

GUIDA COMPLETA ESAME SCRITTO

TECNOLOGIE WEB

Manuale operativo con domande ricorrenti e soluzioni complete

PARTE 1: STRUTTURA DELL'ESAME

Informazioni generali

DURATA: 1 ora e 15 minuti / 1 ora e 30 minuti

PUNTEGGIO MASSIMO: 32 punti totali

PUNTEGGI PER RISPOSTA:

- Risposta corretta: +2 punti
- Risposta sbagliata: -1 punto
- Risposta in bianco: 0 punti

Composizione esame

1. 5-6 domande VERO/FALSO motivate (valgono più di 2 punti ciascuna)
2. 1 esercizio CALCOLO SPECIFICITÀ CSS
3. 2 DOMANDE APERTE (risposte complete richieste)
4. 1 esercizio TABELLA ACCESSIBILE (codice XHTML/HTML valido)

Cosa NON viene chiesto

IMPORTANTE: L'esame NON chiede JavaScript o PHP avanzato. Al massimo può chiedere la differenza tra GET e POST.

Sessioni e bonus

- 5 scritti disponibili per anno accademico
- 3 consegne progetti per sessione (una consegna unica per sessione)
- 2 punti BONUS per chi consegna nella PRIMA sessione
- Possibilità di anticipo consegna per laureandi (contattare la prof)

PARTE 2: DOMANDE VERO/FALSO

IMPORTANTE: Ogni risposta V/F deve essere MOTIVATA in modo completo. La motivazione vale più della risposta stessa!

ACCESSIBILITÀ

D: Un test esaustivo dell'accessibilità può essere fatto in modo automatico?

R: FALSO

MOTIVAZIONE: I test automatici rilevano circa il 30% dei problemi di accessibilità. Sono rapidi ed economici ma non individuano tutti i problemi che sarebbero evidenti a un controllo umano. È sempre necessario l'intervento umano per verificare falsi positivi, validare che gli alt siano effettivamente descrittivi (e non solo presenti), testare la navigazione da tastiera completa, verificare il comportamento con screen reader reali, e controllare l'esperienza utente complessiva. I test manuali costituiscono la maggior parte del processo di verifica.

D: I test automatici sono sufficienti per verificare l'accessibilità?

R: FALSO

MOTIVAZIONE: Stesso concetto della domanda precedente. Gli strumenti automatici (WAVE, Lighthouse, AXE DevTools) sono utilissimi ma limitati. Non possono valutare la qualità semantica del contenuto, l'appropriatezza degli alt text, la logica della struttura delle intestazioni, o l'esperienza utente reale con tecnologie assistive.

D: La validazione del codice HTML e CSS è condizione sufficiente per garantire l'accessibilità?

R: FALSO

MOTIVAZIONE: La validazione W3C è una condizione NECESSARIA ma NON SUFFICIENTE. Un codice valido è la base, ma l'accessibilità richiede anche: alternative testuali appropriate, struttura semantica corretta, navigazione da tastiera completa, contrasti cromatici adeguati (minimo 4.5:1 per livello AA), uso corretto di ARIA, form con label associate, e molti altri requisiti che la validazione sintattica non controlla.

D: Permettere la fruibilità tramite screen reader è sufficiente per definire una pagina accessibile?

R: FALSO

MOTIVAZIONE: Il supporto per screen reader è NECESSARIO ma NON SUFFICIENTE. L'accessibilità comprende molte altre esigenze: utenti con disabilità motorie (navigazione tastiera), utenti con problemi di vista (contrast, zoom), utenti con disabilità cognitive (linguaggio chiaro, struttura logica), utenti anziani, utenti con tecnologie obsolete, utenti in contesti ambientali difficili. L'accessibilità è un concetto molto più ampio della sola compatibilità con screen reader.

D: Il contrasto dei colori influenza l'accessibilità?

R: VERO

MOTIVAZIONE: Il contrasto è fondamentale per l'accessibilità visiva. WCAG 2.1 livello AA richiede un rapporto di contrasto minimo di 4.5:1 per testo normale e 3:1 per testo grande. L'informazione non deve MAI essere veicolata solo attraverso il colore. Utenti con problemi di vista, daltonismo, o che usano schermi in condizioni di luce intensa necessitano di contrasti adeguati per leggere i contenuti.

D: L'AGID impone il livello AAA delle WCAG?

R: FALSO

MOTIVAZIONE: L'AGID (Agenzia per l'Italia Digitale) e la Legge Stanca in Italia richiedono il rispetto del livello AA delle WCAG 2.1, NON il livello AAA. Il livello AA rappresenta il requisito legale obbligatorio. Il livello AAA è considerato un obiettivo avanzato, eccellente ma non obbligatorio, e spesso difficile da raggiungere per tutti i contenuti.

STRUTTURA E ORGANIZZAZIONE

D: La struttura organizzativa gerarchica deve essere poco ampia e molto profonda?

R: FALSO

MOTIVAZIONE: È esattamente il contrario! Una struttura gerarchica efficiente deve essere AMPIA e POCO PROFONDA. L'ampiezza ideale è di 7 voci per livello (massimo 10), mentre la profondità ideale è di 3-4 livelli (massimo 5, eccezionalmente 7 per informazioni molto puntuali). Una profondità eccessiva richiede troppi clic e disorienta l'utente. Una struttura ampia e poco profonda è anche più facile da mantenere e modificare nel tempo, permettendo di aggiungere nuove sezioni senza riorganizzare tutta la gerarchia.

