag, apposito per le tabelle per rendere la lettura più veloce con screen reader, fornisce intestazione abbreviata), evitare celle vuote inutili.
- Per HTML5 si possono mettere i data-title per garantire che siano leggibili con attributi semantici le singole righe (a condizione che il tipo di dati abbia senso per ragionarla in quel modo).
- Servono inoltre dove utile gli span lang per fare in modo le parti linguistiche della tabella vengano lette correttamente.

Nota interna:

- le parole che vedi in grassetto ha capito che le usa la prof per indicarci quando usare th, quindi non si inserisca , , etc. Rompe la separazione struttura/presentazione
- Differenza tra id e class

L'id permette di definire un identificatore univoco per un elemento all'interno di una pagina mentre class consente di definire delle classi ed applicarle ad un particolare elemento della pagina. Id può essere usato come ancora di un link e non può iniziare con un numero; generalmente, comincia con _ oppure con una lettera e può essere usato per specificare uno specifico elemento all'interno di un URL. In una pagina ce n'è sempre uno solo, mentre la class può apparire più volte in una pagina

Inoltre, ID ha valore maggiore nel calcolo delle specificità. Normalmente, id viene usato per indicizzare un singolo elemento e quindi, a livello di stile, specializza ulteriormente la composizione e la separazione

struttura-contenuto qualora si voglia indicare altro a livello estetico; la classe permette, per lo stesso motivo, di considerare gruppi di elementi e modularizzare la costruzione della pagina web sulla base delle esigenze che si hanno.

- Quali sono le tre domande più importanti alle quali devo saper rispondere, pena il fenomeno del disorientamento? (Riferimento: Schema a tre pannelli – Teoria)
- 1) Dove sono?
- 2) Di che cosa si tratta?
- 3) Dove posso andare?

Posso inoltre rispondere ad altre 4 domande:

- 4) Come sono arrivato fin qui?
- 5) Da chi è gestita questa pagina?
- 6) Dove posso trovare informazioni più approfondite?
- 7) Altre informazioni relative al particolare sito web

L'eccesso di collegamenti e di cammini esplorativi può scoraggiare l'utente e fargli perdere di vista lo scopo della ricerca. Il disorientamento è diretta conseguenza dell'aver troppi stimoli (cioè, il sovraccarico cognitivo) e occorre sempre domandarsi se in ogni pagina in cui l'utente può risponderci alle domande precedenti.

- Le WCAG richiedono di identificare in modo diverso i link visitati da quelli non visitati?

Vero, in quanto occorre come informazione aggiuntiva, rispondendo a "dove sono ora", tale da aiutare l'utente a distinguere cosa ha guardato da cosa non ha ancora esaminato. Il loro colore viene segnato come "convenzione interna" qualora ridefinito dal proprio sito, in quanto abitua l'utente ad una particolare abitudine all'interno del proprio sito; se modificato, tale comportamento deve rimanere vero su tutte le pagine.

- È buona regola segnalare i link che portano ad elementi diversi da pagine web (es. pdf)

Sì, è buona regola segnalare i link che portano ad elementi diversi da pagine web, come ad esempio file PDF, in modo che gli utenti siano consapevoli di ciò a cui stanno per accedere e possano decidere se vogliono o meno seguire il link. Ciò può essere fatto utilizzando descrizioni appropriate o utilizzando icone specifiche per tipi di file diversi; l'importante è non tradire le aspettative dell'utente, pertanto devo rispettare quanto sto offrendo allo stesso in quel momento.

Calcolare la specificità di queste regole CSS e indicare quale definisce l'effetto finale per un paragrafo con classe "orange" e attributo lang="en" contenuto dentro un <div> con classe "primelemento" (in altri termini definire il colore con cui testo e sfondo del paragrafo vengono visualizzati).

```
body > div.primelemento p{
  background-color: grey;
  color:white;
}
p[lang="en"] {
  background-color: black;
  color:red;
}
p.orange[lang="en"]{
  background-color: black;
  color:orange;
}
```

Calcolo → (#id, #attributi/classi, #tag) (sapendo che id hanno #, attributi sono con [], classi sono con il ".", mentre tag sono i tag HTML)

- 1) (0,1,3)
- 2) (0,1,1)
- 3) (0,2,1)

Immaginando siano applicati tutti e tre, prendo quello con specificità maggiore per ordine di rilevanza quindi il terzo (2 vs 1 e 1 vince per importanza rispetto al successivo, dove il primo ha 3).

Calcolare la specificità di queste regole CSS e indicare come viene visualizzata una cella di intestazione con classe "intestazione" contenuta dentro la parte tbody della tabella.

```
table th{
  background-color: grey;
  color:white;
}
table tbody th{
  background-color: white;
  color:blue;
}
table tbody th.intestazione{
  font-variant: small-caps;
}
```

ordine: id, attributi, tag. (Qui non è usato !important altrimenti l'ordine sarebbe: important, id, attributi, tag)

prima regola: 0 0 2

seconda regola: 0 0 3

terza regola: 0 1 3

Quindi per il calcolo della specificità, le regole vengono applicate in questo ordine: 1) terza regola 2) seconda regola 3) prima regola

Quindi la cella ha il background white, colore del testo blue e il font-variant: small-caps

17. Che colore è il th

```
th {
  background-color: #bbb;
  color: #000;
}
th[abbr="Lun"] {
  background-color: #555;
  color: #f00;
}
table tr th.lun{
  background-color: #333;
  color: #fff;
}

<th class="lun" abbr="Lun" scope="col">Lunedì</th>
```

(set di regole) priorità: #id, #attributi, #tag (in img prof: #fff)

Prima regola: (0,0,1)

Seconda regola: (0,1,1)

Terza regola: (0,1,3)

L'ordine di applicazione è quindi terza, seconda, prima e il colore è nero carbone (#333), mentre il testo è bianco (#fff). Non sono inoltre presenti !important, altrimenti l'ordine sarebbe stato (!important, id, attributi, tag).

- Le convenzioni interne devono sempre essere rispettate per evitare di disorientare l'utente

Vero, in quanto contribuiscono a familiarizzare con l'utente, creando uno schema mentale utile, tale da farsi una mappa del come navigarlo ed essere invogliato inconsciamente nel suo utilizzo. Ad esempio i link: quelli visitati hanno un colore, quelli non visitati ne hanno un altro

- È buona norma segnalare i link che portano a pagine in lingua diversa da quella del sito utilizzando le bandiere nazionali

Falso, in quanto le bandiere rappresentano il paese, non la lingua; per quello occorre inserire le iniziali della lingua. Per questo è importante utilizzare un metodo alternativo per segnalare la lingua della pagina, come ad esempio l'utilizzo di codici ISO delle lingue, o di scrittura in caratteri diversi per le lingue diverse. In generale è importante evitare di utilizzare elementi che possono essere percepiti come offensivi o discriminatori e garantire la massima accessibilità per tutti gli utenti.

- Nascondere anche parzialmente le informazioni del menù con un menù a tendina non è corretto

Vero; nascondere parzialmente le informazioni del menù con un menù a tendina può essere considerato scorretto in alcuni casi, perché può rendere difficile per gli utenti trovare le informazioni di cui hanno bisogno (e fa perdere il focus agli utenti). In generale, è importante che il contenuto sia facilmente accessibile e navigabile per gli utenti. Ciò significa che le informazioni dovrebbero essere organizzate in modo logico e facilmente comprensibile, e che gli utenti non dovrebbero dover fare clic su troppi elementi per trovare ciò che stanno cercando. Tuttavia, in alcuni casi, l'utilizzo di un menù a tendina può essere utile per rendere il layout più pulito e ordinato; sconsigliato innestare tendine che ne aprono altre (più livelli di tendine)

- Che cosa sono le convenzioni interne di un sito web e le convenzioni esterne?

- Interne: regole che valgono dentro al nostro sito e non in quello degli altri. Se sono molto particolari vanno spiegate. Queste non devono mai essere rotte.

- Esterne: posizione dei link, posizione degli oggetti, link sottolineati, regole valide per tutti i siti in generale. Tutto il bagaglio culturale che l'utente ha appreso durante la sua esperienza da navigatore. Seguono gli standard del mercato. Non sono controllabili ma devono essere gestite e si possono rompere se opportunamente giustificato.

- Un test dell'accessibilità di una pagina web può essere fatto in modo esaustivo?

Falso; un test completo per verificare l'accessibilità di una pagina web può essere complesso e richiedere tempo, poiché ci sono molte considerazioni da tenere in conto. Tuttavia, è possibile effettuare una serie di controlli per verificare che la pagina soddisfi i criteri di accessibilità.

Alcune delle cose da verificare includono:

- Utilizzo di etichette e descrizioni appropriate per gli elementi dell'interfaccia utente
- Utilizzo di un contrasto adeguato tra il testo e lo sfondo
- Supporto per la navigazione tramite tastiera
- Possibilità di ingrandire il testo senza che venga compromessa la leggibilità
- Utilizzo di tecnologie assistive come screen reader
- Possibilità di effettuare il test su diverse piattaforme e dispositivi

Si comprende che ci sono molti fattori da tenere in considerazione, oltre al fatto che il sito evolve e occorre costruire le pagine in modo tale da essere accessibili sin da subito per facilitare questo processo. Non si può mai sapere per certo se quanto fatto basti effettivamente.

