



CERCA PAROLE

UN PROGETTO DI :

-LEONARDO GRANDOLFO

-ENORE LASI

-ALESSANDRO DI ROMA

-FRANCESCO CARNEVALI

COSA ABBIAMO USATO:

- Linguaggio PHP
- Javascript
- CSS

LA LOGICA DIETRO :

- Siamo partiti da una semplice text-box che analizzasse frasi lunghe prese da input per poi mostrare le 10 parole più usate al suo interno, per potere sviluppare la logica di programmazione.
- Abbiamo creato una funzione sanitize
- Abbiamo utilizzato explode per poter splittare le parole e metterle all'interno di un'array, e abbiamo utilizzato replace per sostituire tutti i segni di punteggiatura e i caratteri speciali con uno spazio per agevolare lo split.

```
9 function sanitize($str){
10     $content = read();
11
12     // RIMUOVO GLI SPAZI BIANCHI E SEPARO TUTTE LE PAROLE CON LO SPAZIO, METTENDOLE POI IN UN ARRAY
13     $replace = array_map("trim", explode(" ", $content));
14
15     // SOSTITUISCO TUTTI I SEGNI DI PUNTEGGIATURA E I CARATTERI SPECIALI CON UNO SPAZIO NELLA STRINGA DATA
16     $str = preg_replace("/[.,\/#!$%\"'\"^&\*';\":{}=\-_~()]/", ' ', $str);
17
18     // SEPARO TUTTE LE PAROLE DEL TESTO DATO CON UNO SPAZIO CREANDO UN ARRAY
19     $words = explode(" ", strtolower($str));
20
21     $return = $words;
```

LA LOGICA DIETRO :

- Abbiamo messo in dei file di testo le 'stop words' cioè le parole escluse nell'algoritmo di calcolo delle parole.

```
3 // FILE NEL QUALE SI TROVANO LE PAROLE DA SCARTARE
4 define("WORDSFILE", "files/stopwords_it.txt");
```

- Inizialmente settato a 10 abbiamo deciso di mostrare un numero di 20 parole più usate.
- Abbiamo utilizzato un foreach per potere scartare tutte le parole nel file di testo contenuto nell'array

```
23 // SE L'ARRAY PRECEDENTEMENTE CREATO CONTIENE UNA DELLE PAROLE DA SCARTARE, TOLGO QUELLE PAROLE DALL'ARRAY
24 foreach($words as $k=>$v){
25     if(in_array($v, $replace) || strlen($v) == 1 || is_numeric($v) || ctype_space($v)){
26         unset($return[$k]);
27     }
28 }
29
30 return array_filter($return);
31 }
```

LA LOGICA DIETRO :

- Abbiamo creato una funzione find words, che prende l'input e ci utilizza la funzione sanitize per potere ottenere un array senza caratteri speciali
- Settiamo una variabile per le occorrenze sul totale e creiamo un'array per le occorrenze
- Facciamo un foreach prendendo una variabile v per potere cercare eventuali parole uguali nel vettore occorrenze, se vengono trovate aumenterà il valore all'interno dell'array occurrences, (per vedere se v era presente nell'array abbiamo utilizzato la funzione array_key_exists)

```
37 function findwords($str, $url){
38     $occurrences = array();
39
40     // PRENDO LE PAROLE "VALIDE" DALL'INPUT DELL'UTENTE
41     $words = sanitize($str);
42
43     // TOTALE DELLE OCCORRENZE (100%)
44     $tot = 0;
45
46     // SE L'ARRAY $occurrences CONTIENE UNA DELLE PAROLE, ALLORA AUMENTO IL SUO VALORE, ALTRIMENTI AGGIUNGO LA PAROLA ALL'ARRAY
47     foreach($words as $v){
48         if(array_key_exists($v, $occurrences))
49             $occurrences[$v] += 1;
50         else
51             $occurrences[$v] = 1;
52
53         $tot++;
54     }
```

LA LOGICA DIETRO :

- Facciamo un `arsort` per ordinare l'array `occurrences` dalla parola che comparsa maggiormente a quella che è comparsa con meno frequenza

```
56 // ORDINO LE PAROLE DA QUELLA CHE APPARE MAGGIORMENTE A QUELLA CHE APPARE MENO  
57 arsort($occurrences);
```

- Limitiamo l'array `occurrences` con il `MAX_RESULTS` (settato a 20) utilizzando la funzione `array_slice`

```
59 // PRENDI SOLO LE PRIME 20 PAROLE CON LA MAGGIOR FREQUENZA  
60 $occurrences = array_slice($occurrences, 0, MAX_RESULTS);
```

LA LOGICA DIETRO :

