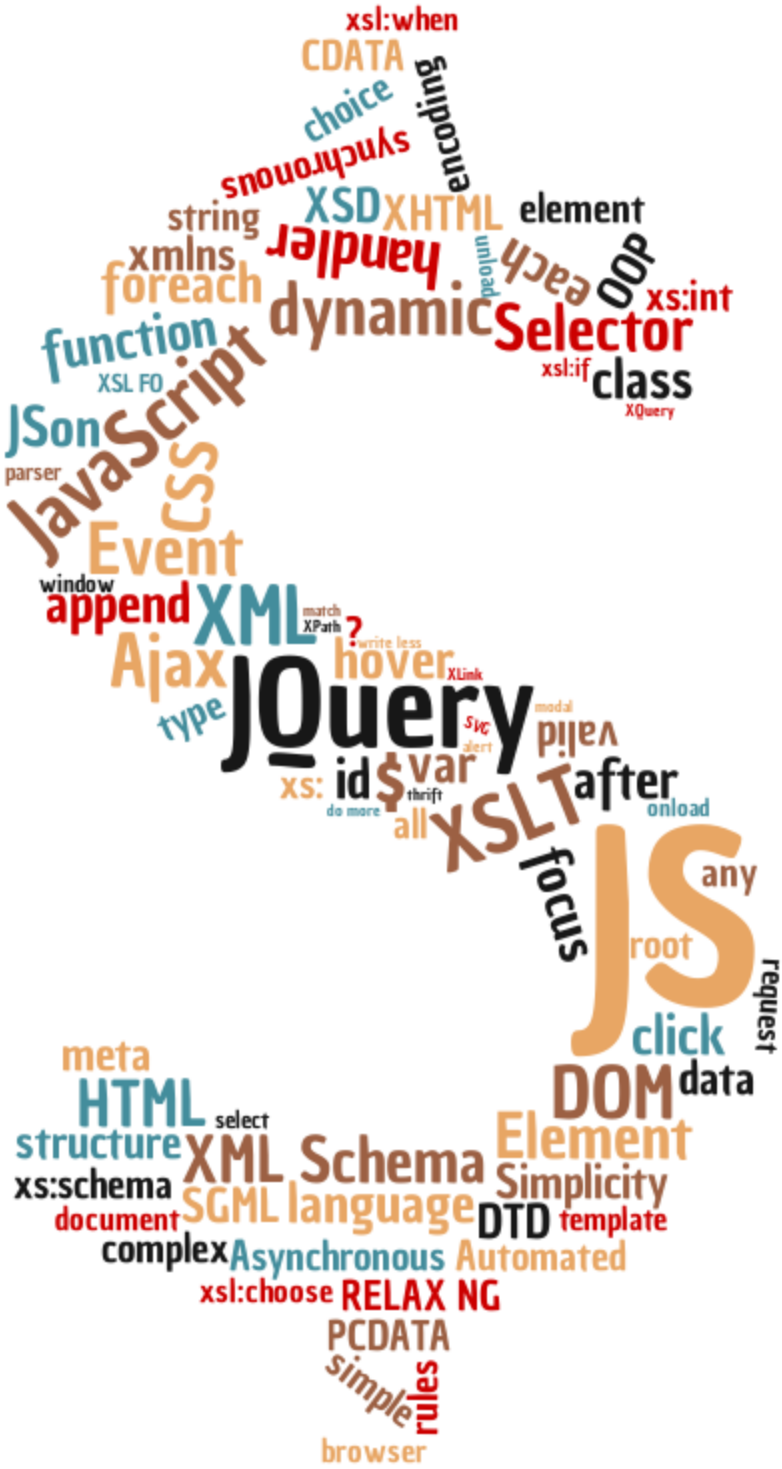
«به نام خدا»



در این تمرین از شما خواسته شده است که با استفاده از JavaScript، XML، XSLT ، Schema XML، JQuery و Ajax نسبت به پیاده‌سازی یک کلاینت ساده و پویا برای یک سایت بازی اقدام کنید.

نسخه‌ی ایستای این سایت را در قالب فایل‌های HTML و CSS در اختیارتان قرار داده‌ایم و شما تنها باید با ارسال درخواست Ajax و دریافت XML پاسخ و سپس پردازش آن با استفاده از JavaScript، JQuery و یا XSLT، DOM صفحه را به طور مناسبی تغییر دهید.

در قسمتی از تمرین نیز لازم است که مطابق با ساختاری که با XML Schema توصیف کرده و در اختیارتان قرار داده‌ایم، یک XML معتبر تولید کرده و برای اعتبارسنجی و دریافت پاسخ به سرور ارسال کنید.

توضیحات تکمیلی و نیازمندی‌های کامل بخش‌های مختلف تمرین در ادامه به تفصیل آورده شده است.

موفق باشید ☺

صورت مسأله

همان‌طور که در ابتدا بیان شد، هدف از این تمرین، پیاده سازی client پویا و ساده‌ای برای یک سایت بازی است. این سایت دارای یک صفحه اصلی به نام خانه (home.html) و چند بازی است. از میان این بازی‌ها تنها دو بازی سودوکو و شطرنج فعال هستند و بازی‌های دیگر هنوز پیاده‌سازی نشده‌اند. با کلیک بر روی هر بازی باید یک درخواست Ajax به آدرسی که در اختیارتان قرار داده می‌شود، ارسال کنید و با استفاده از JavaScript، JQuery و یا XSLT پاسخ دریافتی را پردازش و صفحه‌ی بازی را تولید کنید و در سپس نیازمندی‌های هر بازی را که در ادامه بیان خواهد شد، پیاده‌سازی نمایید. لازم به ذکر است که آدرس صفحه به هیچ وجه **نباید** تغییر کند و همه‌ی موارد باید با حذف و اضافه کردن محتوای جدید از و به صفحه‌ی خانه انجام شوند. محتوای جدید در هر مرحله از پردازش‌های صورت گرفته بر روی XML دریافتی تولید می‌شود.