Mettere in ordine di applicazione gli stili CSS: (scritti già nell'ordine corretto da me)

- Impostazioni di stile predefinite del browser
- Fogli di stile esterni definiti dall'autore (cioè, i file .css)
- Fogli di stile embedded definiti dall'autore

- Impostazioni di stile inline definite dall'autore della pagina (pericolose perché toccate dagli utenti, non dagli autori)
- Impostazioni personali dell'utente (es. zoom nelle pagine, contrasto per utenti ipovedenti/dislessici)

Ordine di applicazione con !important:

- Impostazioni di stile predefinite del browser
- Fogli di stile esterni definiti dall'autore (cioè, i file .css)
- Fogli di stile embedded definiti dall'autore
- Impostazioni di stile inline definite dall'autore della pagina
- Dichiarazioni definite con !important
- Impostazioni personali dell'utente

Quanto segue è invece l'ordine di *priorità* delle regole:

- Impostazioni personali dell'utente
- Impostazioni di stile inline definite dall'autore della pagina
- Fogli di stile embedded definiti dall'autore
- Fogli di stile esterni definiti dall'autore
- Impostazioni di stile predefinite del browser

Se venisse usato !important, l'ordine di priorità sarebbe:

- Impostazioni personali dell'utente
- Dichiarazioni definite con !important
- Impostazioni di stile inline definite dall'autore della pagina
- Fogli di stile embedded definiti dall'autore
- Fogli di stile esterni definiti dall'autore
- Impostazioni di stile predefinite del browser

- Come deve essere il contenuto del tag *title* per essere considerato corretto?

Il contenuto del tag title deve essere dal particolare al generale: avendo pochi caratteri a disposizione solo i primi del titolo sarebbero visibili e quindi potrei comunque preservare il contenuto informativo, composto del titolo della pagina/nome del sito e la descrizione della pagina stessa. Esso viene anche utilizzato dai motori di ricerca per comprendere l'argomento della pagina; per questo motivo, dovrebbe stare entro i 70 caratteri al fine di evitare di essere troncato nelle query di ricerca.

- Il contenuto del tag *description* influenza ricerca SERP?/ Il contenuto del metatag <description> influenza il posizionamento di una pagina web nelle pagine di risposta dei motori di ricerca?

No, poiché i metatag description vengono utilizzati dai motori di ricerca per produrre la descrizione (snippet) che compare sotto al titolo della pagina nella lista dei risultati prodotta a seguito di una query e influenza la visualizzazione dei risultati, non tanto il posizionamento.

- Che cosa si intende per responsive web?

Detto anche layout a variabilità controllata, identifica i cosiddetti punti di rottura/controllo/breakpoints per definire precisamente gli intervalli di variabilità in cui lo schermo cambia e quindi adattarsi alle varie dimensioni. In particolare, ha l'obiettivo di rendere l'esperienza utente ottimale in modo fluido e non dipendente dal dispositivo dell'utente. Questo assicura il contenuto sia sempre leggibile e navigabile.

-

- Un tag input può essere figlio di un tag form (in XHTML)?

Falso, in quanto un tag input (in XHTML) non può essere figlio diretto di un tag form, può invece comparire correttamente all'interno di un altro tag che possa essere figlio diretto di form, come fieldset o div.

Di fatto, un elemento come *form* in XHTML può contenere solo elementi di blocco, cosa che *input* non è. Viene consigliato di riportare gli elementi nei *fieldset*; quindi, anche per HTML5, a domanda, si risponda falso, dato che è meglio metterlo dentro un tag descrittivo.

Non-block-level elements (such as `<input>`) are not valid directly inside `<form>` tags until HTML5.

The statement is legitimate as far as the standard goes: in HTML 4.01, the [definition for `<form>`](#) specifies that it may only contain block elements or `<script>`. As far as what every browser in the world allows, it's fine.

I can only guess that they consider `<form>` to not be a layout tag at all, and they want all inline elements to be contained inside a block element.

- Le tabelle non andrebbero mai usate per svolgere compiti di layout

Vero, dato che questo andrebbe contro il principio di separazione tra struttura e presentazione; andrebbero utilizzate solo quando i dati da presentare devono essere riportati in forma tabellare. Si consideri inoltre che possono dare problemi per l'accessibilità (si riprenda parte della risposta successiva).

- Le tabelle sono un ostacolo alla facile comprensione per le persone con disabilità, quindi andrebbero evitate ove possibile

Vero, dato che spesso vengono utilizzate in modo improprio per motivi di layout. Si devono utilizzare quando si ha bisogno di dati in formato tabellare o comunque per organizzare più facilmente le informazioni e renderle comprensibili; normalmente, se si intende usarle, occorre strutturarle in modo accessibile (usando elementi semantici descrittivi quali `thead`/`tbody`/`tfoot`, usando gli `scope` oppure gli `headers`, usando un riassunto del suo contenuto tramite un tag `summary` per XHTML oppure un `id` e un `aria-describedby`, usare le abbreviazioni e attributi `lang` quando utile, etc.)

- I titoli delle pagine devono contenere informazioni dal generale al particolare

Falso, perché potrebbe causare problemi di accessibilità; il flusso di informazioni dovrebbe invece andare dal particolare al generale. Questo aiuta gli utenti a comprendere rapidamente il contenuto della pagina e a orientarsi all'interno del sito ed è utile anche per i motori di ricerca. Inoltre, se abbiamo pochi caratteri a disposizione, solo i primi caratteri del titolo sarebbero visibili e quindi andando dallo specifico al generico, andrei comunque "preservare" il contenuto informativo (si riprenda parte di una delle risposte sopra simile a questa).

- Descrivere la differenza di utilizzo dei metodi GET e POST in una form
- Metodo *GET*: è il predefinito. Viene utilizzato per leggere dati. Il browser allega la *stringa di query* all'interno dello stesso URL. In questo modo, *chiunque* abbia accesso a questo vede i dati.
 - `http://server/path/file.cgi?parametro=valore` (coppia chiave-valore messo dall'utente)
 - Se si usa il metodo *GET* la stringa viene inserita dal server in una variabile d'ambiente
 - Vantaggi:
 - è possibile fare cache con *get*, sotto forma di bookmark/segnalibro al link (perché i dati, inseriti una serie di volte, possono essere reinseriti sempre allo stesso modo);
 - sono richieste facili da costruire e facilmente condivisibili
 - Svantaggi:
 - limite alla lunghezza della stringa oppure alla lunghezza massima dell'URL

- devono essere usati solo per recuperare dati e non per modificare i dati sul server, poiché non sono idempotenti (cioè, più richieste identiche possono avere effetti diversi sul server).
- Metodo *POST*: viene usato per inviare dati. La stringa di query viene passata come input standard nel *corpo* della richiesta e non è visibile
 - Se si usa il metodo *POST* si deve leggere la stringa di query dall'input standard
 - Vantaggi:
 - Si possono inviare grandi quantità di dati
 - Sono più sicure, dato che i dati non sono visibili negli URL
 - Possono essere usate per modificare dati sul server, essendo che sono idempotenti
 - I dati vengono mandati ogni volta (il browser avviserà l'utente che i dati saranno inviati)
 - Possono codificare dati in binario
 - Svantaggi:
 - Sono più complesse da gestire e richiedono più risorse
 - Non vengono salvate in cache e sono più lente delle GET
 - Non è possibile salvarle come segnalibro
- La separazione tra struttura, contenuto e presentazione aiuta il posizionamento nelle pagine SERP

Falso, in quanto non è possibile separare struttura e contenuto, ma si ha tra struttura, presentazione e comportamento; ciò aiuta il posizionamento delle pagine, dato che i browser interpretando le informazioni/metadati in HTML, sono in grado di valutare "meglio" pagine con tale separazione. Nello specifico, una buona struttura è valutata positivamente (es. utilizzo di intestazioni oppure gerarchie di contenuti), una buona presentazione certamente aiuta a livello di formattazione e di esperienza utente, specie quando si adatta a vari dispositivi sulla base del Web Design Adattivo e un buon comportamento, che a livello di caricamento di pagine e facilità d'uso, influisce a livello di permanenza dell'utente e come primo impatto.

(Quindi, appunto, occhio che sta scritto "contenuto" e non "comportamento")

- L'uso dei menù a scomparsa non influisce sull'accessibilità di un sito

Falso; l'uso di menù a scomparsa può causare problemi di accessibilità tra cui: 1) se il menù è di scarsa qualità non tutte le sue voci possono essere raggiungibili mediante il tasto TAB 2) un menù non bene evidenziato può portare certi utenti a perdere informazioni importanti per la buona fruizione del contenuto. Di fatto, oltre a perdere il focus, l'ottica usabilità ne è compromessa; la navigazione a tastiera e l'evidenziazione fanno in modo che i contenuti possano essere più facilmente raggiunti.

- Descrivi quali sono i vantaggi di inserire le dimensioni di un'immagine (width e height) tramite il linguaggio HTML o CSS (risposta da inserire a mano)
 - HTML
 - Fa conoscere la dimensione dell'immagine al browser (senza scaricarla)
 - Ottimizza il caricamento/rendering della pagina, in quanto la specifica delle dimensioni permette al browser di riservare lo spazio necessario per visualizzare l'immagine prima del suo caricamento completo
 - CSS
 - Separazione tra struttura e presentazione definendo attributi custom/personalizzati all'immagine come voluto

- L'attributo `id` si differenzia dall'attributo `class` solo perchè permette di identificare un unico elemento e non un insieme di elementi

No (ovviamente, ci sono altri motivi spiegati sopra per cui *id* si differenzia da *class*, ma è il solo che frega)

- Il metodo GET (in un vero o falso)
 - o “permette di reinviare gli stessi dati al server senza ricompilare la form”
 - o “ha un limite nella lunghezza della query string”
- HTML5 definisce una gestione standard degli errori quindi ora non è più importante che il codice HTML sia valido
 - o Falso, tuttora è un problema non ancora risolto. Non esiste infatti un metodo universale di gestione degli errori, dato che i browser stessi e anche i singoli dispositivi (a seconda della loro “età”) possono non interpretare correttamente contenuto non valido e potrebbe comportare problemi al sito per come lo si era pensato.

