

مکتب شریف

اولین بوتکمپ آموزشی - استخدامی ایران



سری دهم

مکتب ۸۱





۱- بخش JavaScript

1-1 - قطعه کد زیر را در نظر بگیرید:

```
<form name="search">
  <label>Search the site:
    <input type="text" name="search">
  </label>
  <input type="submit" value="Search!">
</form>

<hr/>

<form name="search-person">
  Search the visitors:
  <table id="age-table">
    <tbody><tr>
      <td>Age:</td>
      <td id="age-list">
        <label>
          <input type="radio" name="age" value="young">less than
18</label>
        <label>
          <input type="radio" name="age" value="mature">18-50</label>
        <label>
          <input type="radio" name="age" value="senior">more than
50</label>
        </td>
      </tr>

      <tr>
        <td>Additionally:</td>
        <td>
          <input type="text" name="info[0]">
          <input type="text" name="info[1]">

```



```

        <input type="text" name="info[2]">
      </td>
    </tr>

  </tbody></table>
  <input type="submit" value="Search!">
</form>

```

حال با استفاده از **selector** ها، قسمت های خواسته شده را مشخص کنید.

1. The table with `id="age-table"`.
2. All `label` elements inside that table (there should be 3 of them).
3. The first `td` in that table (with the word "Age").
4. The form with `name="search"`.
5. The first `input` in that form.
6. The last `input` in that form.

1-2- تابعی بنویسید که با توجه به ورودی کاربر، یک لیست ایجاد کند.

برای هر آیتم لیست:

- از کاربر در مورد محتوای آن با استفاده از **prompt** بپرسید.
- `` را با مقدار ورودی کاربر ایجاد کرده و به `` اضافه کنید.
- ادامه دهید تا کاربر افزودن ورودی را لغو کند (با فشار دادن **Esc** یا از طریق یک ورودی خالی).

توجه داشته باشید که همه عناصر باید به صورت پویا ایجاد شوند و اگر کاربر تگ های **HTML** را تایپ کند، باید با آنها مانند یک متن رفتار شود.

1-4- تابع **createTree** را به نحوی بنویسید که یک لیست `ul/li` تودرتو از اشیا تودرتو ایجاد میکند.

به عنوان مثال اگر **Object** زیر را داشته باشیم:

```

let data = {
  "Fish": {
    "trout": {},
    "salmon": {}
  },

```




```
"Tree": {
  "Huge": {
    "sequoia": {},
    "oak": {}
  },
  "Flowering": {
    "apple tree": {},
    "magnolia": {}
  }
}
```

نتیجه پس از فراخوانی به صورت زیر خواهد بود:

```
let container = document.getElementById('container');
createTree(container, data); // creates the tree in the container
```

- Fish
 - trout
 - salmon
- Tree
 - Huge
 - sequoia
 - oak
 - Flowering
 - apple tree
 - magnolia

5-1- تابع `clear(elem)` را به نحوی بنویسید که همه چیز را از یک عنصر حذف کند.

```
<ol id="elem">
  <li>Hello</li>
  <li>World</li>
</ol>
```

```
<script>
function clear(elem) { /* your code */ }

clear(elem); // clears the list
</script>
```

6-1- تابع `showNotification(Options)` را به گونه ای بنویسید که یک اعلان با محتوای داده شده ایجاد می کند. اعلان باید به طور خودکار پس از 1.5 ثانیه ناپدید شود.

در این تابع، option ها عبارتند از:

```
// shows an element with the text "Hello" near the right-top of the
window
showNotification({
  top: 10, // 10px from the top of the window (by default 0px)
  right: 10, // 10px from the right edge of the window (by default 0px)
  html: "Hello!", // the HTML of notification
  className: "welcome" // an additional class for the div (optional)
});
```

7-1- برنامه ای بنویسید که به محض کلیک بر روی `button`، ابتدا تگ `div` با متن `text` و سپس خود `button` ناپدید شود.

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  <input type="button" id="hider" value="Click to hide the text" />
  <div id="text">Text</div>
</body>
</html>
```

8-1- منویی ایجاد کنید که با کلیک کردن باز/بسته شود:

حالت باز:

Sweeties (click me)!



- Cake
- Donut
- Honey

حالت بسته:

Sweeties (click me)!

9-1- فرض کنید لیستی از پیغام ها وجود دارد. برنامه ای بنویسید که عملکرد دکمه بستن را به گوشه سمت راست بالای هر پیام اضافه کند.

نتیجه باید به صورت زیر باشد و با کلیک بر روی ضربدر، آن پیغام پاک شود.

Horse

[x]

The horse is one of two extant subspecies of *Equus ferus*. It is an odd-toed ungulate mammal belonging to the taxonomic family Equidae. The horse has evolved over the past 45 to 55 million years from a small multi-toed creature, *Eohippus*, into the large, single-toed animal of today.

Donkey

[x]

The donkey or ass (*Equus africanus asinus*) is a domesticated member of the horse family, Equidae. The wild ancestor of the donkey is the African wild ass, *E. africanus*. The donkey has been used as a working animal for at least 5000 years.

10-1- تابعی برای ایجاد رفتار یک tool tip ایجاد کنید.



هنگامی که ماوس روی عنصری با tool tip می آید، tool tip باید روی آن ظاهر شود و وقتی خارج شد، پنهان شود.

جزئیات:

- فاصله بین عنصر و tool tip باید 5 پیکسل باشد.
- در صورت امکان، موقعیت tool tip باید نسبت به عنصر در مرکز قرار گیرد.
- tool tip نباید از window عبور کند. معمولاً باید بالای عنصر باشد، اما اگر عنصر در بالای صفحه است و فضایی برای tool tip وجود ندارد، در زیر آن قرار بگیرید.



۲- بخش HTML و CSS

1-2 طرح زیر را در تمامی نماها (دسکتاپ، موبایل و تبلت) با استفاده از بوت استرپ و کامپوننت های آن پیاده سازی کنید.

لینک طرح : <https://getbootstrap.com/docs/4.4/examples/blog/>

نکات

- مهلت ارسال تمرین تا پایان روز پنج شنبه 1401/06/31 است. پاسخ تمرین را در کارتابل آموزشی خود ارسال کنید.
- نام فایل و ایمیل ارسالی خود را به این صورت قرار دهید: Name_hw10_maktab81 به عنوان مثال .Mohammad_Ali_Kargar_hw10_maktab81
- تصاویر تمرین بخش HTML در کنار این فایل ضمیمه شده است.
- ارسال پاسخ تمرینی که در کلاس تحویل داده شده، الزامی است.
- در صورت لزوم یک فایل word به عنوان توضیح در کنار کدهای خود قرار دهید.
- در صورتی که تمرین شامل چند فایل و فولدر می باشد حتماً آن ها را قالب یک فایل فشرده شده تجمیع کنید.
- به دلیل بروز برخی مشکلات در ارسال فایل ها با پسوند rar، در صورتی که قصد ارسال فایل نهایی به صورت فایل rar دارید، پسوند آن را از rar به ra و در صورت عدم ارسال مجدد به pdf یا ... تغییر دهید.
- از ارسال شدن ایمیل خود اطمینان حاصل فرمایید.
- در صورتی که سوالی دارید در گروه تلگرامی بپرسید.