

# Personalizzazione di Elementor con HTML e Responsive Design

## Prefazione

Questa guida ti aiuterà a personalizzare il tuo sito WordPress con Elementor utilizzando HTML e a rendere il tuo sito responsive per tutti i dispositivi. È pensata per chi si sta introducendo alla materia con l'HTML e desidera personalizzare / ottimizzare ulteriormente il proprio sito.

La struttura modulare di interazione tra Wordpress – Html e temi (come ELEMENTOR), consente di organizzare in modo veloce e dinamico non solo lo stile ma anche le interazioni del sito web che si va a generare. In più, mantenendo sempre tutte le proprietà di interazione e di gestione creativa collaborativa che consente di portare avanti Wordpress.

## Indice

### Sommario

Personalizzazione di Elementor con HTML e Responsive Design .....	1
Introduzione.....	2
Basi di HTML in WordPress .....	2
Design Responsive: Concetti Base .....	4
Esempi Pratici.....	8
Consigli Utili e Problemi Comuni .....	12
Esercitazioni con Casi Risolti.....	14
Concetti Base di UX Design Applicati al Design Responsive.....	35

## Introduzione

### Basi di HTML in WordPress

Prima di personalizzare Elementor, è importante comprendere come e dove puoi inserire HTML in WordPress:

#### Widget HTML

WordPress offre un widget "HTML personalizzato" che puoi aggiungere alle aree widget del tuo tema:

1. Vai su **Aspetto → Widget**
2. Trascina il widget "HTML personalizzato" in un'area widget
3. Inserisci il tuo codice HTML e salva

#### Editor di blocchi (Gutenberg)

Nell'editor di WordPress puoi utilizzare il blocco "HTML personalizzato":

1. Nell'editor, clicca sul pulsante "+" per aggiungere un blocco
2. Cerca e seleziona "HTML personalizzato"
3. Inserisci il tuo codice HTML nel blocco

#### Inserimento di codice nei file del tema

**Nota:** Questo metodo è più avanzato e richiede cautela:

1. Vai su **Aspetto → Editor di temi**
2. Seleziona il file che desideri modificare (ad esempio, header.php o footer.php)
3. Aggiungi il tuo codice HTML e salva

**Importante:** Prima di modificare i file del tema, crea sempre un tema figlio per evitare di perdere le modifiche durante gli aggiornamenti.

#### Personalizzare Elementor con HTML

Elementor offre diverse opzioni per inserire codice HTML personalizzato:

#### Widget HTML di Elementor

1. Nell'editor di Elementor, cerca il widget "HTML" nella barra laterale
2. Trascinalo nell'area di editing

## Corso in presenza . (4) Personalizzazione di Wordpress e Responsive & UX Design

3. Inserisci il tuo codice HTML nel campo di testo
4. Clicca su "Aggiorna" per salvare le modifiche

Esempio di codice base che puoi inserire:

### html

```
<div class="contenitore-personalizzato">  
    <h2>Il mio titolo personalizzato</h2>  
    <p>Questo è un paragrafo con <span style="color: red;">testo in rosso</span>.</p>  
    <a href="https://miosito.it/pagina" class="bottone-custom">Clicca qui</a>  
</div>
```

### HTML avanzato: Widget "Shortcode"

Puoi creare shortcode personalizzati con HTML e utilizzarli in Elementor:

1. Aggiungi questo codice in functions.php del tuo tema figlio:

### php

```
function html_personalizzato_shortcode() {  
    return '<div class="contenitore-personalizzato">  
        <h2>Il mio titolo personalizzato</h2>  
        <p>Questo è un paragrafo con <span style="color: red;">testo in rosso</span>.</p>  
    </div>';  
}  
  
add_shortcode('mio_html', 'html_personalizzato_shortcode');
```

2. Nell'editor di Elementor, aggiungi un widget "Shortcode"
3. Inserisci [mio\_html] nel campo shortcode

### Personalizzazione CSS Elementor

Per aggiungere stili CSS personalizzati ai tuoi elementi HTML:

1. Nell'editor di Elementor, clicca sull'icona  (Impostazioni) in basso a sinistra
2. Vai alla scheda "Avanzate"
3. Inserisci il tuo CSS personalizzato nel campo "CSS personalizzato"

Esempio di CSS personalizzato:

**css**

```
.contenitore-personalizzato {  
    background-color: #f5f5f5;  
    padding: 20px;  
    border-radius: 10px;  
    margin-bottom: 20px;  
}  
  
.bottone-custom {  
    display: inline-block;  
    padding: 10px 20px;  
    background-color: #0073aa;  
    color: white;  
    text-decoration: none;  
    border-radius: 5px;  
    transition: all 0.3s ease;  
}  
  
.bottone-custom:hover {  
    background-color: #005177;  
}
```

## Design Responsive: Concetti Base

Il design responsive permette al tuo sito di adattarsi automaticamente a diverse dimensioni di schermo. Consiste nell'implementazione di aggiornamenti in tempo reale alla distribuzione dei contenuti, tenendo in considerazione la possibilità di utilizzo da qualsiasi dispositivo e con particolare riguardo ai servizi mobile (piccoli dispositivi, layout preparato per la lettura in orizzontale e/o tramite tablet di diverse dimensioni)

### Punti di interruzione (Breakpoints)

I breakpoints sono le larghezze di schermo in cui il layout cambia:

- **Desktop:** > 1024px

- **Tablet:** 768px - 1024px
- **Mobile:** < 768px

Elementor utilizza questi breakpoints standard, ma puoi personalizzarli in **Elementor → Impostazioni → Responsive.**

### Viewport Meta Tag

Assicurati che questo tag sia presente nell'header del tuo sito:

#### html

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

La maggior parte dei temi WordPress moderni lo include già.