In che colore viene scritto il testo? (rispetto ad un esempio che ha mostrato la prof, riportato qui sotto)

- 1) verde
- 2) arancio
- 3) rosso
- 4) bianco

```
<head>
<title>Specificità</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<style type="text/css" media="all">
  pre
  {
    color: green;
    background-color: black;
  }
  div.exception pre { color: red; }
  div#error pre { color: orange; }
  ol[id="debug_list"] li div > pre { color: white; }
</style>
</head>
<body>
  <ol id="debug_list">
    <li>
      <div id="error" class="exception">
        <pre>
          Divisione di 22 per 0
          Exception in thread "main" ExceptionDivideByZero_REV_1: Errore di
          divisione per 0! at ExceptionDivideByZeroClient REV 2.
```

La risposta arancio è corretta perché:

- il primo selettore colora tutto il testo di verde dei tag `pre` con un colore di sfondo nero
- il secondo selettore indicizza i tag `pre` nella classe `div.exception`, colorandolo di rosso
- il terzo selettore indicizza i tag `pre` all'interno dei singoli elementi `error` dentro ai `div`, colorandoli in arancio
- il quarto selettore considera tutti i `pre` figli di `div`, tutti questi figli di `li`, considerando l'attributo `[id="debug_list"]` di `ol`

Per *ereditarietà*, il più specifico è proprio il selettore che contiene il colore arancio (indicizzando gli elementi `error` dentro ai `div`).

Calcoliamo nell'esempio sopra le specificità (`id`, attributi/classi, tag) riferiti ai selettori da `pre` fino a `ol[id="debug_list"]`:

- 1) [0, 0, 1] → non ci sono `id` né attributi, ci sta solo il tag `pre`
- 2) [0, 1, 2] → non ci sono `id`, c'è la classe `div.exception`, e ci sono due tag (`div`, `pre`)
- 3) [1, 0, 2] → ci sta solo un `id` (`#error`), non ci sono attributi/classi, ci sono due tag (`div`, `pre`)
- 4) [0, 1, 3] → (`id` non ci sono `[id="debug_list"]` non conta come `id`, essendo di `debug`, l'attributo è `[id="debug_list"]`, ci sono tre tag [`ol`/`li`/`div`])

nav > a:hover::before

- 0 IDs
- 1 Classes, attributes and pseudo-classes
- 3 Elements and pseudo-elements

ul#primary-nav li.active

- 1 IDs
- 1 Classes, attributes and pseudo-classes
- 2 Elements and pseudo-elements

CSS Selector Specificity - Cheat Sheet

	*	id	classes	elements	result*
	a	b	c	d	
li	0	0	0	1	1
.lorem	0	0	10	0	10
#ipsum	0	100	0	0	100
style = " "	1000	0	0	0	1000
ul li	0	0	0	2	2
ul ol + li	0	0	0	3	3
ul ol li.red	0	0	10	3	13
li:first-line	0	0	0	2	2
#ipsum li	0	100	0	1	101
a:link	0	0	10	1	11

```
h1 {
  /* Native elements have a weight of 0001. */
}

::after {
  /* Pseudo-elements have a weight of 0001. */
}

.class {
  /* Class selectors have a weight of 0010. */
}

[type="text"] {
  /* Attributes selectors have a weight of 0010. */
}

a:hover {
  /* Pseudo-classes have a weight of 0010. */
}
```

ul#nav li.active a

Style attribute: 0, ID: 1, Class, pseudo-class, attribute: 1, Elements: 3

body.ie7.col_3 h2 ~ h2

Style attribute: 0, ID: 0, Class, pseudo-class, attribute: 2, Elements: 3

<li style="color: red;">

Style attribute: 1, ID: 0, Class, pseudo-class, attribute: 0, Elements: 0

ul > li ul li ol li:first-letter

Style attribute: 0, ID: 0, Class, pseudo-class, attribute: 0, Elements: 7

```
#id {
  /* Id's have a weight of 0100. */
}

/* Inline styles have a weight of 1000. */
<div style="color: red;">Text</div>

h1 {
  /* !important has a weight of 10000. */
  color: red !important;
}
```

HTML

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Page Title</title>
5 </head>
6 <body>
7
8 <h1>My First Heading</h1>
9 <p lang="en">My first paragraph.</p>
10
11 </body>
12 </html>
```

CSS 45 unsaved changes X

```
1 body > p{
2 background-color: grey !important;
3 color: white;
4 }
5
6 p[lang="en"]{
7 background-color: black;
8 color: orange;
9 }
```

My First Heading

My first paragraph.

Di fatto, l'important viene applicato solo alla proprietà marcata, mentre sugli altri calcoli prevale l'importanza solita.

- Nel caso ci siano due specificità con lo stesso valore che succede?
 - o Si prende per buona quella definita per ultima

Soluzione - B

```
body > div.primoelemento p{      (#id, #classi o attributi,
    background-color: grey;        #tag)
    color:white;
}                                  (0,1,3)
p[lang="en"] {
    background-color: black;      (0,1,1)
    color:red;
}
p.orange[lang="en"]{             (0,2,1) ←
    background-color: black;
    color:orange;
}
```



Soluzione - B

```
body div p{                      (#id, #classi o attributi,
    background-color: grey;      #tag)
    color:white;
}                                (0,0,3)
div.primoelemento p[lang="en"] {
    background-color: black;    (0,2,2) ←
    color:red;
}
div p.orange{                   (0,1,2)
    background-color: black;
    color:orange;
}
```



Soluzione - corpo - 1

```
<tbody>
  <tr>
    <th abbr="Lun" scope="row">Lunedì</th>
    <td colspan="3">Hip Hop</td>
    <td></td>
    <td>Pilates</td>
  </tr>
  <tr>
    <th abbr="Mar" scope="row">Martedì</th>
    <td colspan="2">Karate</td>
    <td>Hip Hop</td>
    <td></td>
    <td>Pilates</td>
  </tr>
```

	9.00-10.00	10.00-11.00	11.00-12.00	14.00-15.00	15.00-16.00
Lunedì		Hip Hop			Pilates
Martedì		Karate	Hip Hop		Pilates
Mercoledì			Hip Hop		
Giovedì		Karate		Ginnastica Artistica	
Venerdì			Hip Hop		Pilates

Soluzione - corpo - 2

```

<tr>
  <th abbr="Mer" scope="row">Mercoledì</th>
  <td></td> <td></td>
  <td>Hip Hop</td>
  <td rowspan="3">Ginnastica Artistica</td><td></td>
</tr><tr>
  <th abbr="Gio" scope="row">Giovedì</th>
  <td colspan="2">Karate</td><td></td><td></td>
</tr><tr>
  <th abbr="Ven" scope="row">Venerdì</th>
  <td></td><td></td><td>Hip Hop</td>
  <td>Pilates</td>
</tr></tbody></table>

```

	9.00-10.00	10.00-11.00	11.00-12.00	14.00-15.00	15.00-16.00
Lunedì			Hip Hop		Pilates
Martedì		Karate	Hip Hop		Pilates
Mercoledì			Hip Hop		
Giovedì		Karate		Ginnastica Artistica	
Venerdì			Hip Hop		Pilates

Soluzione - intestazioni

```

<table summary="... testo esplicativo ..." >
  <caption>Occupazione della palestra</caption>
  <thead>
    <tr>
      <td></td>
      <th scope="col" abbr="9-10">9.00-10.00</th>
      <th scope="col" abbr="10-11">10.00-11.00</th>
      <th scope="col" abbr="11-12">11.00-12.00</th>
      <th scope="col" abbr="14-15">14.00-15.00</th>
      <th scope="col" abbr="15-16">15.00-16.00</th>
    </tr>
  </thead>
</table>

```

	9.00-10.00	10.00-11.00	11.00-12.00	14.00-15.00	15.00-16.00
Lunedì			Hip Hop		Pilates
Martedì		Karate	Hip Hop		Pilates
Mercoledì			Hip Hop		
Giovedì		Karate		Ginnastica Artistica	
Venerdì			Hip Hop		Pilates

Si ricordi che:

- struttura/contenuto e presentazione separati: avvantaggiano nei motori di ricerca e diminuiscono il peso del sito
- struttura/contenuto e comportamento separati: non avvantaggiano nei motori di ricerca né diminuiscono il peso del sito
- presentazione e comportamento separati: avvantaggiano nei motori di ricerca e diminuiscono il peso del sito

Si ha infatti: struttura (HTML), presentazione (CSS) e comportamento (PHP, JS).

Il discorso della separazione è perchè tu non vada ad usare JS per fare struttura (che non verrebbe vista), o HTML per mettere colori o stili (che sporcano la leggibilità e manutenibilità del codice). Di base aggiungere script vari appesantisce il sito ed inoltre c'è il rischio che alcuni browser non riescano a leggerli quindi è meglio separarli così in quel caso la pagina resta accessibile.

Non posso separare struttura e contenuto, perché quest'ultimo è parte integrante della struttura.

- Prova a dare una definizione di accessibilità

L'accessibilità è un insieme di proprietà e regole da seguire per tutelare le varie categorie di utenti, in particolare quelli diversamente abili, rendendoli più facilmente usabili per tutte le categorie da diversi tipi di dispositivi. Più formalmente: "usabilità di un prodotto/servizio/strumento per persone con ampio raggio di capacità". Bisogna pensarci *dal primo momento*, seguendo le buone regole di progettazione; implementarla dopo può essere costoso in termini di tempo/denaro.

- Utilizzando il linguaggio HTML5 creare questa tabella e renderla accessibile.

Giocatori

Giocatore	Ruolo	Maglia	Punti	Squadra di provenienza
Simone Giannelli	Palleggiatore	6	872	Sir Safety Susa Perugia
Riccardo Sbertoli	Palleggiatore	8	393	Itas Trentino
Fabio Balaso	Libero	7	1606	Cucine Lube Civitanova
Leonardo Scanferla	Libero	24	623	Gas Sales Bluenergy Piacenza
Simone Anzani	Centrale	17	2276	Cucine Lube Civitanova
Gianluca Galassi	Centrale	14	1144	Vero Volley Monza
Alessandro Michieletto	Schiacciatore	5	960	Itas Trentino
Daniele Lavia	Schiacciatore	15	1604	Itas Trentino
Yuri Romanò	Schiacciatore	17	1865	Gas Sales Bluenergy Piacenza
Totale Punti			11343	

<h1>La squadra</h1>

In questa tabella vengono riassunti i dati dei giocatori in 5 colonne: nome, ruolo, maglia in nazionale, punti e squadra di provenienza.

<table id="tabellaGiocatori" aria-describedby="sumTabella">

<caption>Giocatori</caption>

<thead>

<tr>

<th scope="col">Giocatore</th>

<th scope="col">Ruolo</th>

<th scope="col">Maglia </th>

<th scope="col">Punti</th>

<th scope="col">Squadra di provenienza</th>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">Simone Giannelli</th>

<td>Palleggiatore</td>

<td>6</td>

<td>872</td>

<td>Sir Safety Susa Perugia</td>