- Concateniamo in una variabile `returnstring` stringhe di codice html che eventualmente ritorneremo a fine funzione per mostrare il risultato in una tabella e per permetterci di navigare con nav-bars nelle varie funzioni

```
62 // COSTRUISCO UNA TABELLA PER VISUALIZZARE I RISULTATI
63 $returnstring = '<h5 class="text-uppercase font-weight-bold text-center">Le '.MAX_RESULTS.' parole che appaiono di più sono:</h5>';
64
65 $returnstring .= '<ul class="nav nav-tabs border-bottom-0 ml-5 mt-5" role="tablist">';
66 $returnstring .= '<li class="nav-item">';
67 $returnstring .= '<a class="nav-link active" data-toggle="tab" href="#results" role="tab">Risultati</a>';
68 $returnstring .= '</li>';
69 $returnstring .= '<li class="nav-item">';
70 $returnstring .= '<a class="nav-link" data-toggle="tab" href="#cloud" role="tab">Nuvoletta</a>';
71 $returnstring .= '</li>';
72 $returnstring .= '<li class="nav-item">';
73 $returnstring .= '<a class="nav-link" data-toggle="tab" href="#page" role="tab">Nella pagina</a>';
74 $returnstring .= '</li></ul>';
75 $returnstring .= '<div class="tab-content border p-3 rounded">';
76
77 $returnstring .= '<div class="tab-pane fade show active" id="results" role="tabpanel">';
78 $returnstring .= '<table class="table table-striped table-bordered mt-2 mb-0">';
79 $returnstring .= '<thead class="font-weight-bold"><tr>';
80 $returnstring .= '<tr>';
81 $returnstring .= '<td>Parola</td>';
82 $returnstring .= '<td>Occorrenze</td>';
83 $returnstring .= '<td>Percentuale (sul totale)</td>';
84 $returnstring .= '</tr></thead><tbody>';
```


LA LOGICA DIETRO :

- Creiamo un array per la cloud words, mettiamo il titolo dell'articolo in una variabile title, e il contenuto dell'articolo contenuto in paragrafi all'interno della variabile body

```
86      // PAROLE PER LA WORD CLOUD
87      $cloud_words = array();
88
89      $title = "<h4>".searchBytag("title", $url)."</h4>";
90
91      $body = searchByTag("p", $url);
92
```


LA LOGICA DIETRO :

- Calcoliamo la percentuale di parole, e le mettiamo in una tabella con le parole e il numero di volte che le abbiamo contate, rendiamo in grassetto la parola.

```
93     foreach($occurrences as $k=>$v){
94         // PERCENTUALE DELLE OCCORRENZE DELLA PAROLA
95         $perc = number_format((100 * $v) / $tot, 2)."%";
96
97         $returnstring .= '<tr>';
98         $returnstring .= '<td width="50%">'. $k. '</td>';
99         $returnstring .= '<td width="30%">'. $v. '</td>';
100        $returnstring .= '<td width="20%">'. $perc. '</td>';
101        $returnstring .= '</tr>';
102
103        $title = preg_replace("/\b".$k."\b/", "<strong>".$k."</strong>", strtolower($title));
104        // $title = str_replace($k, "<strong>".$k."</strong>", strtolower($title));
105        $body = preg_replace("/\b".$k."\b/", "<strong>".$k."</strong>", strtolower($body));
106        // $body = str_replace($k, "<strong>".$k."</strong>", strtolower($body));
107    }
```

LA LOGICA DIETRO :

- Mettiamo le parole all'interno del vettore `cloud_words` e lo splittiamo usando `implode` in modo da mettere ; alla fine di ogni parola, intanto concateniamo altro codice html(l'iframe con dentro il `cloud_words` collegato al suo API e diamo la funzione `fade` al tab) e ritorno tutto

```
108 // CICLO LE PAROLE PER INSERIRLE NELLA STRINGA DELLA WORD CLOUD
109 for($i = 0; $i < $v; $i++)
110     $cloud_words[] = $k;
111 }
112
113 $returnstring .= '</tbody></table></div>';
114
115 $cloud_words = implode(";", $cloud_words);
116
117 $returnstring .= '<div class="tab-pane fade" id="cloud" role="tabpanel">';
118 $returnstring .= '<iframe width="100%" height="500" class="border-0 mt-5" src="getcloud.php?max='.MAX_RESULTS.'&words='.$cloud_words.'"></iframe>';
119 $returnstring .= '</div>';
120
121 $returnstring .= '<div class="tab-pane fade" id="page" role="tabpanel">';
122 $returnstring .= $title.$body;
123 $returnstring .= '</div></div>';
124
125 insertData($url, $occurrences, $body, $title);
126
127 return $returnstring;
128
```

LA LOGICA DIETRO :