### Implementare il Responsive Design in Elementor

Elementor offre funzionalità integrate per il design responsive:

#### Controlli responsive nell'editor

1. Nell'editor di Elementor, nota i pulsanti per dispositivi (desktop, tablet, mobile) nell'angolo in basso a sinistra
2. Cliccando su questi pulsanti, puoi visualizzare in anteprima come apparirà il tuo sito su diversi dispositivi
3. Le impostazioni che modifichi mentre sei in una specifica visualizzazione si applicano solo a quel dispositivo

#### Impostazioni responsive per colonne e sezioni

1. Seleziona una sezione o colonna
2. Vai alla scheda "Layout" nelle impostazioni
3. Puoi modificare larghezza, margini, padding per ogni dispositivo

#### Nascondere elementi su dispositivi specifici

1. Seleziona qualsiasi elemento
2. Vai alla scheda "Avanzate" → "Responsive"
3. Attiva/disattiva la visibilità per desktop, tablet o mobile

## Utilizzo di Media Queries

Le media queries sono comandi CSS che applicano stili solo quando specifiche condizioni sono soddisfatte (come la larghezza dello schermo).

### Inserire Media Queries in Elementor

Puoi aggiungere media queries personalizzate nelle impostazioni CSS personalizzate:

1. Nell'editor di Elementor, vai su  → "Avanzate" → "CSS personalizzato"
2. Inserisci le tue media queries

Ecco alcuni esempi di media queries:

#### css

```
/* Stili per dispositivi mobili (fino a 767px) */  
 @media only screen and (max-width: 767px) {  
   .contenitore-personalizzato {  
     padding: 10px;  
     margin-bottom: 10px;  
   }  
   .contenitore-personalizzato h2 {  
     font-size: 20px;  
   }  
 }  
  
/* Stili per tablet (da 768px a 1024px) */  
 @media only screen and (min-width: 768px) and (max-width: 1024px) {  
   .contenitore-personalizzato {  
     padding: 15px;  
   }  
   .contenitore-personalizzato h2 {  
     font-size: 24px;  
   }  
 }
```

```
/* Stili per desktop (più di 1024px) */

@media only screen and (min-width: 1025px) {

    .contenitore-personalizzato {

        padding: 30px;

        max-width: 1200px;

        margin: 0 auto 30px;

    }

}
```

### Media queries con Widget HTML

Puoi anche inserire media queries direttamente nel widget HTML, usando il tag <style>:

#### html

```
<div class="mia-sezione-personalizzata">

    <h2>Il mio titolo</h2>

    <p>Il mio contenuto</p>

</div>

<style>

.mia-sezione-personalizzata {

    background-color: #f5f5f5;

    padding: 20px;

    margin-bottom: 20px;

}

@media only screen and (max-width: 767px) {

    .mia-sezione-personalizzata {

        padding: 10px;

        background-color: #e0e0e0;

    }

    .mia-sezione-personalizzata h2 {

        font-size: 18px;

    }

}

</style>
```

## Esempi Pratici

Vediamo alcuni esempi pratici di personalizzazioni responsive con HTML e CSS:

### Esempio 1: Contenuto in due colonne responsive

html

```
<div class="colonne-flex">

    <div class="colonna">
        <h3>Prima colonna</h3>
        <p>Contenuto della prima colonna</p>
    </div>

    <div class="colonna">
        <h3>Seconda colonna</h3>
        <p>Contenuto della seconda colonna</p>
    </div>
</div>

<style>
.colonne-flex {
    display: flex;
    gap: 20px;
}

.colonna {
    flex: 1;
    padding: 20px;
    background-color: #f5f5f5;
}

@media only screen and (max-width: 767px) {
    .colonne-flex {
        flex-direction: column;
    }
}
```

```
.colonna {  
    margin-bottom: 15px;  
}  
}  
</style>
```

### Esempio 2: Menu personalizzato responsive

html

```
<nav class="menu-personalizzato">  
    <button class="toggle-menu">Menu</button>  
    <ul class="menu-items">  
        <li><a href="#">Home</a></li>  
        <li><a href="#">Chi siamo</a></li>  
        <li><a href="#">Servizi</a></li>  
        <li><a href="#">Contatti</a></li>  
    </ul>  
</nav>
```

```
<style>  
.menu-personalizzato {  
    background-color: #333;  
    padding: 10px;  
}
```

```
.menu-items {  
    display: flex;  
    list-style: none;  
    margin: 0;  
    padding: 0;  
}
```

```
.menu-items li {  
    margin-right: 20px;  
}  
  
.menu-items a {  
    color: white;  
    text-decoration: none;  
}  
  
.toggle-menu {  
    display: none;  
    background-color: #444;  
    color: white;  
    border: none;  
    padding: 10px;  
    cursor: pointer;  
}  
  
@media only screen and (max-width: 767px) {  
    .toggle-menu {  
        display: block;  
    }  
  
    .menu-items {  
        display: none;  
        flex-direction: column;  
        margin-top: 10px;  
    }  
  
    .menu-items.show {  
        display: flex;  
    }  
}
```

```
.menu-items li {  
    margin-right: 0;  
    margin-bottom: 10px;  
}  
}  
</style>  
  
<script>  
document.querySelector('.toggle-menu').addEventListener('click', function() {  
    document.querySelector('.menu-items').classList.toggle('show');  
});  
</script>
```