```
</tr>
<tr>
  <th scope="row">Riccardo Sbertoli</th>
  <td>Palleggiatore</td>
  <td>8</td>
  <td>393</td>
  <td>Itas Trentino</td>
</tr>
<tr>
  <th scope="row">Fabio Balaso</th>
  <td>Libero</td>
  <td>7</td>
  <td>1606</td>
  <td>Cucine Lube Civitanova</td>
</tr>
<tr>
  <th scope="row">Leonardo Scanferla</th>
  <td>Libero</td>
  <td>24</td>
  <td>623</td>
  <td><span lang="en">Gas Sales Bluenergy</span> Piacenza</td>
</tr>
<tr>
  <th scope="row">Simone Anzani</th>
  <td>Centrale</td>
  <td>17</td>
  <td>2276</td>
  <td>Cucine Lube Civitanova</td>
</tr>
<tr>
  <th scope="row">Gianluca Galassi</th>
  <td>Centrale</td>
  <td>14</td>
  <td>1144</td>
  <td>Vero <span lang="en">Volley</span> Monza</td>
</tr>
<tr>
  <th scope="row">Alessandro Michieletto</th>
  <td>Schiacciatore</td>
  <td>5</td>
  <td>960</td>
  <td><span lang="en">Itas Trentino</span></td>
</tr>
<tr>
  <th scope="row">Daniele Lavia</th>
  <td>Schiacciatore</td>
  <td>15</td>
  <td>1604</td>
  <td><span lang="en">Itas Trentino</span></td>
```

```
</tr>
<tr>
  <th scope="row">Yuri Romanò</th>
  <td>Schiacciatore</td>
  <td>17</td>
  <td>1865</td>
  <td><span lang="en">Gas Sales Bluenergy</span> Piacenza</td>
</tr>
</tbody>

<tfoot>
<tr>
  <td colspan="3">Totale Punti</td>
  <td>11343</td>
  <td></td>
</tr>
</tfoot>
</table>
```

Alternativamente in HTML5 (aggiungendo data-title):

<p id="dTable" class="aiuti">In questa tabella vengono riassunti i dati dei giocatori in 5 colonne:
Giocatore, Ruolo, Maglia, Punti, Squadra di Provenienza</p>

```
<table aria-describedby="dTable">
  <caption>Giocatori</caption>
  <thead>
    <tr>
      <th scope="col">Giocatore</th>
      <th scope="col">Ruolo</th>
      <th scope="col">Maglia </th>
      <th scope="col">Punti</th>
      <th scope="col">Squadra di provenienza</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th data-title="Giocatore" scope="row">Simone Giannelli</th>
      <td data-title="Ruolo">Palleggiatore</td>
      <td data-title="Maglia">6</td>
      <td data-title="Punti">872</td>
      <td data-title="Squadra di provenienza"><span lang="en">Sir Safety</span> Susa
Perugia</td>
    </tr>
    <tr>
      <th data-title="Giocatore" scope="row">Riccardo Sbertoli</th>
      <td data-title="Ruolo">Palleggiatore</td>
      <td data-title="Maglia">8</td>
      <td data-title="Punti">393</td>
      <td data-title="Squadra di provenienza">Itas Trentino</td>
    </tr>
  </tbody>
</table>
```

```
</tr>
<tr>
  <th data-title="Giocatore" scope="row">Fabio Balaso</th>
  <td data-title="Ruolo">Libero</td>
  <td data-title="Maglia">7</td>
  <td data-title="Punti">1606</td>
  <td data-title="Squadra di provenienza">Cucine Lube Civitanova</td>
</tr>
<tr>
  <th data-title="Giocatore" scope="row">Leonardo Scanferla</th>
  <td data-title="Ruolo">Libero</td>
  <td data-title="Maglia">24</td>
  <td data-title="Punti">623</td>
  <td data-title="Squadra di provenienza">Gas <span lang="en">Sales Bluenergy</span>
```

Piacenza</td>

```
</tr>
<tr>
  <th data-title="Giocatore" scope="row">Simone Anzani</th>
  <td data-title="Ruolo">Centrale</td>
  <td data-title="Maglia">17</td>
  <td data-title="Punti">2276</td>
  <td data-title="Squadra di provenienza">Cucine Lube Civitanova</td>
</tr>
<tr>
  <th data-title="Giocatore" scope="row">Gianluca Galassi</th>
  <td data-title="Ruolo">Centrale</td>
  <td data-title="Maglia">14</td>
  <td data-title="Punti">1144</td>
  <td data-title="Squadra di provenienza">Vero <span lang="en">Volley</span> Monza</td>
</tr>
<tr>
  <th data-title="Giocatore" scope="row">Alessandro Michieletto</th>
  <td data-title="Ruolo">Schiacciatore</td>
  <td data-title="Maglia">5</td>
  <td data-title="Punti">960</td>
  <td data-title="Squadra di provenienza">Itas Trentino</td>
</tr>
<tr>
  <th data-title="Giocatore" scope="row">Daniele Lavia</th>
  <td data-title="Ruolo">Schiacciatore</td>
  <td data-title="Maglia">15</td>
  <td data-title="Punti">1604</td>
  <td data-title="Squadra di provenienza">Itas Trentino</td>
</tr>
<tr>
  <th data-title="Giocatore" scope="row">Yuri Romanò</th>
  <td data-title="Ruolo">Schiacciatore</td>
  <td data-title="Maglia">17</td>
  <td data-title="Punti">1865</td>
```

```

        <td data-title="Squadra di provenienza">Gas <span lang="en">Sales Bluenergy</span>
Piacenza</td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <th scope="row" colspan="3">Totale Punti</th>
        <td>11343</td>
        <td></td>
    </tr>
</tfoot>
</table>

```

Scrivere il codice XHTML che permette di visualizzare a video questa tabella. La tabella deve essere accessibile.

Occupazione delle aule				
Torre Archimede				
	Aula 1A150	Aula 1C150	Aula 1AD100	Aula 2BC
8.00-9.00	Esame di Tecnologie Web		Tutorato studenti	
9.00-10.00		Human and Machine Vision		
10.00-11.00		Human and Machine Vision		
11.00-12.00				Esame di Data Mining

<table summary="La tabella informa l'occupazione delle aule 1A150, 1C150, 1AD100, 2BC della Torre Archimede dalle ore 8 alle 12 indicando quale esame o corso si tiene">

<caption>Occupazione delle aule</caption>

<thead>

<tr>

<th colspan="5" scope="colgroup">Torre Archimede</th>

</tr>

<tr>

<th></th>

<th abbr="1A150" scope="col">Aula 1A150</th>

<th abbr="1C150" scope="col">Aula 1C150</th>

<th abbr="1AD100" scope="col">Aula 1AD100</th>

<th abbr="2BC" scope="col">Aula 2BC</th>

</tr>

</thead>

<tbody>

<tr>

<th abbr="8 - 9" scope="row">8.00-9.00</th>

<td rowspan="3">Esame di Tecnologie Web</td>

<td></td>

<td colspan="2">Tutorato studenti</td>

Recorded with iTop Screen Recorder


```
<th abbr="2BC" scope="col">Aula 2BC</th>
</tr>
</thead>
<tbody>
<tr>
<th abbr="8 - 9" scope="row">8.00-9.00</th>
<td rowspan="3">Esame di Tecnologie Web</td>
<td></td>
<td colspan="2">Tutorato studenti</td>
</tr>
<tr>
<th abbr="9 - 10" scope="row">9.00-10.00</th>
<td>Human and Machine Vision</td>
<td></td> <td></td>
</tr>
<tr>
<th abbr="10 - 11" scope="row">10.00-11.00</th>
<td>Human and Machine Vision</td>
<td></td>
<td rowspan="2">Esame di Data Mining</td>
</tr>
<tr>
<th abbr="11 - 12" scope="row">11.00-12.00</th>
<td></td> <td></td> <td></td>
</tr>
</tbody>
</table>
```

Recorded with iTop Screen Recorder

Commento:
Mancano tre xmidang

```
<table>
<tr>
<td></td><td>Lunedì</td><td>Martedì</td><td>
Mercoledì</td><td>Giovedì</td><td>Venerdì</td>
</tr>
<tr>
<td>Danza</td><td colspan="3">9.00-10.00</td><td></
td><td>10.00-11.00</td>
</tr>
<tr>
<td>Karate</td><td>10.00-11.00</td><td>12.00-13.00
</td><td></td><td>12.00-13.00</td> <td>10.00-11.00
</td>
</tr>
</table>
```

<table summary="Nella tabella vengono forniti gli orari degli allenamenti di Karate e danza. Un riga contiene gli orari riguardanti danza, mentre la seconda riguardanti karate, le colonne invece sono i giorni della settimana.">

```
<tr>
<th></th>
<th abbr="Lun" scope="col"> Lunedì </th>
<th abbr="Mer" scope="col"> Martedì </th>
<th abbr="Gio" scope="col"> Mercoledì </th>
<th abbr="Ven" scope="col"> Giovedì </th>
<th abbr="Sab" scope="col"> Venerdì </th>
</tr>
<tr>
<th scope="row"> Danza </th>
<td colspan="3">9.00-10.00</td>
<td></td> <td>10.00-11.00</td>
</tr>
<tr>
<th scope="row"> Karate </th>
<td>10.00-11.00</td> <td>12.00-13.00</td> <td></td> <td>12.00-13.00</td>
<td>10.00-11.00</td>
</tr>
</table>
```

	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì
Danza	9.00-10.00				10.00-11.00
Karate	10.00-11.00	12.00-13.00		12.00-13.00	10.00-11.00

Scrivere il codice XHTML che permette di visualizzare a video questa tabella. La tabella deve essere accessibile.

Occupazione della palestra					
	9.00-10.00	10.00-11.00	11.00-12.00	14.00-15.00	15.00-16.00
Lunedì	Hip Hop				Pilates
Martedì	Karate		Hip Hop		Pilates
Mercoledì			Hip Hop	Ginnastica Artistica	
Giovedì	Karate				
Venerdì			Hip Hop		Pilates