D: Il numero massimo di voci in un menù è 6?

R: FALSO (dipende dal numero proposto)

MOTIVAZIONE: Il numero ideale di voci in un menù è 7, con un massimo accettabile di 10. Questo limite deriva da studi di psicologia cognitiva (regola del 7 ± 2 di Miller). Con troppe voci l'utente subisce un sovraccarico cognitivo e fatica a scegliere. Con più di 10 voci si rischia lo scroll orizzontale e la perdita di orientamento. Meglio una struttura ampia in larghezza ma organizzata in categorie logiche.

CONVENZIONI

D: Le convenzioni interne non devono essere rotte per non disorientare l'utente?

R: VERO

MOTIVAZIONE: Le convenzioni interne sono regole che valgono all'interno del nostro sito e rappresentano comportamenti attesi dall'utente. Creare convenzioni interne coerenti aiuta l'utente a costruirsi una mappa mentale del sito e facilita la navigazione. Una volta stabilite, **NON** devono **MAI** essere rotte perché tradirebbero le aspettative dell'utente causando disorientamento e frustrazione. Esempio: se i link esterni sono sempre evidenziati con un'icona, questa convenzione deve essere mantenuta in tutto il sito.

DIFFERENZA CON CONVENZIONI ESTERNE: Le convenzioni esterne (standard web generali: logo in alto a sinistra che porta alla home, link sottolineati e colorati, carrello in alto a destra negli e-commerce) possono essere rotte **SOLO** se opportunamente giustificato e con un valido tornaconto, e **SEMPRE** comunicando chiaramente il cambiamento all'utente.

LINK

D: Per distinguere link visitati da non visitati è sufficiente cambiarne il colore?

R: FALSO

MOTIVAZIONE: Cambiare solo il colore NON è sufficiente per accessibilità. L'informazione non deve MAI essere veicolata unicamente attraverso il colore. Utenti daltonici o con problemi di percezione cromatica non potrebbero distinguere lo stato del link. È necessario fornire un'alternativa: sottolineatura, icona, peso del font diverso, o qualsiasi altro indicatore visivo NON basato solo sul colore. La combinazione colore + altro indicatore visivo garantisce l'accessibilità.

D: È buona norma segnalare link a pagine in lingua diversa con bandiere nazionali?

R: FALSO

MOTIVAZIONE: Le bandiere rappresentano il PAESE, NON la LINGUA. Questo può essere fuorviante e persino offensivo. Esempi: la lingua spagnola è parlata in molti paesi con bandiere diverse, l'arabo è lingua ufficiale di oltre 20 paesi, l'inglese è parlato in UK, USA, Australia, Canada, ecc. La soluzione corretta è usare: codici ISO delle lingue (IT, EN, ES), abbreviazioni testuali delle lingue, o scrivere il nome della lingua nei suoi caratteri nativi (English, Español, 日本語). Questo evita discriminazioni e garantisce chiarezza.

TABELLE

D: Le tabelle vanno sempre evitate?

R: FALSO

MOTIVAZIONE: Le tabelle NON vanno sempre evitate, ma vanno usate SOLO per dati tabulari reali, MAI per layout. L'uso scorretto delle tabelle per il layout è deprecato e crea gravi problemi di accessibilità. Quando sono necessarie per presentare dati in formato tabellare, le tabelle devono essere strutturate in modo accessibile usando: caption (titolo), thead/tbody/tfoot (struttura semantica), scope o headers sulle celle di intestazione (th), abbreviazioni (abbr) quando utili, summary (XHTML) o aria-describedby (HTML5), attributi lang quando necessario. Una tabella ben strutturata semanticamente È accessibile.

MENU E NAVIGAZIONE

D: Nascondere parzialmente le informazioni con menu a tendina non è corretto?

R: VERO

MOTIVAZIONE: I menu a tendina possono causare problemi di accessibilità: perdita di focus durante la navigazione, difficoltà con la navigazione da tastiera (non tutte le voci raggiungibili con Tab), contenuti nascosti che l'utente potrebbe non scoprire, complessità per utenti con difficoltà motorie o cognitive. È fortemente sconsigliato innestare tendine che aprono altre tendine (più livelli). Se proprio necessari, i menu a tendina devono essere implementati correttamente con: pieno supporto tastiera, focus visibile chiaro, timeout adeguati, indicatori evidenti della presenza di sottomenu.

D: L'uso dei menu a scomparsa non influisce sull'accessibilità?

R: FALSO

MOTIVAZIONE: I menu a scomparsa INFLUISCONO negativamente sull'accessibilità se non implementati correttamente. Problemi comuni: voci non raggiungibili via tastiera, perdita del focus dell'utente, menu poco evidenziati che nascondono informazioni importanti, difficoltà per utenti con disabilità motorie o cognitive. L'ottica di usabilità ne risulta compromessa. La navigazione da tastiera completa e l'evidenziazione chiara sono essenziali.

SEO E METATAG

D: Il metatag description influenza il posizionamento nelle SERP?

