Sommario:

1. [Tecnologie Usate 2](#_bookmark0)
   1. [Linguaggi usati 2](#_bookmark1)
   2. [Codice 2](#_bookmark2)
2. [Documentazione Codice 2](#_bookmark3)
   1. [script.js 2](#_bookmark4)
      1. [Funzioni 2](#_bookmark5)
   2. [style.css 3](#_bookmark6)
      1. [Funzioni 3](#_bookmark7)
   3. [html 8](#_bookmark8)
      1. [Funzioni 8](#_bookmark9)

# Tecnologie Usate

## Linguaggi usati

L’applicativo IL CORPO UMANO è scritto utilizzando un linguaggio markup quale HTML5

(HyperText Markup Language) associato a CSS3 (sigla di Cascading Style Sheets, in italiano fogli di stile a cascata). Inoltre, è stato utilizzato JavaScript (è un linguaggio di scripting orientato agli oggetti e agli eventi).

## Codice

Il codice sorgente dell’applicativo, oltre ai file HTML e CSS creati per strutturare e curare l’aspetto grafico, è composto da file JS che gestiscono gli elementi dinamici dell’applicativo. Il file in questione è: script.js, che contiene tutte le funzioni che gestiscono la dinamicità delle pagine HTML dell’applicazione, come apertura e chiusura del menù e delle pagine mappa e chi siamo.

1.3   
abbiamo studiato architettura del percorso di apprendimento che gli utenti entrando seguiranno  
E’ stato nec per dare piu’ qualità a

# Documentazione Codice

Verranno esposte le varie funzioni seguendo l’ordine nei file.

## script.js

### Funzioni

#### openModal()

Funzione utilizzata per aprire una finestra modale specifica.

#### closeModal()

Funzione per chiudere la finestra modale attualmente aperta.

#### toggleMenu()

Funzione per mostrare o nascondere il menù di navigazione.

#### navigateTo(page)

Funzione per caricare dinamicamente il contenuto della pagina specificata senza ricaricare l’intera pagina.

## style.css

### Funzioni

#### Reset e Impostazioni Base

* *\* { margin: 0; padding: 0; box-sizing: border-box; }*

– Imposta margini e padding a zero e definisce il box-sizing per tutti gli elementi.

#### Impostazioni del Body

* *body { min-height: 100vh; display: flex; }*

– Garantisce che il body occupi almeno l’altezza della viewport e lo trasforma in un flex container.

#### Splash Screen

* + - * 1. ***Contenitore Splash Screen***

*.splash-screen { position: fixed; width: 100%; height: 100vh; background: linear-gradient(to bottom, #4a90e2, #0052cc); color: white; display: flex; flex-direction: column; align-items: center; justify-content: space-around; text-align: center; z-index: 1200; padding: 30px; }*

– Posiziona il contenitore a schermo intero, con background a gradiente, testo bianco e layout centrato.

#### Titolo Splash

*.title { display: flex; align-items: center; justify-content: center; font-size: 4rem; font-family: myFont; }*

– Centra il titolo e imposta dimensione e font personalizzato.

#### Container Animato

*.container { position: relative; width: 100%; max-width: 1200px; height: 100%; display: flex; justify-content: center; align-items: center; opacity: 0; transform: scale(0.8); transition: opacity 1s ease, transform 1s ease; }*

* Definisce un contenitore centrato, inizialmente invisibile e scalato, con transizione per opacità e trasformazione.

*.container.visible { opacity: 1; transform: scale(1); }*

* Rende visibile il container, riportando la scala a 1.

#### Gestione delle Figure

*.figure { position: absolute; width: 15vw; height: 85%; transition: transform 1s ease; }*

* Posiziona in modo assoluto le immagini, definendone dimensioni e transizione per la trasformazione.

*.figure img { width: 100%; height: 100%; object-fit: contain; }*

* Adatta l’immagine all’elemento mantenendo le proporzioni.

*.expanded .figure:nth-child(1) { transform: translateX(-250%); }*

*.expanded .figure:nth-child(2) { transform: translateX(-150%); }*

*.expanded .figure:nth-child(3) { transform: translateX(-50%); }*

*.expanded .figure:nth-child(4) { transform: translateX(50%); }*

*.expanded .figure:nth-child(5) { transform: translateX(150%); }*

*.expanded .figure:nth-child(6) { transform: translateX(250%); }*

* Applica differenti trasformazioni orizzontali in base alla posizione della figura, per animare l’effetto "expanded".

#### Contenitore Bottoni Splash

*.splash-button-container { margin-top: 20px; opacity: 0; transition: opacity 1s ease; }*

* Imposta il margine superiore e l’opacità iniziale con transizione.

*.splash-button-container.visible { opacity: 1; }*

* Rende visibile il contenitore dei bottoni.

#### Bottoni

*.button { padding: 15px 50px; font-size: 1.5rem; background: #fff; color: #4a90e2; border: none; border-radius: 5px; cursor: pointer; transition: background 0.3s ease; }*

* Definisce lo stile del bottone (spaziatura, dimensioni, colori, bordi arrotondati e transizione).

*.button:hover { background: #d3d1d1; }*

* Modifica il background al passaggio del mouse.

#### Layout Principale

* + - * 1. ***Container Layout***

*.layout { display: flex; width: 100vw; height: 100vh; display: none; }*

– Imposta un layout flessibile, con dimensioni pari alla viewport, nascosto inizialmente (sarà mostrato dopo il click su “Entra”).