```
<table summary="Tabella rappresentatne l'occupazione della palestra durante i giorni dal lunedì al venerdì dalle 9:00 alle 12:00 e dalle 14:00 alle 16:00">
```

```
<caption>Occupazione della palestra</caption>
```

```
<thead>
```

```
<tr>
```

```
<th></th>
```

```
<th abbr=" 9 - 10" scope="col">9:00-10:00</th>
```

```
<th abbr=" 10 - 11" scope="col">10:00-11:00</th>
```

```
<th abbr=" 11 - 12" scope="col">11:00-12:00</th>
```

```
<th abbr=" 14 - 15" scope="col">14:00-15:00</th>
```

```
<th abbr=" 15 - 16" scope="col">15:00-16:00</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<th abbr="Lun." scope="row"> Lunedì </th>
```

```
<td colspan="3">HipHop</td>
```

```
<td></td>
```

```
<td>Pilates</td>
```

```
</tr>
```

```
<tr>
```

```
<th abbr="Mart." scope="row"> Martedì </th>
```

```
<td colspan="2">Karate</td>
```

```
<td>HipHop</td>
```

```
<td></td>
```

```
<td>Pilates</td>
```

```
</tr>
```

```
<tr scope="row">
```

```
<th abbr="Merc." scope="row"> Mercoledì </th>
```

```
<td><td/>
```

```
<td></td>
```

```
<td></td>
```

```
<td>HipHop</td>
```

```
<td rowspan="3">Ginnastica Artistica</td>
```

```
<td></td>
```

```
</tr>
```

```
<tr>
```

```
<th abbr="Giov" scope="row">Giovedì</th>
```

```
<td colspan="2">Karate</td>
```

```
<td></td>
```

```
<td></td>
```

```
</tr>
```

```
<tr>
```

```
<th abbr="Ven." scope="row">Venerdì</th>
```

```
<td></td>
```

```
<td></td>
```

```
<td>HipHop</td>
```

```
<td>Pilates</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

Creazione di una tabella accessibile

			Presente In			
Personaggio	Specializzazione	QI	Prima Stagione	Seconda Stagione	Terza Stagione	Quarta Stagione
Sheldon Cooper	Fisica	183	Presente			
Penny	Attrice	N.D.	Presente			
Leslie Winkle	Sconosciuta	N.D.	Assente	Comparsa		Presente
Bernadette Rostenkowski	Farmacia	100	Assente			Comparsa
Amy Farrah Fowler	Bioteologie	115	Assente		Comparsa	Presente
Stuart Bloom	Fumetteria	40	Assente			
Emily Sweeney	Sconosciuta	N.D.				
Nome	Specializzazione	QI	Prima Stagione	Seconda Stagione	Terza Stagione	Quarta Stagione

Soluzione (linguaggio XHTML):

<table id="tableResumeCharacters" summary="In questa tabella viene fatto un piccolo riassunto delle caratteristiche di alcuni dei personaggi presenti in The Big Bang Theory. Per ciascun personaggio viene presentata la specializzazione, il suo Quoziente Intellettivo, e la sua partecipazione o meno alle prime quattro stagioni della serie">

```

<caption>Personaggi della serie</caption>
<thead>
  <tr>
    <th colspan="3"></th>
    <th scope="colgroup" colspan="4">Presente In</th>
  </tr>
  <tr>
    <th scope="col">Personaggio</th>
    <th scope="col">Specializzazione</th>
    <th scope="col"><abbr title="quoziente intellettuale">QI</abbr></th>
    <th scope="col">Prima Stagione</th>
    <th scope="col">Seconda Stagione</th>
    <th scope="col">Terza Stagione</th>
    <th scope="col">Quarta Stagione</th>
  </tr>
</thead>

<tfoot>
  <tr>
    <td>Nome</td>
    <td>Specializzazione</td>
    <td><abbr title="Quoziente Intellettivo">QI</abbr></td>
    <td>Prima Stagione</td>
    <td>Seconda Stagione</td>
    <td>Terza Stagione</td>
  </tr>

```

```
<td>Quarta Stagione</td>
</tr>
</tfoot>

<tbody>
<tr>
<th scope="row" xml:lang="en" >Sheldon Cooper</th>
<td>Fisica</td>
<td>183</td>
<td class="centerText" colspan="4">Presente</td>
</tr>

<tr>
<th xml:lang="en" scope="row">Penny</th>
<td>Attrice</td>
<td><abbr title="Non Disponibile">N.D.</abbr></td>
<td class="centerText" colspan="4">Presente</td>
</tr>

<tr>
<th id="lelie" xml:lang="en" scope="row">Leslie Winkle</th>
<td>Sconosciuta</td>
<td><abbr title="Non Disponibile">N.D.</abbr></td>
<td class="centerText">Assente</td>
<td class="centerText" colspan="2">Comparsa</td>
<td class="centerText">Presente</td>
</tr>

<tr>
<th xml:lang="en" scope="row">Bernadette Rostenkowski</th>
<td>Farmacia</td>
<td>100</td>
<td class="centerText" colspan="3">Assente</td>
<td>Comparsa</td>
</tr>

<tr>
<th xml:lang="en" scope="row">Amy Farrah Fowler</th>
<td>Biotecnologie</td>
<td>115</td>
<td class="centerText" colspan="2">Assente</td>
<td class="centerText">Comparsa</td>
<td>Presente</td>
</tr>

<tr>
<th id="stuart" xml:lang="en" scope="row">Stuart Bloom</th>
<td>Fumetteria</td>
<td>40</td>
```

```
<td colspan="4" rowspan="2" class="centerText">Assente</td>
</tr>

<tr>
<th xml:lang="en" id="emily" scope="row">Emily Sweeney</th>
<td>Sconosciuta</td>
<td><abbr title="Non Disponibile">N.D.</abbr></td>
</tr>
</tbody>

</table>
```

La validazione del codice HTML e CSS di un sito web è condizione necessaria per garantire la sua accessibilità → Vero

- È condizione necessaria ma non sufficiente, poiché per arrivare ad avere una valutazione esaustiva dell'accessibilità di una pagina servono altri controlli (es. controllo umano), che si differenziano in base a scopo e contesto.

L'uso dei colori non influenza l'accessibilità del sito → Falso

- Falso; l'informazione non dovrebbe mai essere veicolata unicamente attraverso il colore e, per questo motivo, alcune precauzioni devono essere adottate per garantire che le informazioni siano accessibili a tutti gli utenti, compresi coloro che hanno difficoltà a vedere i colori.
 - o Ad esempio, è importante assicurarsi che il testo abbia un contrasto sufficiente rispetto allo sfondo per essere facilmente leggibile, e che ci siano alternative testuali per le informazioni presentate solo attraverso i colori. Inoltre, è importante testare il sito con diversi sistemi di aiuto per la vista per garantire che le informazioni siano accessibili a tutti gli utenti.

Cosa rappresenta l'espressione regolare
/^[([w\-\+\.])*\@([w\-\+\.])*\.[([w\-\+\.])*\\$]/ ?

☒ 1 Una URL, ed assicura che sia corretta

☐ 2 Una URL, ma non assicura che sia corretto

☐ 3 Un indirizzo email, ed assicura che sia corretto

☐ 4 Un indirizzo email, ma non assicura che sia corretto

Domanda inerente a PHP; fatta su Wooclap sia per lezioni PHP che JS, dunque la inserisco qui:

- Un indirizzo e-mail, ma non assicura che sia corretto
 - o Potrebbe non essere corretta perché non tiene conto di tutte le regole di sintassi valide per gli indirizzi e-mail. Alcune ragioni per cui questa regex potrebbe non essere corretta sono:
 - Non verifica se il nome utente (la parte prima della " @ ") è lungo almeno un carattere e non contiene caratteri non validi come spazi o caratteri speciali.
 - Non verifica se il dominio (la parte dopo la " @ ") è lungo almeno due caratteri e non contiene caratteri non validi come spazi o caratteri speciali.
 - Non verifica se il TLD (estensione del dominio) è lungo almeno due caratteri e non contiene caratteri non validi come spazi o caratteri speciali.

- Non verifica se l'indirizzo e-mail contiene caratteri di commento, come "+" o "." prima della "@"
 - Non verifica se l'indirizzo e-mail contiene il carattere di commento "." alla fine del nome utente o del dominio.
 - Non verifica se l'indirizzo e-mail contiene caratteri di commento ripetuti, come ".." o "--".
- Convenzioni
 - o Le convenzioni interne → devono sempre essere rispettate
 - o Le convenzioni esterne → possono essere rotte se questo ha un tornaconto
 - Metafora della pesca
 - o Tiro perfetto → L'utente sa esattamente cosa sta cercando
 - o Trappola per aragoste → L'utente ha un'idea abbastanza chiara ma si aspetta di imparare qualcosa durante la ricerca
 - o Pesca con la rete a strascico → L'utente non lascia intentata nessuna possibilità
 - o Boa di segnalazione → L'utente vuole ritrovare in un secondo momento qualcosa che ha già letto
 - Inserire un esempio di schema per ognuna delle due tipologie:
 - Schemi esatti
 - o Schema alfabetico
 - Schemi ambigui
 - o Schema metaforico
 - È sempre possibile usare uno schema esatto per il menu principale → Falso

Gli schemi esatti servono quando l'utente sa esattamente cosa sta cercando; l'organizzazione della struttura organizzativa, cioè come i collegamenti sono organizzati tra di loro, dipende dal contesto che si vuole creare. Normalmente, una buona strategia può essere quella di adottare uno schema ibrido; si pensi ai siti di notizie, che hanno i principali fatti in ordine cronologico (schema esatto) ma organizzazione in categorie contestuali ai fatti più rilevanti o a tematiche di raggruppamento (ambiguo, perché dipendente dal contesto), per meglio adattarsi alle esigenze dell'utente.