R: FALSO

MOTIVAZIONE: Il metatag description NON influenza direttamente il RANKING (posizionamento) nelle pagine dei risultati di ricerca (SERP). I motori di ricerca hanno smesso di considerare i metatag per il ranking perché erano troppo abusati. TUTTAVIA, la description influenza il CTR (Click-Through Rate): una description ben scritta, accattivante e con keywords può convincere l'utente a cliccare sul risultato anziché su altri. Quindi influenza INDIRETTAMENTE il traffico al sito, ma NON il posizionamento algoritmico.

D: Il metatag keywords influenza il posizionamento?

R: FALSO

MOTIVAZIONE: Il metatag keywords è completamente IGNORATO dai principali motori di ricerca (Google, Bing) dal 2009 circa, proprio perché era abusato con keyword stuffing. Non ha alcuna influenza sul posizionamento. È considerato obsoleto. Le keywords devono essere inserite naturalmente nel contenuto visibile della pagina: title, intestazioni H1-H6, paragrafi, alt delle immagini.

D: Il tag title deve essere sempre dal particolare al generale?

R: VERO

MOTIVAZIONE: Il title deve seguire la struttura PARTICOLARE → GENERALE perché: 1) Le schede del browser trancano i title lunghi mostrando solo l'inizio; 2) I risultati di ricerca mostrano circa 55-60 caratteri; 3) L'informazione più specifica e importante deve essere visibile subito. Esempio CORRETTO: "iPhone 15 Pro - Smartphone Apple - TechStore". Esempio SBAGLIATO: "TechStore - Smartphone Apple - iPhone 15 Pro". Il title è uno dei fattori SEO on-page più importanti.

SEPARAZIONE CONTENUTO/PRESENTAZIONE

D: La separazione tra struttura, contenuto e presentazione è importante?

R: VERO

MOTIVAZIONE: La separazione è FONDAMENTALE per molteplici ragioni: 1) ACCESSIBILITÀ: contenuto accessibile anche senza CSS; 2) SEO: i motori di ricerca leggono i primi byte, se pieni di CSS non leggono il contenuto; 3) MANUTENIBILITÀ: modifiche allo stile non richiedono modifica del contenuto; 4) PERFORMANCE: CSS esterni sono cacheable dal browser; 5) RIUSABILITÀ: stesso CSS applicabile a più pagine; 6) PROFESSIONALITÀ: codice più pulito, comprensibile e manutenibile.

D: La separazione tra contenuto e presentazione contribuisce ad abbassare il peso del sito?

R: VERO

MOTIVAZIONE: Sì, perché lo stesso file CSS esterno può essere applicato a multiple pagine ed è cacheable dal browser. Scaricato una volta, non viene riscaricato per le pagine successive. Inoltre, evitando stili inline ripetuti in ogni tag HTML, si riduce drasticamente la dimensione complessiva del codice. Un file CSS esterno di 50KB condiviso da 100 pagine è molto più efficiente di 100 pagine con 50KB di stili inline ciascuna.

LAYOUT

D: Il layout a schede è fortemente consigliato per tassonomie con molte voci?

R: FALSO

MOTIVAZIONE: Il layout a schede è SCONSIGLIATO per diverse ragioni: 1) Si sviluppa ORIZZONTALMENTE rischiando scroll laterale (da evitare sempre); 2) Con molte voci diventa inutilizzabile; 3) È poco FLESSIBILE e va facilmente incontro a problemi di manutenzione nel tempo; 4) Difficile aggiungere nuove voci senza ridisegnare tutto; 5) Non adatto a siti in evoluzione. Preferire menu verticali o dropdown ben strutturati.

PARTE 3: CALCOLO SPECIFICITÀ CSS

Regola di calcolo

La specificità si calcola con la quadrupla: (INLINE, ID, CLASSI/ATTRIBUTI, ELEMENTI)

- INLINE: stili direttamente nel tag HTML style="..." = 1 punto
- ID: selettori con # = 1 punto per ogni ID
- CLASSI/ATTRIBUTI: selettori con ., [attr], :pseudo-class = 1 punto cadauno
- ELEMENTI: selettori di tag (div, p, span) e ::pseudo-elementi = 1 punto cadauno

Ordine di importanza (dal più forte al più debole)

5. 1. !important UTENTE (massima priorità)
6. 2. Stili INLINE nel tag HTML
7. 3. Stili nel tag <style> (embedded/interni)
8. 4. Stili in file CSS esterno <link>
9. 5. !important AUTORE
10. 6. Stili di default del BROWSER

NOTA: A parità di specificità vince l'ultima regola dichiarata (ordine nel codice).

Esempi completi

ESEMPIO 1

CSS:

```
div.menu { background-color: black; color: white; }
```

```
div#principale { background-color: black; color: red; }
```

```
div#principale.menu span.evidenziato { background-color: white; color: blue; }
```

HTML:

```
<div id="principale" class="menu"><span  
class="evidenziato">Testo</span></div>
```

DOMANDA: Di che colore è il testo all'interno del tag ?