#### Contenuto Principale

*.main-content { flex-grow: 1; border: none; height: 100vh; width: 100%; background-color: #f4f4f4; }*

– Definisce l’area principale dell’applicazione, espandendosi nello spazio disponibile e con background chiaro.

#### Sidebar e Menu

* + - * 1. ***Sidebar***

*.side-bar { width: 350px; height: 100vh; background: linear-gradient(to bottom, #4a90e2, #0052cc); display: flex; flex-direction: column; overflow-y: scroll; }*

– Crea una sidebar con larghezza fissa, background a gradiente e scroll verticale.

#### Contenitore Menu

*.menu-container { display: flex; flex-direction: column; height: 100%; }*

– Organizza gli elementi del menu in colonna.

#### Personalizzazione della Scrollbar

*.side-bar::-webkit-scrollbar { height: 3px; width: 5px; }*

*.side-bar::-webkit-scrollbar-thumb { background: linear-gradient(45deg, rgba(56,56,56,0.7), rgba(56,56,56,0.7)); border-radius: 10px; }*

*.side-bar::-webkit-scrollbar-track { background: rgba(0,0,0,0.4); }*

– Modifica l’aspetto della scrollbar della sidebar.

#### Titolo del Menu

*.menu-title { text-align: center; font-size: 22px; font-weight: bold; color: white; padding: 20px 0; text-transform: uppercase; }*

– Stila il titolo del menu con allineamento centrale, dimensioni, colore e trasformazione in maiuscolo.

#### Menu Principale

*.menu { flex-grow: 1; }*

* Garantisce che il menu occupi lo spazio disponibile, spingendo il menu in basso.

*.menu .item { position: relative; cursor: pointer; }*

* Imposta la posizione relativa e il cursore pointer per gli elementi del menu.

*.menu .item a { display: block; color: white; text-decoration: none; font-size: 18px; padding: 15px 25px; transition: background 0.3s ease; }*

* Definisce lo stile dei link del menu, inclusi spaziatura, colore e transizione.

*.menu .item a:hover { background: rgba(255,255,255,0.2); }*

* Applica un effetto hover ai link del menu.

#### Submenu

*.sub-menu { display: none; background: #003ea3; }*

* Imposta il submenu inizialmente nascosto e con background specifico.

*.sub-menu a { text-align: center; }*

* Centra il testo dei link all’interno del submenu.

#### Icona Dropdown

*.menu .item .dropdown { position: absolute; right: 20px; transition: transform 0.3s ease; }*

* Posiziona l’icona per il dropdown e ne definisce la transizione.

*.rotate { transform: rotate(90deg); }*

* Ruota l’icona per indicare l’espansione del submenu.

#### Menu in Basso

*.bottom-menu { display: flex; flex-direction: column; margin-top: 5vh; }*

* Organizza il menu inferiore in colonna con margine superiore.

*.bottom-menu .item a { text-align: center; padding: 15px 25px; display: block; color: white; font-size: 18px; }*

* *Stila i link del menu in basso.*

*.bottom-menu .item a:hover { background: rgba(255,255,255,0.2); }*

* Applica un effetto hover ai link del menu in basso.

#### Bottone di Uscita

*.exit-btn { font-weight: bold; background: rgba(255,0,0,0.3); }*

* Definisce lo stile del bottone di uscita, con enfasi sul colore.

*.exit-btn:hover { background: rgba(255,0,0,0.5); }*

* Modifica il background al passaggio del mouse per un effetto hover.

## html

### Funzioni

* + - 1. *<meta name="viewport" content="width=device-width, initial-scale=1.0">:*

– Imposta la larghezza del viewport del dispositivo e il livello di zoom iniziale per la visualizzazione su dispositivi mobili.

* + - 1. *<link rel="stylesheet" type="text/css" href="tipidipinguino.css">:*

– Collega un file CSS esterno per applicare stili alla pagina.

* + - 1. *<div class="container">:*
         * Crea un contenitore div con la classe "container" per raggruppare elementi all'interno della pagina.
      2. *<h1><center>RAZZE&PINGUINI</center></h1>:*
         * Inserisce un'intestazione di primo livello (H1) con il testo centrato.
      3. *<div class="container" id="home">:*
         * Crea un altro contenitore div con la classe "container" e l'ID "home".
      4. *<div class="text-container">:*
         * Crea un contenitore div per il testo descrittivo sulla pagina.
      5. *<p>:*
         * Inserisce paragrafi di testo.
      6. *<img>:*
         * Inserisce immagini.
      7. *<div class="menu-overlay" id="menuOverlay">:*
         * Crea un menu sovrapposto con l'ID "menuOverlay".
      8. *<a>:*
         * Crea collegamenti ipertestuali a diverse pagine del sito, ciascuno con un'icona e un testo descrittivo.
      9. *<a href="homepage.html" class="avanti-icon"> e <a href="tipi\_di\_pinguino2.html" class="indietro-icon">:*
         * Creano collegamenti per avanzare e tornare indietro tra le pagine, ciascuno con un'icona di freccia.
      10. *<script src="java.js"></script>:*
          * Collega un file JavaScript esterno chiamato "java.js" per l'esecuzione di script aggiuntivi sulla pagina.

***Riconoscimenti:***

Applicazione a cura di:

© 2025 **Alessandro Trianni, Vincenzo Grottola, Marco Rossetto e Daniele Girardi**

Tutti i diritti riservati.