### Esempio 3: Griglia di immagini responsive

html

```
<div class="griglia-immagini">  
    <div class="immagine"></div>  
    <div class="immagine"></div>  
    <div class="immagine"></div>  
    <div class="immagine"></div>  
    <div class="immagine"></div>  
    <div class="immagine"></div>  
</div>  
  
<style>  
.griglia-immagini {  
    display: grid;
```

```
grid-template-columns: repeat(3, 1fr);
gap: 20px;
}

.immagine img{
width: 100%;
height: auto;
display: block;
}

@media only screen and (max-width: 1024px) {
.griglia-immagini {
grid-template-columns: repeat(2, 1fr);
}
}

@media only screen and (max-width: 767px) {
.griglia-immagini {
grid-template-columns: 1fr;
}
}

</style>
```

## Consigli Utili e Problemi Comuni

### Consigli per il Design Responsive

1. **Mobile-first:** Progetta prima per dispositivi mobili, poi espandi per schermi più grandi
2. **Test frequenti:** Testa regolarmente il tuo sito su diversi dispositivi
3. **Immagini ottimizzate:** Usa immagini responsive con max-width: 100%; height: auto;
4. **Font leggibili:** Usa font size di almeno 16px per dispositivi mobili
5. **Spazio sufficiente:** Aumenta padding (distanziamento dai bordi) e spazio tra elementi sui dispositivi mobili per facilitare il tocco

### Problemi Comuni e Soluzioni

#### Problema: Elementi che escono dallo schermo su mobile

Soluzione:

css

```
.elemento-problematico {  
    max-width: 100%;  
    overflow-x: hidden;  
}
```

#### Problema: Testo troppo piccolo su mobile

Soluzione:

css

```
@media only screen and (max-width: 767px) {  
    body, p, li {  
        font-size: 16px;  
    }  
    h1 {  
        font-size: 24px;  
    }  
    h2 {  
        font-size: 20px;  
    }  
}
```

#### Problema: Immagini troppo grandi su mobile

Soluzione:

css

```
img {  
    max-width: 100%;  
    height: auto;  
}
```

**Problema: Menu che scompare su mobile**

**Soluzione:** Implementa un menu hamburger come nell'esempio 2 sopra.

**Strumenti per il test responsive**

- **Chrome DevTools:** Premi F12 e usa il toggle device toolbar (Ctrl+Shift+M)
- **Responsinator:** <https://www.responsinator.com/>
- **BrowserStack:** <https://www.browserstack.com/> (servizio a pagamento)
- **Elementor:** Usa i controlli responsive integrati nell'editor

**Ottimizzazione delle prestazioni per mobile**

1. **Comprimi immagini:** Usa plugin come "Smush" o "ShortPixel"
2. **Caricamento lazy:** Aggiungi loading="lazy" alle immagini
3. **Minimizza CSS e JavaScript:** Usa plugin per la minificazione
4. **Cache del browser:** Implementa cache efficaci

---

## Esercitazioni con Casi Risolti

### Esercizio 1: Personalizzare un titolo con HTML e CSS in WordPress

**Problema:** Volete creare un titolo con un testo evidenziato e uno sfondo colorato.

**Soluzione:**

1. Nell'editor di WordPress, aggiungi un blocco "HTML personalizzato"
2. Inserisci questo codice:

**html**

```
<h2 class="titolo-personalizzato">Il mio progetto <span  
class="evidenziato">creativo</span></h2>
```

```
<style>  
.titolo-personalizzato {  
    font-family: 'Arial', sans-serif;  
    font-size: 32px;  
    padding: 15px;  
    background-color: #f0f0f0;  
    border-left: 5px solid #0073aa;
```

```
}

.evidenziato {

background-color: #0073aa;

color: white;

padding: 0 10px;

}

</style>
```

**Risultato:** Un titolo elegante con la parola "creativo" evidenziata su sfondo blu e l'intero titolo con uno sfondo grigio chiaro e una linea blu a sinistra.

### Esercizio 2: Creare un Blocco HTML Personalizzato in Elementor

**Problema:** Volete creare un box informativo con un'icona, un titolo e un testo, il tutto personalizzato.

**Soluzione:**

1. Nell'editor di Elementor, trascina un widget "HTML" nella tua pagina
2. Inserisci questo codice:

**html**

```
<div class="info-box">

<div class="icon">

<svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"
fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-
linejoin="round">

<circle cx="12" cy="12" r="10"></circle>

<line x1="12" y1="16" x2="12" y2="12"></line>

<line x1="12" y1="8" x2="12.01" y2="8"></line>

</svg>

</div>

<div class="content">

<h3>Informazione Importante</h3>

<p>Questo è un messaggio importante che voglio evidenziare nel mio sito web.</p>

</div>

</div>
```

```
<style>

.info-box {
    display: flex;
    background-color: #e1f5fe;
    border-left: 4px solid #03a9f4;
    padding: 20px;
    margin-bottom: 20px;
    border-radius: 4px;
}

.icon {
    color: #03a9f4;
    margin-right: 15px;
    flex-shrink: 0;
}

.content h3 {
    margin-top: 0;
    color: #0288d1;
}

.content p {
    margin-bottom: 0;
}
</style>
```