Rimane il fatto che normalmente si può sempre adottare uno schema ambiguo per il menù principale.

- È sempre possibile usare uno schema ambiguo per il menu principale → Sì

Di fatto, gli schemi ambigui si adattano al fatto che l'utente non abbia completa conoscenza di ciò che vuole, pertanto questo tipo di schema organizzativo, cioè come sono riportate le informazioni, va sempre bene (per diretta conseguenza di quanto scritto appena sopra)

- Sulla base della risposta data alla domanda precedente, fai un esempio di schema adatto per la navigazione principale specificandone contesto e tipologia (ambiguo, esatto)
- - o Schema esatto: Ricette ordinate alfabeticamente
 - o Schema ambiguo: Sito con categorie ordinate per area privato/area azienda, suddivisione per argomenti degli articoli di giornale
- Quale struttura organizzativa è più adatta per la navigazione principale? → Gerarchia
- Come deve essere una gerarchia? → Deve essere ampia o molto profonda

- Quali sono i pregi e i difetti di avere una gerarchia molto ampia o molto profonda?
- Ampiezza
 - o Pro: Buona divisione informazioni e buona varietà, meno click per trovare info, con possibilità e specializzazione per le varie scelte
 - o Contro: possibile sovraccarico cognitivo (troppe informazioni) e tempo di caricamento lungo (ci sono tanti contenuti)
- Profondità
 - o Pro: Struttura più organizzata e focalizzata su poche informazioni, accedendo a cose specifiche
 - o Contro: Gli utenti possono avere difficoltà a trovare ciò che stanno cercando
- Datemi una definizione di area sicura
 - o Pixel disponibili per visualizzare l'informazione iniziale di una qualsiasi pagina web senza scroll da ogni dispositivo (quindi, che vedo senza nessuna interazione); da questa distinguiamo l'area visibile, cioè la prima vista al caricamento di una pagina. Questi pixel dipendono dal tipo di dispositivo e dalle sue dimensioni/impostazioni; per la stampa, corrisponde ai punti di schermo stampabili su supporto cartaceo.
 - o Di fatto, "above the fold" ed "area sicura" sono la stessa cosa e sono un fattore di grande impatto come prima impressione per l'utente web.
- Cosa mettere "above the fold"
 - o Tutto, ma non è realistico.
 - Identificazione del sito: titolo, logo
 - Parte di navigazione principale: menù (desktop) o menù hamburger (mobile)
 - Contenuti informativi più importanti (informazioni, CTA, etc.)
- Qual è la profondità massima accettabile per una gerarchia? (Si può inserire solo un numero) → 5 (se si hanno informazioni particolarmente puntuali, si sale fino a 7)
- Qual è la larghezza massima accettabile per una gerarchia? → 10 (entro le 7 è meglio, ma non è il massimo)
- Quale di questi titoli è un titolo corretto?

Opzioni (corretta è sottolineata; non essendo nella homepage, si accetta quella perché si presume l'utente sappia di cosa sta parlando):

- Unione Italiana dei Ciechi e degli Ipovedenti ONLUS-APS
- Documentazione – UIC
- Documentazione - Unione Italiana dei Ciechi e degli Ipovedenti ONLUS-APS
- DOC – UIC
- Per la progettazione di un'interfaccia mobile si deve considerare un utente come?
 - o Poco attento, magari con le mani occupate, spesso in movimento; meglio dire "un occhio ed un pollice"

- Associa ad ogni punto di controllo i dispositivi che hanno uno schermo uguale o inferiore a quel numero di pixel:
 - 320 pixel
 - Piccoli schermi, telefoni usati in modalità portrait
 - 480 pixel
 - Piccoli schermi, telefoni usati in modalità landscape
 - 600 pixel
 - Piccoli tablet (Amazon Kindle) usati in modalità portrait
 - 768 pixel
 - Tablet da 10 pollici (iPad 1024 x 768) usati in modalità portrait
 - 1024 pixel
 - Tablet (iPad) usati in modalità landscape, piccoli desktop o portatili, in generale una finestra che non occupa tutto lo schermo in un qualsiasi schermo
 - 1200 pixel
 - Schermi grandi, pensato per computer ad alta definizione e/o desktop
- Il layout a schede va facilmente incontro a problemi di manutenzione nel tempo

Vero; ciò accade per una serie di motivazioni:

- Con l'aumentare del numero di schede, può diventare difficile per gli utenti navigare e trovare i contenuti che stanno cercando. Ciò può generare confusione e frustrazione e può richiedere uno sforzo significativo per riorganizzare e ristrutturare il layout a schede al fine di renderlo più facile da usare.
- Quando i contenuti vengono aggiunti, rimossi o modificati, può essere necessario aggiornare il layout a schede per riflettere tali cambiamenti. Ciò può comportare l'aggiornamento delle etichette e del contenuto delle singole schede, nonché l'aggiunta o la rimozione di schede (specie se strutturate in orizzontale), se necessario. Questa operazione può richiedere molto tempo e un'attenta cura dei dettagli per garantire che il layout a schede rimanga organizzato e facile da usare.
- Possono incorrere in problemi di manutenzione se non sono progettati tenendo conto dell'accessibilità. Ad esempio, i layout a schede possono non essere facilmente navigabili dagli utenti con disabilità visive o non funzionare bene su dispositivi con schermi piccoli. In questi casi, potrebbe essere necessario uno sforzo significativo per rendere il layout a schede più accessibile a una gamma più ampia di utenti.
- La divisione tra contenuto e presentazione diminuisce il peso totale di un sito web

Vero, in quanto consente di separare e ottimizzare i contenuti e le presentazioni in modo indipendente.

Quando il contenuto e la presentazione sono combinati nello stesso file o documento, può essere più difficile ottimizzare ogni aspetto del sito web per le prestazioni. Ad esempio, il contenuto può essere ottimizzato per la leggibilità e la chiarezza, ma la presentazione può non essere ottimizzata per garantire tempi di caricamento rapidi o un uso efficiente delle risorse. Se invece si struttura tutto in modo definito (struttura, presentazione e comportamento), è possibile modificare i singoli moduli e quindi le unità di queste tre componenti, migliorando la definizione di quanto presente in una pagina web.

- La divisione tra contenuto e layout influenza il posizionamento di una pagina web nelle pagine di risposta dei motori di ricerca

Vero; è sempre buona norma dividere contenuto e presentazione; questo ha un effetto anche sul posizionamento del sito web perché i motori di ricerca raccolgono in modo automatico il contenuto della

pagina al fine della creazione dei loro indici (una buona separazione favorisce quindi un miglior posizionamento)

- Dare una definizione di linguaggio di markup, descriverne le principali caratteristiche e fornire alcuni esempi di linguaggi di markup conosciuti (evidenziando in modo opportuno le differenze)

Un linguaggio di markup è un linguaggio formato dalle regole che descrivono i meccanismi di rappresentazione (strutturali, semantici e presentazionali) di un dato strutturato, generalmente un documento quale ad esempio una pagina web. Il linguaggio individua delle parole chiave dette tag (marcatori) attraverso cui è possibile esplicitare il significato/funzione di una particolare area di testo.

Esistono due tipi di markup:

- *strutturale*, in cui possiamo distinguere un insieme di tag che condizionano in qualche modo il contenuto all'interno di essi. Qui si adottano degli *stili logici*, i quali descrivono il significato, il contesto o l'uso dell'elemento che racchiudono. Questi tag sono principalmente semantici:
 - o `<h1>`, `<p>`, ``, ``, etc.
- di *presentazione*, che dipende dal browser utilizzato (e ovviamente può essere modificato tramite CSS/fogli di stile), tuttavia ci sono delle convenzioni comuni. Esso utilizza etichette/tag CSS come *font-size*, *color*, *margin*, *padding* per descrivere e definire l'aspetto visivo della pagina.

La distinzione tra markup strutturale e di presentazione è importante perché permette di separare la logica del contenuto dalla logica della presentazione, rendendo più facile la manutenzione e la modifica del sito. Inoltre, il markup strutturale consente ai motori di ricerca e ai dispositivi di accessibilità di comprendere meglio il contenuto di una pagina, mentre il markup di presentazione consente di adattare l'aspetto visivo del sito a diverse risoluzioni e dispositivi.

- Dare una breve descrizione del Document Object Model(DOM). In particolare si indichi come questo viene rappresentato e quali sono le sue funzioni principali.

Il DOM (Document Object Model) è uno standard W3C che definisce un modello standard per l'accesso dinamico al contenuto di un documento (ad es. un file XML), consentendone la visualizzazione e l'aggiornamento di contenuto/struttura/presentazione, attraverso un'interfaccia neutrale rispetto ad uno specifico linguaggio di programmazione o scripting. Un esempio classico di utilizzo è l'accesso ad una particolare pagina XHTML-che verrà rappresentata come un albero a valle del "parsing" della pagina stessa- e che potrà essere visitata o modificata utilizzando opportuni metodi per l'accesso a ciascun nodo dell'albero, ad es. utilizzando JavaScript.

- In XHTML, il frammento `` risulta essere corretto

No, il tag così come riportato non è chiuso (in XHTML occorre chiudere tutti i tag)

- Descrivete la differenza tra linguaggi di tipo server side e client side

I linguaggi lato server sono linguaggi di programmazione che vengono eseguiti su un server, piuttosto che in un browser web. Vengono utilizzati per costruire siti e applicazioni web dinamiche e sono responsabili dell'elaborazione delle richieste lato server, dell'interazione con i database. Lo scripting lato server è una tecnica utilizzata nello sviluppo web che prevede l'impiego di script su un server web che produce una risposta personalizzata per ogni richiesta dell'utente (client) al sito web.

Alcuni esempi di linguaggi lato server sono PHP, ASP e Ruby.