SOLUZIONE:

Prima regola: (0, 0, 1, 1) = 0011

Seconda regola: (0, 1, 0, 1) = 0101

Terza regola: (0, 1, 2, 3) = 0123

VINCE la TERZA regola: (0, 1, 2, 3) è maggiore delle altre

COLORE TESTO SPAN: BLUE (dalla terza regola)

ESEMPIO 2

CSS:

```
th { background-color: gray; }
```

```
th.lun { background-color: blue; font-size: 14px; }
```

```
table#calendario th.lun { background-color: green; font-size: 16px; }
```

HTML:

```
<table id="calendario"><tr><th class="lun"
abbr="Lun">Lunedì</th></tr></table>
```

DOMANDA: Di che colore è lo sfondo e che dimensione ha il font del <th>?

SOLUZIONE:

Prima regola: (0, 0, 0, 1) = 0001

Seconda regola: (0, 0, 1, 1) = 0011

Terza regola: (0, 1, 1, 2) = 0112

VINCE la TERZA regola: (0, 1, 1, 2) ha un ID quindi supera le altre

BACKGROUND: GREEN, FONT-SIZE: 16px (dalla terza regola)

PARTE 4: DOMANDE APERTE RICORRENTI

1. Metafora della pesca

DOMANDA: Descrivi la metafora della pesca e i tipi di utenti.

RISPOSTA COMPLETA:

La metafora della pesca descrive i diversi comportamenti degli utenti nella ricerca di informazioni su un sito web. Identifica quattro tipologie di utenti con strategie di ricerca diverse:

1) IL TIRO PERFETTO (Known-item seeking)

Gli utenti sanno ESATTAMENTE cosa stanno cercando. Utilizzano spesso una strategia di navigazione depth-first (vanno dritti al punto). Per questi utenti sono efficaci: casella di ricerca interna, alberatura piatta e chiara, schemi organizzativi esatti (alfabetico, cronologico). Esempio: cerco "iPhone 15 Pro 256GB Nero" su un e-commerce.

2) TRAPPOLA PER ARAGOSTE (Exploratory seeking)

Gli utenti hanno un'IDEA PRECISA di cosa cercano e si aspettano di IMPARARE qualcosa durante il processo esplorativo che li aiuti nella prossima iterazione della ricerca. Non sono insoddisfatti se non trovano subito, anzi sono contenti di scoprire informazioni utili. Utilizzano navigazione breadth-first. Per questi utenti servono: contenuti interessanti in ogni pagina, keywords evidenziate, esposizione per punti chiara, schemi ambigui, menu gerarchici da generale a specifico. Accettano più clic purché imparino qualcosa di utile. Esempio: cerco una palestra vicino a casa, scopro sport che non conoscevo e prezzi interessanti.

3) PESCA CON LA RETE (Exhaustive research)

Gli utenti NON lasciano nulla di intentato e vogliono esaminare TUTTO all'interno di un certo argomento. Guardano un po' tutto, non hanno un'idea precisa. Occorre fare in modo che non si perdano e non subiscano disorientamento. Servono: sistema di navigazione molto chiaro, breadcrumb ben visibile, indicatori di posizione, possibilità di salvare/bookmarking interno. Esempio: sto cercando informazioni su un argomento vasto e leggo molti articoli diversi.

4) BOA DI SEGNALE (Re-finding)

Gli utenti vogliono RITROVARE un elemento informativo già visto in precedenza. Servono strumenti come: bookmark/segnalibri del browser, wishlist/lista desideri interna al sito, cronologia, funzione "salva per dopo", evidenziazione automatica dei link visitati. Esempio: voglio ritrovare quell'articolo interessante letto la settimana scorsa.

2. Differenza tra GET e POST

DOMANDA: Spiega la differenza tra metodi HTTP GET e POST.

RISPOSTA COMPLETA:

GET	POST
Parametri nell'URL (visibili)	Parametri nel corpo HTTP (non visibili)
Limite ~255 caratteri (dipende browser)	Nessun limite pratico di dimensione
Può essere salvato nei bookmark	NON può essere salvato nei bookmark
Rimane nella cronologia browser	NON rimane nella cronologia
Idempotente (ripetibile)	NON idempotente
Per RECUPERO dati (lettura)	Per INVIO dati (scrittura/modifica)
Meno sicuro (password visibile)	Più sicuro (ma serve HTTPS!)

QUANDO USARE GET: Ricerche, filtri, paginazione, link condivisibili, operazioni di sola lettura senza effetti collaterali.

QUANDO USARE POST: Login, registrazioni, upload file (OBBLIGATORIO), form di contatto, acquisti, qualsiasi operazione che modifica dati sul server.

NOTA SICUREZZA: POST è più sicuro di GET ma NON è sicuro senza HTTPS! Senza HTTPS entrambi i metodi trasmettono dati in chiaro intercettabili.

3. Schema organizzativo

DOMANDA: Cosa sono gli schemi organizzativi? Descrivi le tipologie con esempi.

RISPOSTA COMPLETA:

Gli schemi organizzativi permettono di suddividere gli oggetti informativi in raggruppamenti logici. Definiscono COME sono disposte le informazioni nel sito. Si dividono in due categorie:

SCHEMI ESATTI (Exact Organization Schemes)

Ogni elemento appartiene a un'unica classe senza ambiguità. Si basano su conoscenze comuni e oggettive. Adatti quando l'utente sa esattamente cosa cerca.