**Risultato:** Un elegante box informativo con icona, stile moderno e colori coordinati.

### Esercizio 3: Rendere Responsive un Layout a Due Colonne

**Problema:** Avete bisogno di visualizzare due blocchi di contenuto affiancati su desktop, ma impilati su mobile.

**Soluzione:**

1. Nell'editor di Elementor, trascina un widget "HTML" nella tua pagina
2. Inserisci questo codice:

**html**

```
<div class="flex-container">

    <div class="flex-box">
        <h3>Servizio #1</h3>
        <p>Descrizione dettagliata del nostro primo servizio. Questo testo spiega tutti i vantaggi che otterrai.</p>
        <a href="#" class="box-link">Scopri di più</a>
    </div>

    <div class="flex-box">
        <h3>Servizio #2</h3>
        <p>Descrizione dettagliata del nostro secondo servizio. Questo testo spiega tutti i vantaggi che otterrai.</p>
        <a href="#" class="box-link">Scopri di più</a>
    </div>
</div>

<style>
.flex-container {
    display: flex;
    gap: 20px;
    margin-bottom: 30px;
}

.flex-box {
    flex: 1;
    background-color: white;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
    padding: 25px;
    border-radius: 8px;
}
```

```
}

.flex-box h3{
    color: #333;
    margin-top: 0;
}

.box-link {
    display: inline-block;
    margin-top: 15px;
    padding: 8px 15px;
    background-color: #4CAF50;
    color: white;
    text-decoration: none;
    border-radius: 4px;
    font-weight: bold;
}

@media only screen and (max-width: 767px) {
    .flex-container {
        flex-direction: column;
    }

    .flex-box {
        margin-bottom: 20px;
    }
}
```

**Risultato:** Due box affiancati su desktop che si trasformano in box impilati verticalmente su dispositivi mobili.

#### Esercizio 4: Creare un'Intestazione Responsive con Media Queries

**Problema:** Volete una intestazione con logo a sinistra e titolo a destra su desktop, ma centrati su mobile.

**Soluzione:**

1. Nell'editor di Elementor, trascina un widget "HTML" nella tua pagina
2. Inserisci questo codice:

**html**

```
<header class="intestazione-custom">

    <div class="logo">
        
    </div>

    <div class="titolo-sito">
        <h1>Nome della Mia Azienda</h1>
        <p>Slogan o breve descrizione dell'azienda</p>
    </div>
</header>

<style>
.intestazione-custom {
    display: flex;
    align-items: center;
    padding: 20px;
    background-color: #f9f9f9;
    border-bottom: 1px solid #e0e0e0;
}

.logo {
    margin-right: 30px;
}

.logo img {
    display: block;
}
```

```
max-width: 100%;  
height: auto;  
}  
  
.titolo-sito h1 {  
margin: 0 0 5px 0;  
font-size: 28px;  
color: #333;  
}  
  
.titolo-sito p {  
margin: 0;  
color: #666;  
}  
  
@media only screen and (max-width: 767px) {  
.intestazione-custom {  
flex-direction: column;  
text-align: center;  
}  
  
.logo {  
margin-right: 0;  
margin-bottom: 15px;  
}  
  
.titolo-sito h1 {  
font-size: 24px;  
}  
}  
}  
</style>
```

**Risultato:** Un'intestazione con logo a sinistra e titolo a destra su desktop, che si trasforma in layout verticale centrato su mobile.

### Esercizio 5: Creare un Carosello di Testimonianze Responsive

**Problema:** Volete mostrare tre testimonianze affiancate su desktop, due su tablet e una sola su mobile.

**Soluzione:**

1. Nell'editor di Elementor, trascina un widget "HTML" nella tua pagina
2. Inserisci questo codice:

**html**

```
<div class="testimonianze-grid">

    <div class="testimonianza">
        <div class="testimonianza-contenuto">
            <p>"Questo servizio ha superato ogni mia aspettativa. Altamente consigliato!"</p>
        </div>
        <div class="testimonianza-autore">
            
            <div class="autore-info">
                <h4>Mario Rossi</h4>
                <p>CEO, Azienda XYZ</p>
            </div>
        </div>
    </div>

    <div class="testimonianza">
        <div class="testimonianza-contenuto">
            <p>"Professionalità impeccabile e risultati straordinari. Continuerò a collaborare!"</p>
        </div>
        <div class="testimonianza-autore">
            
            <div class="autore-info">
                <h4>Laura Bianchi</h4>
```

## Corso in presenza . (4) Personalizzazione di Wordpress e Responsive & UX Design

```
<p>Marketing Manager, Azienda ABC</p>
</div>
</div>
</div>

<div class="testimonianza">
    <div class="testimonianza-contenuto">
        <p>"Il miglior servizio nel settore. Tempi rapidi e qualità superiore!"</p>
    </div>
    <div class="testimonianza-autore">
        
        <div class="autore-info">
            <h4>Giovanni Verdi</h4>
            <p>Direttore Tecnico, Azienda DEF</p>
        </div>
    </div>
</div>

<style>
.testimonianze-grid {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 20px;
    margin-bottom: 30px;
}

.testimonianza {
    background-color: white;
    border-radius: 8px;
    box-shadow: 0 3px 10px rgba(0,0,0,0.1);
    overflow: hidden;
}
```

}

```
.testimonianza-contenuto {  
    padding: 25px;  
    background-color: #f8f9fa;  
    font-style: italic;  
    position: relative;  
}
```

```
.testimonianza-contenuto:after {  
    content: "";  
    position: absolute;  
    bottom: -10px;  
    left: 30px;  
    border-width: 10px 10px 0;  
    border-style: solid;  
    border-color: #f8f9fa transparent;  
}
```

```
.testimonianza-autore {  
    padding: 15px 25px;  
    display: flex;  
    align-items: center;  
}
```

```
.testimonianza-autore img {  
    width: 60px;  
    height: 60px;  
    border-radius: 50%;  
    margin-right: 15px;  
}
```

```
.autore-info h4 {  
    margin: 0 0 5px 0;  
    font-size: 18px;  
}  
  
.autore-info p {  
    margin: 0;  
    color: #666;  
    font-size: 14px;  
}  
  
@media only screen and (max-width: 1024px) and (min-width: 768px) {  
    .testimonianze-grid {  
        grid-template-columns: repeat(2, 1fr);  
    }  
}  
  
@media only screen and (max-width: 767px) {  
    .testimonianze-grid {  
        grid-template-columns: 1fr;  
    }  
}  
}
```

**Risultato:** Un grid di testimonianze che mostra tre box su desktop, due su tablet e uno su mobile, con un design professionale e responsive.

