
Delivery of €13 million in submarine cables for the Shanhaiguan offshore p

Data: 2025-09-28 10:08:08

Autor: Inteligência Against Invaders

```
document.addEventListener('DOMContentLoaded', function() {
const buttons = document.querySelectorAll('.wpcode-tts-button');
if (!buttons.length || !('speechSynthesis' in window)) return;

const ajaxUrl = 'https://www.redhotcyber.com/wp-admin/admin-ajax.php';

function recordAudioView(postId) {
fetch(ajaxUrl, {
method: 'POST',
headers: {
'Content-Type': 'application/x-www-form-urlencoded'
},
body: 'action=wpcode_tts_view&post_id=' + encodeURIComponent(postId)
})
.then(response => response.json())
.then(data => {
if (!data.success) {
console.error('Errore logico AJAX:', data.data);
}
})
.catch(error => console.error('Errore di rete AJAX:', error));
}

let voices = [];
let voicesLoaded = false;

function loadVoices() {
voices = speechSynthesis.getVoices();
if (voices.length > 0) {
voicesLoaded = true;
}
}

// Tentativo di caricamento iniziale
loadVoices();

// Assicurati che le voci vengano ricaricate quando l'elenco cambia (evento cruciale)
if (speechSynthesis.onvoiceschanged !== undefined) {
speechSynthesis.onvoiceschanged = loadVoices;
}
```

```

}

/**
 * Funzione principale per avviare la sintesi vocale, con logica di retry.
 * @param {HTMLElement} button L'elemento del pulsante.
 * @param {number} attempt Contatore dei tentativi (max 2).
 */
function startSpeaking(button, attempt = 1) {
  const text = button.getAttribute('data-article');
  const lang = button.getAttribute('data-lang') || 'it-IT';
  const pref = button.getAttribute('data-voice-pref') || '';
  const rate = parseFloat(button.getAttribute('data-rate')) || 1;
  const pitch = parseFloat(button.getAttribute('data-pitch')) || 1;

  if (!text) { console.error('Testo mancante o vuoto.');
```

}

```

  // *** LOGICA DI RETRY PER IL CARICAMENTO DELLE VOCI ***
  if (!voicesLoaded && attempt === 1) {
    loadVoices(); // Riprova a caricare
    if (!voicesLoaded) {
      console.warn("Voci non caricate al 1° tentativo. Riprovo in 500ms.");
      button.textContent = "? Caricamento vocale... (Riprovo)";

      // Riprova dopo 500ms
      setTimeout(() => startSpeaking(button, 2), 500);
      return;
    }
  } else if (!voicesLoaded && attempt === 2) {
    console.error("Voci non caricate al 2° tentativo. L'API TTS non è disponibile.");
    button.textContent = "? API vocale non disponibile ?";
    setTimeout(() => {
      if (button.getAttribute('data-speaking') !== 'true') {
        button.textContent = "? Ascolta l'articolo";
      }
    }, 3000);
    return;
  }
  // *** FINE LOGICA DI RETRY ***

  // Cancella eventuali letture precedenti e setta lo stato
  speechSynthesis.cancel();
  document.querySelectorAll('.wpcode-tts-button').forEach(b => {
    if (b !== button) {
      b.setAttribute('data-speaking', 'false');
      b.textContent = "? Ascolta l'articolo";
    }
  });

  const utter = new SpeechSynthesisUtterance(text);
  utter.lang = lang;

```

```
utter.rate = rate;
utter.pitch = pitch;

// Selezione voce migliorata
let selectedVoice = null;
if (voices.length) {
  const italianVoices = voices.filter(v => v.lang.startsWith("it"));

  // Priorità 1: Voci di alta qualità
  if (pref === 'female') {
    selectedVoice = italianVoices.find(v => v.name.includes("Google") && /(it-IT|Italian)/i.test(v.lang) &&
    /(female|femmina|alessia|lucia|carla)/i.test(v.name)) ||
    italianVoices.find(v => v.name.includes("Microsoft") && /(female|femmina)/i.test(v.name));
  } else if (pref === 'male') {
    selectedVoice = italianVoices.find(v => v.name.includes("Google") && /(it-IT|Italian)/i.test(v.lang) &&
    /(male|maschio|luca|paolo)/i.test(v.name)) ||
    italianVoices.find(v => v.name.includes("Microsoft") && /(male|maschio)/i.test(v.name));
  }

  // Priorità 2 & 3: Fallback
  if (!selectedVoice) {
    if (pref === 'female') selectedVoice = italianVoices.find(v =>
    /(female|femmina|alessia|lucia|carla)/i.test(v.name));
    else if (pref === 'male') selectedVoice = italianVoices.find(v =>
    /(male|maschio|luca|paolo|giorgio)/i.test(v.name));
  }
  if (!selectedVoice && italianVoices.length) selectedVoice = italianVoices[0];
}
if (selectedVoice) utter.voice = selectedVoice;

// Handlers
utter.onstart = function(){ button.setAttribute('data-speaking', 'true'); button.textContent="?? Ferma
la lettura"; };
utter.onend = function(){ button.setAttribute('data-speaking', 'false'); button.textContent="? Ascolta
l'articolo"; };
utter.onerror = function(e){
  console.error('TTS error – Codice:', e.error, 'dettaglio:', e);
  button.setAttribute('data-speaking', 'false');
  //button.textContent="? Errore di lettura ?";
  setTimeout(() => {
    if (button.getAttribute('data-speaking') !== 'true') {
      button.textContent="? Ascolta l'articolo";
    }
  }, 3000);
};

speechSynthesis.speak(utter);
button.setAttribute('data-speaking', 'true'); // Setta lo stato qui per evitare click multipli
}
```

```
buttons.forEach(function(button) {
button.addEventListener('click', function() {
const speaking = button.getAttribute('data-speaking') === 'true';

// Se stiamo parlando, ferma e resetta. Non contare il click di stop.
if (speaking) {
speechSynthesis.cancel();
button.setAttribute('data-speaking', 'false');
button.textContent=""? Ascolta l'articolo";
return;
}

// *** LOGICA DI CONTEGGIO E AVVIO ***

// 1. Registra la vista (se non stiamo già parlando)
const postId = button.getAttribute('data-post-id');
if (postId) {
recordAudioView(postId);
}

// 2. Avvia la sintesi vocale con logica di retry
startSpeaking(button);

// *****

});
});
});

const lazyloadRunObserver = () => {
const lazyloadBackgrounds = document.querySelectorAll( `.e-con.e-parent:not(.e-lazyloaded)` );
const lazyloadBackgroundObserver = new IntersectionObserver( ( entries ) => {
entries.forEach( ( entry ) => {
if ( entry.isIntersecting ) {
let lazyloadBackground = entry.target;
if( lazyloadBackground ) {
lazyloadBackground.classList.add( 'e-lazyloaded' );
}
lazyloadBackgroundObserver.unobserve( entry.target );
}
});
}, { rootMargin: '200px 0px 200px 0px' } );
lazyloadBackgrounds.forEach( ( lazyloadBackground ) => {
lazyloadBackgroundObserver.observe( lazyloadBackground );
} );
});
const events = [
'DOMContentLoaded',
'elementor/lazyload/observe',
];
events.forEach( ( event ) => {
```

```
document.addEventListener( event, lazyloadRunObserver );  
});
```