- Creiamo la funzione `searchByTag` utilizzata precedentemente, e prendo il contenuto della pagina html dall'url preso in input, e creiamo una variabile di contenuto HTML chiamata `dom`, prendo la variabile presa in input la chiamo `html`, la trasformo nella pagina vera e propria e la metto nel `dom` (la funzione `libxml_use_internal_errors` ci permette di nascondere gli eventuali errori di pagina).

```
130 function searchByTag($tag, $page){  
131     // PRENDO IL CONTENUTO DEL DOCUMENTO TRAMITE L'URL CHE MI VIENE DATO  
132     $dom = new DOMDocument();  
133     libxml_use_internal_errors(true);  
134  
135     $html = file_get_contents($page);  
136  
137     // CARICO L'HTML PROVENIENTE DALL'URL  
138     $dom->loadHTML($html);
```

LA LOGICA DIETRO :

- Calcoliamo la lunghezza dell'oggetto con la funzione `length` se il tag non è titolo perché il titolo sarà sempre 1, inizializziamo la variabile `testo` e ci mettiamo al suo interno tutto il testo preso dalla pagina html con il ciclo `for`, e poi lo ritorniamo.

```
143 // LA LUNGHEZZA DELL'OGGETTO CHE VIENE CREATO
144 $count = $tag == "title" ? 1 : $books->length;
145
146 $testo = "";
147
148 // CICLO TUTTI I TAG P PER PRENDERNE IL CONTENUTO
149 for($i = 0; $i < $count; $i++)
150     $testo .= $books->item($i)->nodeValue;
151
152 return $testo;
153 }
```

LA LOGICA DIETRO :

- Creiamo la funzione connectToDatabase che ci permetterà di accedere al database utilizzando le credenziali di accesso .

```
162 function connectToDatabase(){  
163     return mysqli_connect("localhost:3308","root","password","find");  
164 }
```

- Creiamo la funzione insertData che utilizzando la funzione connectToDatabase ci permetterà di accedere al database e inserire le variabili url,results,text,title nel DB

```
function insertData($url, $results, $text, $title){  
    $conn = connectToDatabase();  
  
    $res = "";  
  
    foreach($results as $k=>$v)  
        $res .= $k." (". $v." volte) ";  
  
    $conn->query("INSERT INTO ricerca (url, risultati, testo, titolo) VALUES ('.$url.', '$res.', '".addslashes($text)."', '".addslashes($title). "') pr die  
    (mysqli_error($conn));  
}
```

LA LOGICA DIETRO :

- Creiamo la funzione connectToDatabase che ci permetterà di accedere al database utilizzando le credenziali di accesso .

```
162 function connectToDatabase(){  
163     return mysqli_connect("localhost:3308","root","password","find");  
164 }
```

- Creiamo la funzione insertData che utilizzando la funzione connectToDatabase ci permetterà di accedere al database e inserire le variabili url,results,text,title nel DB

```
function insertData($url, $results, $text, $title){  
    $conn = connectToDatabase();  
  
    $res = "";  
  
    foreach($results as $k=>$v)  
        $res .= $k." (". $v." volte) ";  
  
    $conn->query("INSERT INTO ricerca (url, risultati, testo, titolo) VALUES ('.$url.', '$res.', '".addslashes($text)."', '".addslashes($title). "') pr die  
    (mysqli_error($conn));  
}
```

LA LOGICA DIETRO

- Creiamo la funzione `getDataFromDB` che ci permetterà di mostrare la data e l'id dal database e metterle all'interno del nav-bar laterale in modo da decidere quale contenuto selezionare in base a quando è stata fatta la ricerca.

```
177 function getDataFromDB(){
178     $conn = connectToDatabase();
179
180     $history = $conn->query("SELECT data, id FROM ricerca ORDER BY data DESC");
181
182     $div = "";
183
184     while($h = $history->fetch_assoc()){
185         $div .= '<a href="#page-content" class="d-block p-2 border-bottom pointer nav-link small"><span id="'. $h['id'].'">'. $h['data']. '</span></a>';
186     }
187
188     return $div;
189 }
```


LA LOGICA DIETRO

- Creiamo la funzione getContent che ci permette di prendere il titolo il testo e i risultati e mostrarli all'interno della pagina.

```
201 // PER CHIAMATA AJAX
202 function getContent($id){
203     $conn = connectToDatabase();
204     // PRENDO IL SALVATAGGIO SELEZIONATO DAL DATABASE
205     $content = $conn->query("SELECT titolo, testo, risultati FROM ricerca WHERE id = '". $id ."' LIMIT 1")->fetch_assoc();
206
207     // RESTITUISCO IL TITOLO, IL CONTENUTO DELL'ARTICOLO (CON RISULTATI EVIDENZIATI) E LE PAROLE TROVATE
208     $return = $content['titolo'];
209     $return .= '<p>'. $content['testo']. '</p>';
210     $return .= '<p>'. $content['risultati']. '</p>';
211     return $return;
212 }
```



RINGRAZIAMENTI

- FITSTIC
- MATTEO VENTURI
- MATTEO MASCELLANI
- LE NOSTRE MAMME CHE CI HANNO SOSTENUTO
- I FIGLI DI ENORE