#### Esercizio 6: Implementare un Pulsante di "Torna su" Responsive

**Problema:** Volete aggiungere un pulsante "Torna su" che appare quando l'utente scorre verso il basso, ma che sia posizionato diversamente su desktop e mobile.

**Soluzione:**

1. Nell'editor di Elementor, trascina un widget "HTML" nella tua pagina (preferibilmente alla fine)
2. Inserisci questo codice:

**html**

```
<button id="torna-su" title="Torna su">↑</button>

<style>
#torna-su {
    display: none;
    position: fixed;
    bottom: 30px;
    right: 30px;
    width: 50px;
    height: 50px;
    background-color: #333;
    color: white;
    border: none;
    border-radius: 50%;
    font-size: 24px;
    cursor: pointer;
    box-shadow: 0 2px 10px rgba(0,0,0,0.3);
    transition: background-color 0.3s;
    z-index: 999;
}

#torna-su:hover {
    background-color: #555;
}

@media only screen and (max-width: 767px) {
    #torna-su {
        bottom: 20px;
    }
}
```

```
        right: 20px;  
        width: 40px;  
        height: 40px;  
        font-size: 20px;  
    }  
}  
</style>  
  
<script>  
  
// Mostra/nasconde il pulsante in base allo scroll  
  
window.addEventListener('scroll', function() {  
    var tornaSuBtn = document.getElementById('torna-su');  
    if (window.pageYOffset > 300) {  
        tornaSuBtn.style.display = 'block';  
    } else {  
        tornaSuBtn.style.display = 'none';  
    }  
});  
  
// Funzione per tornare in cima alla pagina  
  
document.getElementById('torna-su').addEventListener('click', function() {  
    window.scrollTo({  
        top: 0,  
        behavior: 'smooth'  
    });  
});  
</script>
```

**Risultato:** Un pulsante "Torna su" che appare quando l'utente scorre la pagina per più di 300px, con dimensioni diverse per desktop e mobile.

### Esercizio 7: Creare un Form di Contatto Responsive

**Problema:** Volete implementare un form di contatto personalizzato che sia facile da usare sia su desktop che su mobile.

**Soluzione:**

1. Nell'editor di Elementor, trascina un widget "HTML" nella tua pagina
2. Inserisci questo codice:

**html**

```
<div class="form-container">

    <h2>Contattaci</h2>

    <form id="contact-form" class="custom-form">

        <div class="form-row">
            <div class="form-group">
                <label for="nome">Nome</label>
                <input type="text" id="nome" name="nome" required>
            </div>
            <div class="form-group">
                <label for="cognome">Cognome</label>
                <input type="text" id="cognome" name="cognome" required>
            </div>
        </div>
        <div class="form-group">
            <label for="email">Email</label>
            <input type="email" id="email" name="email" required>
        </div>
        <div class="form-group">
            <label for="oggetto">Oggetto</label>
            <input type="text" id="oggetto" name="oggetto" required>
        </div>
        <div class="form-group">
            <label for="messaggio">Messaggio</label>
            <textarea id="messaggio" name="messaggio" rows="5" required></textarea>
        </div>
    </form>
</div>
```

```
</div>

<div class="form-group">
    <button type="submit" class="submit-btn">Invia Messaggio</button>
</div>
</form>
</div>

<style>
.form-container {
    max-width: 800px;
    margin: 0 auto;
    padding: 30px;
    background-color: #f9f9f9;
    border-radius: 8px;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.form-container h2 {
    margin-top: 0;
    margin-bottom: 20px;
    text-align: center;
    color: #333;
}

.custom-form {
    display: flex;
    flex-direction: column;
    gap: 20px;
}

.form-row {
    display: flex;
```

```
gap: 20px;  
}  
  
.form-group {  
    flex: 1;  
    display: flex;  
    flex-direction: column;  
}  
  
.form-group label {  
    margin-bottom: 8px;  
    font-weight: bold;  
    color: #555;  
}  
  
.form-group input,  
.form-group textarea {  
    padding: 12px 15px;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    font-size: 16px;  
}  
  
.form-group textarea {  
    resize: vertical;  
}  
  
.submit-btn {  
    background-color: #4CAF50;  
    color: white;  
    border: none;  
    padding: 12px 20px;
```

```
font-size: 16px;  
font-weight: bold;  
border-radius: 4px;  
cursor: pointer;  
transition: background-color 0.3s;  
}  
  
.submit-btn:hover {  
background-color: #45a049;  
}  
  
@media only screen and (max-width: 767px) {  
.form-container {  
padding: 20px;  
}  
  
.form-row {  
flex-direction: column;  
gap: 20px;  
}  
  
.submit-btn {  
width: 100%;  
}  
}  
  
</style>
```

**Risultato:** Un form di contatto responsive con nome e cognome affiancati su desktop ma impilati su mobile, con stile moderno e professionale.