- ALFABETICO: elementi ordinati A-Z (dizionari, glossari, elenco prodotti)
- CRONOLOGICO: ordinati per data/tempo (archivi news, blog, eventi storici)
- GEOGRAFICO: per località (meteo per città, sedi aziendali, mappe)

SCHEMI AMBIGUI (Ambiguous Organization Schemes)

La classificazione dipende dal contesto e dall'interpretazione. Più soggettivi ma più flessibili. Adatti quando l'utente esplora o non ha idea precisa.

- PER ARGOMENTO/TOPIC: raggruppati per categoria tematica (sito news: Sport, Politica, Tecnologia)
- TASK-ORIENTED: organizzati per compiti/obiettivi utente (sito banca: Apri conto, Richiedi mutuo, Investi)
- METAFORICI: basati su metafore del mondo reale (desktop computer: cartelle, cestino)
- PER AUDIENCE: per tipo di utente (sito aziendale: Area Privati, Area Aziende, Area Pubbliche Amministrazioni)

NOTA: È sempre possibile usare uno schema AMBIGUO per il menu principale (perché si adatta a utenti con diversi livelli di conoscenza), mentre NON è sempre possibile usare uno schema ESATTO (funziona solo se l'utente sa cosa cerca).

4. Struttura organizzativa

DOMANDA: Cosa sono le strutture organizzative? Descrivi le tipologie.

RISPOSTA COMPLETA:

Le strutture organizzative definiscono le RELAZIONI tra le informazioni, cioè COME sono organizzati i collegamenti tra gli elementi. Decidono la navigazione. Tipologie principali:

1) GERARCHIA (Hierarchical Structure)

La più importante e diffusa. Organizzazione ad albero con nodo radice (home) e figli. Relazione padre-figlio.

CARATTERISTICHE: Ampiezza ideale 7 voci (max 10), Profondità ideale 3-4 livelli (max 5, eccezionalmente 7). Meglio AMPIA e POCO PROFONDA per facilitare manutenzione futura e ridurre clic necessari. Ogni clic corrisponde a un livello di profondità.

ESEMPIO: Sito e-commerce → Elettronica → Smartphone → Apple → iPhone 15

2) SEQUENZIALE (Sequential/Linear Structure)

Sequenza lineare di pagine: Pagina 1 → Pagina 2 → Pagina 3. Ordine rigido, un solo percorso possibile.

QUANDO USARE: Tutorial passo-passo, processi di checkout, wizard di configurazione, corsi online con lezioni in sequenza.

ESEMPIO: Acquisto online → Carrello → Dati spedizione → Pagamento → Conferma

3) IPERTESTUALE (Hypertext/Web Structure)

Rete di collegamenti non gerarchici. Ogni pagina può linkare qualsiasi altra. Massima libertà ma rischio di disorientamento.

QUANDO USARE: Wiki, documenti con molti riferimenti incrociati, contenuti altamente interconnessi.

ATTENZIONE: Necessita strumenti di orientamento forti (breadcrumb, mappa del sito) per evitare che l'utente si perda.

4) DATABASE/FACCETTE (Faceted/Database Structure)

Consente ricerca multi-dimensionale con filtri combinabili. L'utente affina progressivamente i risultati.

QUANDO USARE: Cataloghi prodotti, ricerca libri, ricerca immobiliare.

ESEMPIO: E-commerce → Filtra per: Marca (Apple) + Prezzo (500-800€) + Colore (Nero) + Memoria (256GB)

BEST PRACTICE: Prediligere schemi IBRIDI con GERARCHIA alla base, coadiuvata da sequenze per processi specifici e ipertesto per collegamenti contestuali.

5. Emotional Design

DOMANDA: Cosa è l'Emotional Design? Descrivi le emozioni utilizzabili con esempi.

RISPOSTA COMPLETA:

L'Emotional Design è l'approccio progettuale che cerca di creare e disegnare prodotti/servizi per suscitare specifiche EMOZIONI negli utenti, trasmettendo sensazioni positive e creando una buona esperienza. Le emozioni sono importanti perché:

- Hanno forte impatto nella memoria a lungo termine
- Aiutano a trasformare un utente casuale in cliente fedele
- Creano connessione emotiva con il brand
- Facilitano decisioni d'acquisto (l'amigdala rilascia dopamina)

PRINCIPIO BASE: MAI forzare l'utente al cambiamento!

EMOZIONI PIÙ EFFICACI:

1) SORPRESA + PIACERE

Si viene sorpresi quando il cervello trova una deviazione da una situazione standard (CONTRASTO). Due tipi di contrasto: VISUALE (forma, colore, aspetto) e COGNITIVO (rottura esperienze passate, ricordi). La sorpresa è accompagnata da sensazione piacevole e provoca risposta veloce immediata, evitando ragionamento eccessivo. Ottima per acquisti impulsivi.

ATTENZIONE: NON tradire le aspettative dell'utente, altrimenti lo perdi definitivamente.

ESEMPLI: Errore 404 creativo (Pixar), pulsante "Non tirare" (Photojojo), animazioni sorprendenti, messaggi personalizzati inaspettati.

2) ANTICIPAZIONE

Effetto ottenuto mostrando una PARTE del tutto e lasciando correre l'immaginazione. Suscita curiosità e aspettativa.