I linguaggi lato client, invece, sono linguaggi di programmazione che vengono eseguiti in un browser web, piuttosto che su un server. Sono responsabili della gestione delle richieste e delle interazioni sul lato client e sono utilizzati per costruire applicazioni web interattive e reattive.

Alcuni esempi di linguaggi lato client sono CSS, C++, JavaScript.

In generale, i linguaggi lato server sono utilizzati per gestire la logica e le funzionalità back-end di un sito o di un'applicazione web, mentre i linguaggi lato client sono utilizzati per gestire il design front-end e l'interfaccia utente. Entrambi sono importanti per costruire un'applicazione web funzionale e facile da usare e spesso lavorano insieme per fornire un'esperienza utente senza soluzione di continuità.

- Le tabelle non causano mai problemi di accessibilità

Falso, le tabelle causano spesso problemi di accessibilità, soprattutto se utilizzate impropriamente per funzioni di presentazione (si vedano risposte date sopra).

- Consentire la fruibilità di una pagina attraverso lo screen reader è una condizione necessaria e sufficiente per definire una pagina accessibile

Consentire la fruibilità di una pagina attraverso uno screen reader è un aspetto importante dell'accessibilità web, ma non è l'unica condizione sufficiente per definire una pagina accessibile (è certamente necessaria).

L'accessibilità del web si riferisce alla pratica di rendere i siti e le applicazioni web utilizzabili dalle persone con disabilità, comprese quelle con problemi visivi, uditivi, motori e cognitivi. Si tratta di progettare e costruire siti web in modo da renderli utilizzabili e comprensibili al maggior numero possibile di persone, indipendentemente dalle loro capacità o dai loro dispositivi.

Permettere l'accesso tramite screen reader è una delle varie condizioni da rispettare per garantire accessibilità; ulteriori sono:

- La possibilità di utilizzare la pagina con uno screen reader o un'altra tecnologia assistiva
- la possibilità di navigare nella pagina utilizzando una tastiera o un altro dispositivo di input assistivo
- l'uso di titoli, etichette e testi di collegamento chiari e descrittivi
- L'uso di un adeguato contrasto di colori per rendere il testo e la grafica visibili agli utenti con disabilità visive.
- L'uso di un testo alt appropriato e descrittivo per le immagini e altri elementi non testuali.

Il rispetto di queste e altre linee guida sull'accessibilità può contribuire a garantire che una pagina sia utilizzabile e comprensibile per il maggior numero possibile di persone, comprese quelle con disabilità.

- La presenza di una barra con il path di contesto in una pagina web evita il sovraccarico cognitivo dell'utente

Falso; Il sovraccarico cognitivo si riferisce alla sensazione di essere sopraffatti o sovraccaricati di informazioni, che può portare a una diminuzione della produttività e a difficoltà di comprensione e interazione con un sito web o un'applicazione web.

Una barra di percorso contestuale è un elemento di navigazione che visualizza la gerarchia delle pagine all'interno di un sito o di un'applicazione web, partendo dalla pagina principale e scendendo attraverso i vari livelli di sottopagine. Questo può aiutare gli utenti a capire la struttura del sito e la relazione tra le diverse pagine, facilitando l'orientamento e la ricerca di contenuti specifici.

Fornendo agli utenti questo contesto e queste informazioni aggiuntive, una barra di percorso contestuale può contribuire a ridurre il sovraccarico cognitivo e a facilitare la navigazione e l'interazione con il sito. Tuttavia, è importante utilizzare le barre dei percorsi contestuali in modo appropriato e moderato, poiché informazioni eccessive o non necessarie possono comunque contribuire al sovraccarico cognitivo.

- Quali di questi test devono essere fatti per migliorare l'accessibilità di una pagina web?
 - Validità del codice HTML
 - Controllo esistenza di link circolari (È più un fattore di usabilità)
 - Controllo esistenza di link rotti (È più un fattore di usabilità)
 - Controllo presenza di alternative multisensoriali
 - Controllo della presenza e dell'adeguatezza degli attributi alt
 - Controllo estetica dei colori (Non interessa particolarmente)
- Che livello di accessibilità WCAG richiede la legge italiana ed europea? → AA (Doppia A)
- Un attributo aria-label può essere usato per associare un'etichetta ad un tag input? → No

È possibile farlo, ma è scorretto. Per il tag input, esiste già label che è semanticamente corretto, ma aria-label è il modo di farlo se non avessi nessun altro modo (ARIA serve solo a dare semantica quando non altrimenti possibile; in altri casi, bene usare i tag semantici).

- Tutti i test di accessibilità possono essere fatti in modo automatico? → Falso
- Le tabelle sono sempre un ostacolo alla facile comprensione del contenuto e per questo andrebbero evitate ove possibile → Vero

È bene evitare l'uso delle tabelle ove possibile, eventualmente è preferibile una loro linearizzazione, si utilizzino con criterio (ad esempio corredandole con un summary e adeguate intestazioni). Bene usarle quando i dati siano naturalmente disposti in forma tabellare e rappresentazioni alternative siano difficilmente gestibili o poco informative.

- Che cos'è uno schema organizzativo? Discutere le differenze tra schemi esatti e schemi ambigui riportando degli esempi

Servono per organizzare l'informazione prima sulla base di dati e poi sul sito web dove la visualizzo. Gli schemi organizzativi possono essere:

- Esatti, in questo caso definisco in maniera univoca dove viene posizionata l'informazione e quindi le classi sono mutuamente esclusive, sono comode per l'amministratore perché più mantenibili nel tempo, gli oggetti vanno sempre in una classe unica e non ho duplicazioni, richiedono tuttavia che l'utente sappia in modo molto preciso cosa sta cercando; alcuni esempi sono schemi geografici, schemi per ordine alfabetico, schemi temporali (per data, orario, ecc).
- Ambigui, i più utilizzati, in particolare perché rispondono alle esigenze di quegli utenti che non sanno di preciso cosa stanno cercando ma cercano in maniera generica qualcosa; l'informazione non viene inserita in modo univoco in un'unica categoria o classe ma potrebbe comparire anche in più di queste comportando problemi di duplicazione e maggiore difficoltà di mantenimento nel tempo oltre al fatto che persone differenti potrebbero posizionare una certa cosa in modo diverso, l'abbigliamento è uno dei casi classici poiché definire se un capo è un pantalone, un leggings o altro non è sempre immediato; le classi non sono mutuamente esclusive ma presentano potenziali intersezioni; alcuni esempi possono essere gli schemi metaforici (da usare con cautela), gli schemi per argomento, e a volte gli schemi suddivisi per

compiti o azioni (task). Ove realizzabile è buona norma fornire entrambi gli schemi organizzativi per soddisfare le esigenze del maggior numero di visitatori possibile, con l'accortezza però di tenerle ben separate e non mescolare risultati esatti con risultati ambigui.

- In tutte le pagine, le immagini che esprimono la validità HTML e CSS sono:
 - o Immagini di contenuto (tag `img`)



Prova a descrivere questa immagine per una persona non vedente che deve prenotare una camera.



Prova a descrivere questa immagine per una persona non vedente che deve prenotare una camera.



Prova a descrivere questa immagine per una persona non vedente che deve prenotare una camera.



Prova a descrivere questa immagine per una persona non vedente che deve prenotare una camera.

Utile ragionare per keywords per descrivere entrambi i contesti.

Per entrambi i contesti, utile definire:

- dettagli descrittivi (per esempio hotel e non quanti piani ha)
- dettagli ambientali primari (ad esempio le ombre per capire se c'è un ingombro oppure per capire se usare il cane/bastone) → caso seconda immagine – stanza luminosa, quella in alto a dx
- dettagli ambientali secondari (rappresenta ostacolo fisico) → caso seconda immagine
 - o utile descrivere anche la presenza del soffitto (da un punto di vista di altezze/ostacolo) → caso terza immagine, quella in basso a sx
 - o utile descrivere anche la vista marina/illuminazione ambientale/rumore → caso quarta immagine, quella in basso a dx

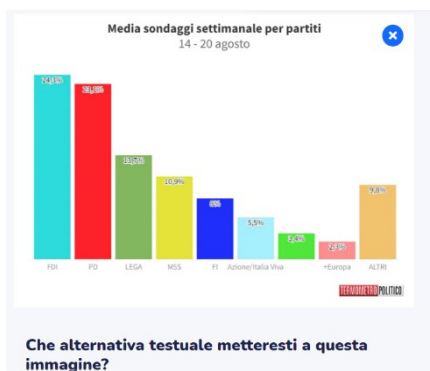
Gli attributi “alt”, dunque, devono essere sufficientemente descrittivi, tali da coprire esattamente con i dettagli principali quanto serve. Se si tratta di immagini di presentazione, è buona norma lasciare un “alt” vuoto per indicare che l'immagine non ha alcun contenuto informativo.



1) Logo di Twitter: un uccello azzurro



2) Grafico della funzione quadratica $y = x^2$ definita come parabola



3) Per descrivere tutti gli elementi di questo grafico, usiamo una tabella (tutte le colonne), leggendo colonne ed elementi singoli, assieme all'immagine (può avere anche alt vuoto)

Massimi caratteri di un alt: da 75 a 100 max

(da considerare quando si crea la tabella).

Usare anche una *caption* corretta.



La pagina nella figura è accessibile?

No, per un errato uso dei font (e anche di contrasto dei colori). Anche se ci sono componenti terze (banner/Google Maps, etc.), dobbiamo garantire accessibilità anche a quelle.

Quale di queste espressioni è corretta per iniziare una mail?

Cari studenti e care studentesse → Non si usa Carə studentə o Car* student* (non è accessibile, in quanto decisamente illeggibile dagli screen reader e assolutamente difficoltoso per utenti con dislessia).



Che emozioni sta usando DropBox (chiaramente, nel contesto dell'immagine precedente)?

- Rewards/Ricompense
- Status/Esclusività
- Anticipazione

L'espressione regolare
/a?+b*c?/
è corretta?