#### Esercizio 8: Creare un Menu a Schede Responsive

**Problema:** Volete creare un menu a schede che funzioni bene sia su desktop che su dispositivi mobili.

**Soluzione:**

1. Nell'editor di Elementor, trascina un widget "HTML" nella tua pagina
2. Inserisci questo codice:

**html**

```
<div class="tabs-container">

<div class="tabs">

    <button class="tab-btn active" data-tab="tab1">Servizi</button>

    <button class="tab-btn" data-tab="tab2">Progetti</button>

    <button class="tab-btn" data-tab="tab3">Testimonianze</button>

    <button class="tab-btn" data-tab="tab4">Chi Siamo</button>

</div>

<div class="tab-content active" id="tab1">

    <h3>I Nostri Servizi</h3>

    <p>Offriamo una vasta gamma di servizi professionali per soddisfare le vostre esigenze. Dalla consulenza strategica allo sviluppo web, siamo qui per aiutarvi a raggiungere i vostri obiettivi aziendali.</p>

</div>

<div class="tab-content" id="tab2">

    <h3>I Nostri Progetti</h3>

    <p>Esplora i nostri progetti recenti che dimostrano la nostra esperienza e capacità. Abbiamo collaborato con aziende di vari settori per fornire soluzioni innovative e di alta qualità.</p>

</div>

<div class="tab-content" id="tab3">

    <h3>Testimonianze dei Clienti</h3>

    <p>Leggi cosa dicono i nostri clienti soddisfatti. La loro fiducia e i loro feedback positivi sono la prova del nostro impegno verso l'eccellenza e la soddisfazione del cliente.</p>

</div>

<div class="tab-content" id="tab4">
```

```
<h3>Chi Siamo</h3>

<p>Siamo un team di professionisti dedicati con anni di esperienza nel settore. La nostra missione è fornire soluzioni di alta qualità e superare le aspettative dei nostri clienti.</p>

</div>
</div>

<style>

.tabs-container {
    max-width: 100%;
    margin-bottom: 30px;
    border: 1px solid #ddd;
    border-radius: 8px;
    overflow: hidden;
}

.tabs {
    display: flex;
    background-color: #f0f0f0;
    border-bottom: 1px solid #ddd;
}

.tab-btn {
    flex: 1;
    padding: 15px;
    border: none;
    background: none;
    font-size: 16px;
    font-weight: bold;
    cursor: pointer;
    transition: background-color 0.3s;
}

</style>
```

```
.tab-btn:hover {  
    background-color: #e0e0e0;  
}  
  
.tab-btn.active {  
    background-color: white;  
    border-bottom: 3px solid #4CAF50;  
}  
  
.tab-content {  
    display: none;  
    padding: 25px;  
    background-color: white;  
}  
  
.tab-content.active {  
    display: block;  
}  
  
@media only screen and (max-width: 767px) {  
    .tabs {  
        flex-direction: column;  
    }  
  
.tab-btn {  
    border-bottom: 1px solid #ddd;  
    text-align: left;  
    padding: 12px 15px;  
}  
  
.tab-btn.active {  
    border-bottom: 1px solid #ddd;
```

```
border-left: 3px solid #4CAF50;  
}  
  
.tab-content {  
    padding: 20px 15px;  
}  
}  
</style>  
  
<script>  
  
// Funzione per cambiare scheda  
document.querySelectorAll('.tab-btn').forEach(button => {  
    button.addEventListener('click', function() {  
        // Rimuovi active da tutti i pulsanti e contenuti  
        document.querySelectorAll('.tab-btn').forEach(btn => btn.classList.remove('active'));  
        document.querySelectorAll('.tab-content').forEach(content =>  
            content.classList.remove('active'));  
  
// Aggiungi active al pulsante cliccato  
        this.classList.add('active');  
  
// Mostra il contenuto corrispondente  
        const tabId = this.getAttribute('data-tab');  
        document.getElementById(tabId).classList.add('active');  
    });  
});  
</script>
```

**Risultato:** Un menu a schede orizzontale su desktop che diventa verticale su mobile, con stile adattato per ogni dispositivo.

## Concetti Base di UX Design Applicati al Design Responsive

Il design responsivo non riguarda solo l'adattamento tecnico di un sito web a diverse dimensioni di schermo, ma anche l'ottimizzazione dell'esperienza utente (UX) su tutti i dispositivi. Alla base consiste nel capire e definire quale sarà il percorso che seguirà l'utente di un App e, di conseguenza, pianificare e sviluppare ogni elemento inserito perché risulti intuitivo e facile da esperimentare.

Ecco alcuni concetti fondamentali di UX Design da considerare quando si crea un sito responsive:

### 1. Principio "Mobile First"

**Concetto:** Progettare prima l'interfaccia per dispositivi mobili e poi estenderla per schermi più grandi.

**Vantaggi:**

- Costringe a concentrarsi sui contenuti essenziali
- Migliora i tempi di caricamento su dispositivi mobili
- Semplifica la progressiva aggiunta di funzionalità per schermi più grandi

**Come applicarlo con Elementor:**

- Utilizza l'editor in modalità mobile per iniziare il design
- Identifica e posiziona prima i contenuti più importanti
- Aggiungi complessità solo quando passi a schermi più grandi

### 2. Gerarchia Visiva e Priorità dei Contenuti

**Concetto:** Organizzare gli elementi visivi in ordine di importanza per guidare l'attenzione dell'utente.

**Applicazione nel design responsive:**

- Su mobile: disporre gli elementi in una sequenza verticale chiara di priorità
- Su desktop: utilizzare dimensioni, colori e posizionamento per creare focus visivo
- Mantenere coerenza nella gerarchia tra i diversi dispositivi