ATTENZIONE: Crea aspettative sul prodotto → usare SOLO se sicuri di non deluderle. Utilizzare il feedback per migliorare il prodotto.

ESEMPLI: Trailer film, anteprime prodotti (Apple), teaser campagne marketing, countdown lancio prodotto, accesso anticipato utenti selezionati (Twitter redesign).

3) STATUS/ESCLUSIVITÀ

Far sentire l'utente SPECIALE, non uno tra centomila. Senso di appartenenza a gruppo esclusivo.

ESEMPLI: Programmi fedeltà con livelli (Bronze/Silver/Gold), sconti riservati, accesso early a funzionalità, badge/distintivi utenti attivi, lettere personalizzate scritte a mano (Wufoo), inviti esclusivi, contenuti premium per membri.

4) REWARDS/RICOMPENSE

Spingere utenti all'acquisto/utilizzo fornendo una ricompensa per azioni specifiche. Meccanismo slot machine/gamification.

ESEMPLI: Punti fedeltà, cashback, sconti progressivi, badge per azioni completate (Duolingo), spazio storage extra (Dropbox: 5GB bonus università), messaggi di congratulazioni (Mailchimp), premi a sorpresa, barre di progresso completamento profilo.

UMANIZZAZIONE DEL PRODOTTO

Dare caratteristiche umane al prodotto/interfaccia crea empatia.

ESEMPLI: Mascotte (MailChimp), Maggiolino Volkswagen (somiglia a sorriso), emoji e animazioni, messaggi personalizzati, pulsante caricamento con animazioni simpatiche.

NOTA IMPORTANTE: Non tutte le emozioni vanno bene per tutti i contesti. Sorpresa e piacere richiedono attenzione (non adatte a banche/PA), mentre anticipazione, status e rewards si adattano meglio a tutte le personalità di brand.

6. Tre domande fondamentali della navigazione

DOMANDA: Quali sono le tre domande fondamentali alle quali devo saper rispondere in ogni pagina per evitare il disorientamento?

RISPOSTA COMPLETA:

Le tre domande fondamentali sono:

11. 1. DOVE SONO? → L'utente deve sempre sapere la sua posizione nel sito
12. 2. DI COSA SI TRATTA? → L'utente deve capire il contenuto della pagina
13. 3. DOVE POSSO ANDARE? → L'utente deve vedere le opzioni di navigazione

STRUMENTI PER RISPONDERE:

- DOVE SONO: Breadcrumb (percorso), evidenziazione menu attivo, titolo pagina chiaro, URL semantico
- DI COSA SI TRATTA: Titolo H1 descrittivo, introduzione chiara, struttura contenuti logica
- DOVE POSSO ANDARE: Menu navigazione, link correlati, CTA chiari, sitemap

ALTRE 2 DOMANDE SECONDARIE:

4. COME SONO ARRIVATO QUI? → Breadcrumb, link "Torna a..."
5. CHI GESTISCE QUESTA PAGINA? → Footer con informazioni contatto, about

L'eccesso di collegamenti e cammini esplorativi può causare DISORIENTAMENTO e SOVRACCARICO COGNITIVO. Necessarie funzioni di contestualizzazione che diano percezione chiara della posizione.

7. Above the fold

DOMANDA: Cosa si deve mettere "above the fold"?

RISPOSTA COMPLETA:

"Above the fold" indica la parte di pagina IMMEDIATAMENTE visibile senza scroll. Deriva dal gergo giornalistico (metà superiore prima pagina giornale). È la zona più importante perché è la prima cosa che l'utente vede.

ELEMENTI ESSENZIALI:

14. 1. IDENTIFICAZIONE: Logo aziendale, titolo sito
15. 2. NAVIGAZIONE PRINCIPALE: Menu principale (desktop) o hamburger menu (mobile)
16. 3. CONTENUTO PRIMARIO: Informazione più importante della pagina, value proposition
17. 4. CALL-TO-ACTION PRINCIPALE: Pulsante azione principale ben visibile
18. 5. RICERCA: Casella di ricerca (se sito grande)

OBIETTIVO: L'utente deve capire IMMEDIATAMENTE di cosa si tratta il sito e cosa può fare, senza dover scrollare.

NOTA: La dicitura "mettere tutto above the fold" non è realistica. Mettere solo l'essenziale. Il resto può stare below the fold purché la pagina inviti chiaramente allo scroll.

8. Responsive Web Design

DOMANDA: Cosa si intende per Responsive Web?

RISPOSTA COMPLETA:

Il Responsive Web Design (detto anche layout a variabilità controllata) è un approccio alla progettazione web che rende le pagine ADATTABILI a qualsiasi dispositivo e dimensione schermo, fornendo un'esperienza utente ottimale su desktop, tablet e smartphone.