Il problema di questa regex è l'uso del simbolo "+" dopo il simbolo "?". Il simbolo "+" indica che il carattere o il gruppo precedente deve essere abbinato una o più volte, mentre il simbolo "?" indica che il carattere o il gruppo precedente è opzionale e deve essere abbinato zero o una sola volta. I simboli "+" e "?" non possono essere usati insieme in questo modo perché hanno significati contrastanti.

- Quali emozioni utilizza l'emotional design? Descrivere l'utilizzo riportando alcuni esempi

Il cosiddetto emotional design cerca di creare e disegnare prodotti/servizi al fine di creare specifiche emozioni negli utenti, in generale trasmettendo sensazioni positive e creando una buona esperienza. Le emozioni sono importanti e vanno sempre considerate; queste aiutano a trasformare un utente generico in un cliente/lettore e vengono normalmente usate per creare una connessione maggiore con l'utente. La base è non forzare mai l'utente al cambiamento, qualora si utilizzino le emozioni tramite il contrasto, dunque qualcosa che l'utente non si aspetta e che rompe un pattern conosciuto.

Ne esistono diverse:

- Sorpresa, quando si trova una deviazione da una situazione standard, spesso accompagnata da una sensazione piacevole; la cosa fondamentale è non tradire le aspettative dell'utente
- Status/Esclusività, cioè quando l'utente è familiarizzato o è spinto a tornare in quanto si sente esclusivo oppure coinvolto in quanto persona;
- Anticipazione, effetto che si ottiene quando si mostra una parte del tutto e lasciando correre l'immaginazione, suscitando curiosità ed aspettative all'utente finale
- Rewards/Ricompense, modo di spingere gli utenti all'acquisto/utilizzo di un servizio, portandoli ad una ricompensa quando viene svolta una particolare azione
- Qual è il media più accessibile?
 - o Testo
 - o Se non lo posso leggere, lo posso ascoltare/tradurre
 - o Comunque, è la principale alternativa che viene fornita al posto degli altri media

2. Specificità

Calcolare la specificità di queste regole CSS e indicare quale definisce l'effetto finale per un tag span con attributo lang="en" contenuto in un tag p con id "invidenza" e classe "riassunta" contenuto a sua volta dentro un tag div (in altri termini definire il colore con cui testo e sfondo dello span vengono visualizzati).

```
div p#invidenza span{
  background-color: black;
  color:white;
}
div p(class="riassunta") span{
  background-color: black;
  color:red;
}
div p(id="invidenza") span(lang="en"){
  background-color: white;
  color:blue;
}
```

Indicare la specificità di tutte le regole.

Prima regola: (1, 0, 3)

Seconda regola: (0, 1, 3)

Terza regola: (0, 2, 3)

Vince la prima regola, in quanto per importanza si dà più priorità ad id che al resto, per quanto per attributi venga superato dalle altre.

Colore di sfondo: nero

Colore del testo: bianco

4. Tabella Accessibile

Scrivere il codice HTML che permetta di visualizzare a video questa tabella. La tabella deve essere accessibile.

Medagliere Olimpiadi Tokyo 2021

Italia			
Medaglie d'Oro	Medaglie d'Argento	Medaglie di Bronzo	Totali
10	10	20	40
Atleta	Medaglia	Specialità	
Marcell Jacobs	2 medaglie d'Oro	100m maschile, staffetta 4x100m	
Vanessa Ferrari	1 medaglia d'argento	Ginnastica Artistica, corpo libero	
Gianmarco Tamberi	1 medaglia d'oro	Salto in alto maschile	

<table summary="Questa tabella descrive le medaglie d'oro, d'argento e di bronzo vinte dall'Italia alle Olimpiadi di Tokyo del 2021">

<caption>Medagliere Olimpiadi Tokyo 2021</caption>








<thead>

Scritto da Gabriel


```
<tr>
<th scope="colgroup" colspan="4">Italia</th>
</tr>
<tr>
<th scope="col">Medaglie d'Oro</th>
<th scope="col">Medaglie d'Argento</th>
<th scope="col">Medaglie di Bronzo</th>
<th scope="col">Totali</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<th scope="col">Atleta</th>
<th scope="col">Medaglie</th>
<th scope="col" colspan="2">Specialità</th>
</tr>
<tr>
<th scope="row" xml:lang="en">Marcell Jacobs</th>
<td>2 medaglie d'Oro</td>
<td colspan="2">100 <abbr title="metri">m</abbr> maschile, 4<abbr title="per">x</abbr>100<abbr title="metri">m</abbr></td>
</tr>
<tr>
<th scope="row">Vanessa Ferrari</th>
<td>1 medaglia d'argento</td>
<td colspan="2">Ginnastica Artistica, corpo libero</td>
</tr>
<tr>
<th scope="row">Gianmarco Tamberi</th>
<td>1 medaglia d'oro</td>
<td colspan="2">Salto in alto maschile</td>
</tr>
</tbody>
</table>
```

Alternativamente, notando che si mischiano sia titoli che contenuto, si può evitare di separare thead e tbody e inserire direttamente tutta la tabella con scope corretti e tutto quanto già presente ora.

Previsioni locali

Padova e pianura centrale	lunedì 24 pom/sera	martedì 25 mattina	martedì 25 pom/sera	mercoledì 26 mattina	mercoledì 26 pom/sera	giovedì 27	venerdì 28
stato del cielo							
temperatura	max 14/16 °C	min 8/10 °C max 20/22 °C		min 8/10 °C max 17/19 °C			
precipitazioni	Qualche pioggia	Qualche pioggia	Qualche pioggia	Qualche pioggia	Roveschi o temporali	Roveschi o temporali	Roveschi o temporali
probabilità precipitazioni	10%	10%	20%	30%	60%	50%	40%
attendibilità previsione		Ottima	Ottima	Ottima	Ottima	Buona	Buona

<table summary="Questa tabella contiene 8 colonne, in cui si descrivono le previsioni di Padova dal lunedì al venerdì in merito a temperatura e precipitazioni">

<thead>

<tr>

<th scope="col">Padova e pianura centrale</th>

<th scope="col">lunedì 24 <abbr title="pomeriggio">pom</abbr>/sera</th>

<th scope="col">martedì 25 mattina</th>

<th scope="col">martedì 25 <abbr title="pomeriggio">pom</abbr>/sera</th>

<th scope="col">mercoledì 26 mattina</th>

<th scope="col">mercoledì 26 <abbr title="pomeriggio">pom</abbr>/sera</th>

<th scope="col">giovedì 27</th>

<th scope="col">venerdì 28</th>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">stato del cielo</th>

<td></td>

<td></td>

<td></td>

<td></td>

<td></td>

<td rowspan="2"></td>

<td rowspan="2"></td>

</tr>

<tr>

<th scope="row">temperatura</th>

<td><abbr title="massimo">max</abbr>14/16 <abbr title="gradi">°C</abbr></td>

<td colspan="2"><p><abbr title="minimo">min</abbr>8/10 <abbr title="gradi">°C</abbr></p><p><abbr title="massimo">max</abbr>20/22 <abbr title="gradi">°C</abbr></p></td>

<td colspan="2"><p>min 8/10 <abbr title="gradi">°C</abbr></p><p>max 17/19 <abbr title="gradi">°C</abbr></p></td>

</tr>

<tr>

<th scope="row">precipitazioni</th>

<td>Qualche pioggia</td>

<td>Qualche pioggia</td>

<td>Qualche pioggia</td>

<td>Qualche pioggia</td>

```

<td>Rovescei o temporali</td>
<td>Rovescei o temporali</td>
<td>Rovescei o temporali</td>
</tr>
<tr>
<th scope="row">probabilità precipitazioni</th>
<td>10<abbr title="per cento">%</abbr></td>
<td>10<abbr title="per cento">%</abbr></td>
<td>20<abbr title="per cento">%</abbr></td>
<td>30<abbr title="per cento">%</abbr></td>
<td>60<abbr title="per cento">%</abbr></td>
<td>50<abbr title="per cento">%</abbr></td>
<td>40<abbr title="per cento">%</abbr></td>
</tr>
<tr>
<th scope="row">attendibilità previsione</th>
<td></td>
<td>Ottima</td>
<td>Ottima</td>
<td>Ottima</td>
<td>Ottima</td>
<td>Buona</td>
<td>Buona</td>
</tr>
</tbody>
</table>

```

		Giorni della settimana				
		Lunedì	Martedì	Mercoledì	Giovedì	Venerdì
Italiano	8.00	1s			3d	
	9.00		2f			1a
Geografia	8.00		4c	6d		
	9.00	2f			1a	

<table summary="La tabella descrive quando vengono svolte la materie Italiano e Geografia per le varie classe nelle ore 8 e 9 nei 5 giorni della settimana: Lunedì, Martedì, Mercoledì, Giovedì e Venerdì">

```

<thead>
<tr>
<th></th>
<th></th>
<th colspan="5" scope="colgroup">Giorni della settimana</th>
</tr>
<tr>
<th abbr="Lun" scope="col"> Lunedì</th>
<th abbr="Mar" scope="col">Martedì</th>
<th abbr="Mer" scope="col">Mercoledì</th>
<th abbr="Gio" scope="col">Giovedì</th>
<th abbr="Ven" scope="col">Venerdì</th>

```

```
</tr>
</thead>
<tbody>
<tr>
<th rowspan="2" scope="rowgroup">Italiano</th>
<th scope="row" abbr="8">8.00</th>
<td>1s</td>
<td></td>
<td></td>
<td>3d</td>
<td></td>
</tr>
<tr>
<th scope="row" abbr="9">9.00</th>
<td></td>
<td>2f</td>
<td></td>
<td></td>
<td>1a</td>
</tr>
<tr>
<th rowspan="2" scope="rowgroup">Geografia</th>
<th scope="row" abbr="8">8.00</th>
<td></td>
<td>4c</td>
<td>6d</td>
<td></td>
<td></td>
</tr>
<tr>
<th scope="row" abbr="9">9.00</th>
<td>2f</td>
<td></td>
<td></td>
<td>1a</td>
<td></td>
</tr>
</tbody>
</table>
```