**Esempio pratico:**

**html**

```
<style>
/* Desktop */
.hero-section {
    display: flex;
```

```
    align-items: center;
}

.hero-content {
  flex: 1;
  padding-right: 30px;
}

.hero-image {
  flex: 1;
}

/* Mobile */

@media only screen and (max-width: 767px) {

  .hero-section {
    flex-direction: column;
  }

  .hero-content {
    order: 1; /* Sposta il contenuto sopra all'immagine su mobile */
    padding-right: 0;
    margin-bottom: 20px;
  }

  .hero-image {
    order: 2;
  }
}

</style>
```

### 3. Accessibilità e Usabilità

#### Principi chiave:

- **Contrasto:** Garantire un contrasto adeguato tra testo e sfondo (rapporto minimo 4.5:1)
- **Dimensione dei tocchi:** Elementi interattivi di almeno 44×44 pixel su mobile
- **Spaziatura:** Aumentare lo spazio tra elementi cliccabili su dispositivi touch

**Come implementarlo in Elementor:**

html

```
<style>

/* Stile base per i pulsanti */

.button {
    padding: 12px 24px;
    font-size: 16px;
    background-color: #0073aa;
    color: white;
    border-radius: 4px;
    display: inline-block;
}

/* Aumenta dimensione su mobile per facilitare il tocco */

@media only screen and (max-width: 767px) {

    .button {
        padding: 16px 24px;
        font-size: 18px;
        margin: 10px 0; /* Maggiore spaziatura verticale */
        width: 100%; /* Pulsante a tutta larghezza */
    }
}

</style>
```

**4. Navigazione Adattiva**

**Concetto:** Trasformare il menu di navigazione per adattarlo a diversi dispositivi mantenendo l'usabilità.

**Tipi di navigazione responsive:**

- **Menu hamburger:** Ideale per mobile, nasconde il menu dietro un'icona
- **Menu a tab:** Adatto per pochi elementi di navigazione
- **Menu a priorità:** Mostra gli elementi più importanti e nasconde gli altri

**Esempio di implementazione:**

html

```
<nav class="main-nav">

    <div class="logo">Brand</div>

    <button class="menu-toggle">☰</button>

    <ul class="menu-items">

        <li><a href="#">Home</a></li>
        <li><a href="#">Servizi</a></li>
        <li><a href="#">Chi siamo</a></li>
        <li><a href="#">Contatti</a></li>
    </ul>

</nav>

<style>

.main-nav {

    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 15px;
}

.menu-items {

    display: flex;
    list-style: none;
    margin: 0;
    padding: 0;
}

```

```
.menu-items li {  
    margin-left: 20px;  
}  
  
.menu-toggle {  
    display: none;  
}  
  
@media only screen and (max-width: 767px) {  
    .menu-items {  
        display: none;  
        position: absolute;  
        top: 60px;  
        left: 0;  
        right: 0;  
        background: white;  
        flex-direction: column;  
        box-shadow: 0 2px 5px rgba(0,0,0,0.1);  
    }  
  
    .menu-items.active {  
        display: flex;  
    }  
  
    .menu-items li {  
        margin: 0;  
        text-align: center;  
    }  
  
    .menu-items a {  
        display: block;  
    }  
}
```

```
padding: 15px;  
border-bottom: 1px solid #eee;  
}  
  
.menu-toggle {  
display: block;  
background: none;  
border: none;  
font-size: 24px;  
cursor: pointer;  
}  
}  
</style>  
  
<script>  
document.querySelector('.menu-toggle').addEventListener('click', function() {  
    document.querySelector('.menu-items').classList.toggle('active');  
});  
</script>
```

## 5. Modelli di Interazione Specifici per Dispositivo

**Concetto:** Adattare le interazioni al dispositivo utilizzato dall'utente.

### Esempi:

- **Desktop:** Hover, drag and drop, interazioni complesse
- **Mobile:** Swipe, tap, gesture multi-touch

### Implementazione in CSS:

#### css

```
/* Effetti hover solo su desktop */  
 @media (hover: hover) {  
     .card:hover {
```

```
        transform: translateY(-5px);  
        box-shadow: 0 10px 20px rgba(0,0,0,0.1);  
    }  
}  
  
/* Interazioni touch specifiche */  
  
@media (hover: none){  
    .swipe-container {  
        overflow-x: auto;  
        -webkit-overflow-scrolling: touch;  
        scroll-snap-type: x mandatory;  
    }  
  
    .swipe-item {  
        scroll-snap-align: start;  
    }  
}
```

## 6. Performance e Velocità di Caricamento

**Importanza UX:** La velocità di caricamento è un fattore critico per la user experience, specialmente su mobile.

### Strategie per migliorare le performance in un design responsive:

- **Immagini responsive:** Usare srcset per caricare immagini ottimizzate
- **Lazy loading:** Caricare le immagini solo quando entrano nel viewport
- **Minimizzare il codice:** Eliminare CSS e JS non necessari

### Esempio di immagini responsive:

html

```

```

### *7. Coerenza dell'Esperienza su Tutti i Dispositivi*

**Concetto:** Mantenere un'esperienza riconoscibile e coerente indipendentemente dal dispositivo.

#### **Strategie:**

- Usare la stessa paletta di colori, font e stile visivo
- Mantenere una terminologia consistente
- Assicurarsi che tutte le funzionalità principali siano disponibili su tutti i dispositivi