PRINCIPI FONDAMENTALI:

19. 1. LAYOUT FLUIDO: Uso di unità relative (% , em , rem , vw/vh) anziché pixel fissi
20. 2. MEDIA QUERIES: Punti di rottura/breakpoint per adattare lo stile
21. 3. IMMAGINI FLESSIBILI: max-width: 100% per adattare alle dimensioni
22. 4. MOBILE FIRST: Progettare prima per mobile, poi espandere per schermi grandi

BREAKPOINT STANDARD:

- 320px: Smartphone piccoli (portrait)
- 480px: Smartphone (landscape)
- 600px: Tablet piccoli (portrait)
- 768px: Tablet (portrait, iPad)
- 1024px: Tablet (landscape), laptop piccoli
- 1200px: Desktop, schermi grandi

OBIETTIVO: Il contenuto deve essere sempre LEGGIBILE e NAVIGABILE indipendentemente dal dispositivo, garantendo esperienza fluida senza zoom orizzontale o contenuti troncati.

META TAG OBBLIGATORIO:

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

PARTE 5: TABELLA ACCESSIBILE

Elementi essenziali per accessibilità

- 23. 1. <caption>: Titolo/descrizione della tabella
- 24. 2. <thead>, <tbody>, <tfoot>: Struttura semantica
- 25. 3. <th> con scope: Intestazioni con scope="row" o scope="col"
- 26. 4. headers (alternativa a scope): Per tabelle complesse
- 27. 5. abbr: Abbreviazioni per intestazioni lunghe (screen reader)
- 28. 6. summary (XHTML): Descrizione tabella (deprecato HTML5)
- 29. 7. aria-describedby (HTML5): Alternativa moderna a summary
- 30. 8. lang: Quando celle contengono testo in lingua diversa
- 31. 9. Alternanza colori: Se tabella molto lunga (tr:nth-child)

ESEMPIO COMPLETO: Menù settimanale (XHTML)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it" lang="it">
<head>
  <title>Menù Settimanale</title>
  <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
</head>
<body>
  <table summary="Menù della mensa scolastica dal lunedì al venerdì">
    <caption>Menù Settimanale della Mensa</caption>
    <thead>
      <tr>
        <th scope="col" abbr="Giorno">Giorno della settimana</th>
        <th scope="col" abbr="Primo">Primo piatto</th>
        <th scope="col" abbr="Secondo">Secondo piatto</th>
        <th scope="col" abbr="Contorno">Contorno</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <th scope="row" abbr="Lun">Lunedì</th>
        <td>Pasta al pomodoro</td>
        <td>Pollo arrosto</td>
        <td>Insalata mista</td>
      </tr>
      <tr>
        <th scope="row" abbr="Mar">Martedì</th>
        <td>Risotto ai funghi</td>
        <td>Pesce al forno</td>
        <td>Patate lesse</td>
      </tr>
      <tr>
        <th scope="row" abbr="Mer">Mercoledì</th>
        <td>Lasagne al ragù</td>
        <td>Cotoletta</td>
        <td>Verdure grigliate</td>
      </tr>
```

```
<tr>
  <th scope="row" abbr="Gio">Giovedì</th>
  <td>Minestrone</td>
  <td>Hamburger</td>
  <td>Carote</td>
</tr>
<tr>
  <th scope="row" abbr="Ven">Venerdì</th>
  <td>Pasta al pesto</td>
  <td>Pizza margherita</td>
  <td>Pomodori</td>
</tr>
</tbody>
</table>
</body>
</html>
```

ESEMPIO COMPLETO: Menù settimanale (HTML5)

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Menù Settimanale</title>
</head>
<body>
  <p id="descrizione-menu">Menù della mensa scolastica dal lunedì al
venerdì,
    con primo piatto, secondo piatto e contorno per ogni giorno.</p>

  <table aria-describedby="descrizione-menu">
    <caption>Menù Settimanale della Mensa</caption>
    <thead>
      <tr>
        <th scope="col" abbr="Giorno">Giorno della settimana</th>
        <th scope="col" abbr="Primo">Primo piatto</th>
        <th scope="col" abbr="Secondo">Secondo piatto</th>
        <th scope="col" abbr="Contorno">Contorno</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <th scope="row" abbr="Lun">Lunedì</th>
        <td>Pasta al pomodoro</td>
        <td>Pollo arrosto</td>
        <td>Insalata mista</td>
      </tr>
      <tr>
        <th scope="row" abbr="Mar">Martedì</th>
        <td>Risotto ai funghi</td>
        <td>Pesce al forno</td>
        <td>Patate lesse</td>
      </tr>
      <tr>
        <th scope="row" abbr="Mer">Mercoledì</th>
        <td>Lasagne al ragù</td>
```

```
<td>Cotoletta</td>
<td>Verdure grigliate</td>
</tr>
<tr>
  <th scope="row" abbr="Gio">Giovedì</th>
  <td>Minestrone</td>
  <td>Hamburger</td>
  <td>Carote</td>
</tr>
<tr>
  <th scope="row" abbr="Ven">Venerdì</th>
  <td>Pasta al pesto</td>
  <td>Pizza margherita</td>
  <td>Pomodori</td>
</tr>
</tbody>
</table>
</body>
</html>
```

DIFFERENZE XHTML vs HTML5 per tabelle

XHTML 1.0 Strict	HTML5
<code>summary="descrizione" su <table></code>	<code>aria-describedby="id" + <p id="..."></code>
<code>xml:lang="it" lang="it" su <html></code>	Solo <code>lang="it"</code> su <code><html></code>
Chiusura obbligatoria tutti i tag	Alcuni tag auto-chiudenti permessi

ATTENZIONE ESAME: Specificare se richiesto XHTML o HTML5! Usare la sintassi corretta. Il codice deve essere VALIDO: aprire e chiudere tutti i tag correttamente!

NOTE AGGIUNTIVE

- `scope="col"`: per intestazioni di COLONNA
- `scope="row"`: per intestazioni di RIGA
- `abbr`: versione abbreviata letta da screen reader (velocizza lettura)
- `headers + id`: per tabelle COMPLESSE con intestazioni multiple
- Evitare celle vuote: inserire sempre contenuto o ` `
- Non annidare tabelle dentro tabelle
- Non usare tabelle per layout (solo per dati tabulari!)

PARTE 6: TEORIA MINIMA INDISPENSABILE

HTML/XHTML - Tag semantici essenziali

- <header>: Intestazione pagina/sezione (logo, nav)
- <nav>: Menu navigazione principale
- <main>: Contenuto principale (UNO solo per pagina)
- <article>: Contenuto autonomo distribuibile
- <section>: Sezione tematica con intestazione
- <aside>: Contenuto correlato/sidebar
- <footer>: Piè di pagina (copyright, contatti)
- <h1>...<h6>: Intestazioni gerarchiche (UNA H1 per pagina)

CSS - Selettori base

- *: Universale (tutti gli elementi)
- elemento: Selettore di tipo (p, div, span)
- .classe: Selettore di classe
- #id: Selettore di ID
- [attr]: Selettore di attributo
- :hover, :focus: Pseudo-classi
- ::before, ::after: Pseudo-elementi

Accessibilità - Principi PURO (WCAG 2.1)

- P - PERCEPIBILE: Informazioni presentabili a utenti in modi percepibili
- U - UTILIZZABILE (Operable): Componenti interfaccia e navigazione utilizzabili
- R - ROBUSTO: Contenuti interpretabili da ampia varietà user agent
- O - COMPRENSIBILE (Understandable): Informazioni e uso interfaccia comprensibili

LIVELLI WCAG:

- A: Livello base minimo
- AA: Livello richiesto per LEGGE (Legge Stanca, European Accessibility Act 2025)
- AAA: Livello avanzato (non obbligatorio)

SEO - Tag fondamentali

- <title>: 55-60 caratteri, particolare→generale, unico per pagina, keywords all'inizio
- <meta name="description">: 145-160 caratteri, CTA, keywords, influenza CTR non ranking
- <h1>: UNA per pagina, contiene keyword principale
- <h2>...<h6>: Gerarchia logica, keywords secondarie
- alt immagini: Descrittivo, keywords naturali, alt="" per immagini decorative
- URL semantici: /prodotti/smartphone/iphone-15 non /page?id=12345

PHP - Solo GET/POST

\$_GET: Parametri visibili nell'URL, limite dimensione, per recupero dati (lettura), salvabile nei bookmark

\$_POST: Parametri nel corpo HTTP, nessun limite, per invio dati (scrittura/modifica),
OBBLIGATORIO per upload file

APPENDICE: CHECKLIST RAPIDA PRE-ESAME

DA RICORDARE ASSOLUTAMENTE

- Test accessibilità automatici: FALSO che siano esaustivi (solo 30%)
- Struttura gerarchica: AMPIA e POCO PROFONDA (7 voci, max 10; 3-4 livelli, max 5)
- Convenzioni interne: NON rompere MAI
- Convenzioni esterne: Si possono rompere se giustificato
- Link visitati: Colore + ALTRO indicatore (non solo colore)
- Bandiere per lingue: FALSO (bandiere = paese, non lingua)
- Tabelle: Usare SOLO per dati tabulari, MAI per layout
- WCAG livelli: AA obbligatorio per legge (NON AAA)
- Menu voci: 7 ideale, max 10
- Screen reader: Necessario ma NON sufficiente
- Validazione codice: Necessaria ma NON sufficiente
- Layout a schede: SCONSIGLIATO (poco flessibile)
- Metatag description: Influenza CTR, NON ranking
- Tag title: Particolare → Generale (55-60 char)
- Separazione contenuto/presentazione: SEMPRE

METAFORA PESCA (sintesi)

32. 1. Tiro perfetto: sa esattamente cosa cerca
33. 2. Trappola aragoste: idea precisa, vuole imparare
34. 3. Pesca rete: esplora tutto
35. 4. Boa segnalazione: vuole ritrovare

EMOTIONAL DESIGN (sintesi)

36. 1. Sorpresa + Piacere: contrasto, non tradire aspettative
37. 2. Anticipazione: mostra parte, crea aspettative
38. 3. Status/Esclusività: far sentire speciale
39. 4. Rewards: ricompense per azioni

SPECIFICITÀ CSS (formula rapida)

(INLINE, ID, CLASSI/ATTRIBUTI, ELEMENTI)

- Inline: 1000 punti
- ID: 100 punti
- Classe/Attributo: 10 punti
- Elemento: 1 punto

TABELLA ACCESSIBILE (elementi obbligatori)

- caption, thead/tbody/tfoot, th con scope, abbr, summary (XHTML) o aria-describedby (HTML5)

TRE DOMANDE NAVIGAZIONE

- 1. Dove sono? 2. Di cosa si tratta? 3. Dove posso andare?