#### **Come implementarlo:**

- Usa variabili CSS per mantenere colori e stili coerenti
- Crea un sistema di componenti riutilizzabili
- Testa regolarmente su diversi dispositivi

#### **css**

```
:root {  
  
    --color-primary: #0073aa;  
  
    --color-secondary: #23282d;  
  
    --font-heading: 'Roboto', sans-serif;  
  
    --font-body: 'Open Sans', sans-serif;  
  
    --spacing-unit: 8px;  
  
}
```

*/\* Utilizzo coerente in tutto il CSS \*/*

```
.button-primary {  
  
    background-color: var(--color-primary);  
  
    font-family: var(--font-body);
```

```
padding: calc(var(--spacing-unit) * 2) calc(var(--spacing-unit) * 3);  
}
```

### 8. Feedback e Microinterazioni

**Concetto:** Fornire riscontri visivi alle azioni dell'utente per migliorare l'usabilità.

#### Esempi di microinterazioni responsive:

- Animazioni di pulsanti
- Feedback visivi durante il caricamento
- Transizioni tra stati dell'interfaccia

#### Implementazione semplice:

##### css

```
/* Feedback per pulsanti */  
  
.button {  
    transition: transform 0.2s, background-color 0.2s;  
}  
  
.button:active {  
    transform: scale(0.98);  
}  
  
/* Versione desktop */  
  
@media (hover: hover){  
    .button:hover {  
        background-color: #005d8c;  
    }  
}  
  
/* Animazione di caricamento responsive */  
  
.loading-spinner {  
    border: 4px solid rgba(0, 0, 0, 0.1);
```

```
border-top: 4px solid var(--color-primary);  
border-radius: 50%;  
width: 30px;  
height: 30px;  
animation: spin 1s linear infinite;  
}  
  
@media only screen and (max-width: 767px) {  
.loading-spinner {  
width: 20px;  
height: 20px;  
}  
}  
  
@keyframes spin {  
0% { transform: rotate(0deg); }  
100% { transform: rotate(360deg); }  
}
```

## 9. Testing UX su Diversi Dispositivi

### Approccio metodico al testing UX responsive:

1. **Test su dispositivi reali:** Non limitarsi agli emulatori
2. **Test con utenti reali:** Osservare come le persone interagiscono con il sito
3. **Heatmap e analisi del comportamento:** Utilizzare strumenti come Hotjar
4. **Analisi delle metriche:** Monitorare tassi di conversione e rimbalzo per dispositivo

### Strumenti utili:

- **Responsive Mode di Chrome DevTools**
- **BrowserStack** per testare su diversi dispositivi
- **Google PageSpeed Insights** per analisi performance

#### 10. Contenuti Prioritari e Design "Content-First"

**Concetto:** Progettare partendo dai contenuti più importanti per l'utente piuttosto che dal layout.

#### Strategie:

- Identificare contenuti essenziali vs. contenuti secondari
- Posizionare i contenuti prioritari "above the fold" su mobile
- Utilizzare la tecnica del "progressive disclosure" per contenuti complessi

#### Esempio di implementazione:

##### html

```
<div class="card-container">

    <div class="card">

        <!-- Contenuto prioritario sempre visibile -->

        <div class="card-essential">
            <h3>Titolo della card</h3>
            <p class="card-summary">Riassunto delle informazioni principali.</p>
        </div>

        <!-- Contenuto secondario espandibile -->
        <div class="card-details">
            <p>Dettagli aggiuntivi che vengono mostrati solo su richiesta dell'utente.</p>
            <ul>
                <li>Punto 1</li>
                <li>Punto 2</li>
            </ul>
        </div>

        <button class="toggle-details">Leggi di più</button>
    </div>
</div>

<style>
```

```
.card {  
    padding: 20px;  
    border-radius: 8px;  
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
    margin-bottom: 20px;  
}  
  
.card-details {  
    display: none;  
    margin-top: 15px;  
    padding-top: 15px;  
    border-top: 1px solid #eee;  
}  
  
.card-details.active {  
    display: block;  
}  
  
@media only screen and (min-width: 1024px) {  
    /* Su desktop possiamo mostrare più contenuti direttamente */  
    .card-details {  
        display: block;  
    }  
  
.toggle-details {  
    display: none;  
}  
}  
}  
</style>  
  
<script>  
document.querySelector('.toggle-details').addEventListener('click', function() {
```

```
const details = this.previousElementSibling;  
details.classList.toggle('active');  
this.textContent = details.classList.contains('active') ? 'Mostra meno' : 'Leggi di più';  
});  
</script>
```

Applicando questi principi di UX Design al tuo sito responsive, non solo lo renderai visivamente adattabile a diversi dispositivi, ma creerai anche un'esperienza utente efficace e soddisfacente indipendentemente dal dispositivo utilizzato.

L'obiettivo finale, rimane sempre che gli utenti possano interagire con il tuo sito in modo naturale e senza ostacoli, sia che utilizzino uno smartphone, un tablet o un computer desktop.

---

Questa guida ti ha fornito le basi per personalizzare Elementor con HTML e creare un design responsive, insieme a diversi esercizi pratici per mettere in pratica queste conoscenze e concetti di UX Design per migliorare l'esperienza utente.

Ricorda di testare sempre il tuo sito su diversi dispositivi e di apportare modifiche graduali, che consentono di testare ogni singolo scenario apportando le modifiche specifiche in modo diretto.

Buon lavoro!

Ferrara, Aprile 2025

Ulises Gabriel Miranda

Siti web di riferimento:

<https://ievolutiongroup.com>  
<https://ulises.it>

Sito web di collaborazione e proposte:

<https://iegroup.org>