HTML و CSS مورد نیاز تمرین در اختیارتان قرار داده شده است و شما تنها باید به پیاده سازی موارد خواسته شده بپردازید. **(مجاز به تغییر کدهای HTML و CSS تمرین که در اختیارتان قرار داده شده است، نیستید.)**

روند انجام تمرین

فرآیند کلی اجرایی در این تمرین به قرار زیر است (جزئیات بیشتر را در ادامه خواهید دید) :

پس از باز کردن صفحه‌ی خانه، با ارسال درخواست Ajax به این آدرس، به XML مربوط به صفحه دسترسی پیدا می‌کنید. در این فایل علاوه بر اطلاعاتی در مورد صفحه‌ی خانه، لیست کل بازی‌های سایت و ارجاعاتی به آدرس فایل‌های XML-ای که اطلاعات بازی‌های سودوکو و شطرنج در آن‌ها ذخیره شده است، وجود دارد. با کلیک بر روی آیتم مربوط به هر بازی فعال، باید با ارسال درخواست Ajax به آدرس مربوط به آن بازی، اطلاعات مربوط به آن بازی را دریافت و صفحه را به روز رسانی کنید. به طور خلاصه:

- اطلاعات موجود در XML اولیه (صفحه‌ی خانه) را با **JavaScript** پردازش کرده و صفحه اول سایت، همان صفحه‌ی خانه را ایجاد می‌کنید / تغییر می‌دهید. (در موارد مربوط به صفحه‌ی خانه مجاز به استفاده از JQuery نیستید)
- با کلیک بر روی بازی سودوکو، جدول اولیه‌ی مربوط به بازی سودوکو را با دریافت اطلاعات بازی از XML آن و سپس پردازش آن اطلاعات با **XSLT** تولید می‌کنید و در ادامه مجاز به استفاده از JavaScript و JQuery هستید.
- با کلیک بر روی بازی شطرنج و دریافت اطلاعات بازی در قالب XML، نسبت به پیاده‌سازی نیازمندی‌های آن اقدام می‌کنید. در بازی شطرنج مختارید که به انتخاب خود از JavaScript، JQuery و XSLT و یا ترکیبی از آن‌ها استفاده کنید.
- با کلیک بر روی بازی‌هایی که هنوز پیاده سازی نشده‌اند در div#main-container عبارت « This game is not implemented yet!» نمایش داده شود.
- با کلیک بر روی هر بازی، محتوای div#main-container با محتوای جدیدی که مربوط به بازی مورد نظر است جایگزین شود.

در ادامه جزییات بیشتری شرح داده شده است، ضمناً موارد **این رنگی** اختیاری هستند!

صفحه‌ی خانه

با باز کردن صفحه‌ی خانه یک درخواست Ajax به آدرس زیر فرستاده و XML اطلاعات صفحه را دریافت کنید.

http://ie.ce-it.ir/hw3/xml/home.xml

این فایل XML، ساختار زیر را دارد:

```
<home>
	<header>
		<background>#358FBA</background>
		<pwd>#002641</pwd>
		<gameicon color="#3B096E" hover="red">
			<game color="#C79953" hover="#613C05" />
		</gameicon>
	</header>
	<games max-onlines-background="red" max-onlines-border-width="2px" max-onlines-border-color="rgb(255, 134, 131)" max-onlines-border-style="dashed">
		<game active="true">
			<name>sudoku</name>
			<image>./sudoku.png</image>
			<url>path/to/xml/file</url>
			<onlines>6</onlines>
			<text color="black" hover="#88000D">Play Sudoku!</text>
		</game>
		.
		.
		.
	</games>
</home>
```

توضیحات XML خانه

- :home به عنوان ریشه‌ی XML است و همه‌ی اطلاعات مربوط به صفحه‌ی خانه در آن وجود دارد.
  - :header حاوی اطلاعات زیر است.
    - :background رنگ پس‌زمینه‌ی header
    - :pwd رنگ مربوط به span#pwd
    - :gameicon اطلاعات مربوط به آیکون بازی موجود در هدر (اولین عنصر li در ul#games) و لیست بازی‌های فعال (سایر عناصر li در ul#games) در این برچسب قرار دارند.
      - :color رنگ آیکون بازی در حالت عادی
      - :hover رنگ آیکون بازی در حالت اشاره شده
      - :game اطلاعات مربوط به نحوه‌ی نمایش بازی‌ها در لیست بازی‌های فعال در این برچسب قرار دارند.
        - :color رنگ بازی‌های موجود در لیست بازی‌های فعال در حالت عادی

        - :hover رنگ بازی‌های موجود در لیست بازی‌های فعال در حالت اشاره شده
  - :games حاوی اطلاعات مربوط به تمام بازی‌هاست.
    - :max-onlines-background رنگ پس‌زمینه‌ی بلوک بازی‌ای که بیشترین تعداد افراد آنلاین را دارد.
    - :max-onlines-border-width ضخامت کادر بلوک بازی‌ای که بیشترین تعداد افراد آنلاین را دارد.
    - :max-onlines-border-color رنگ کادر بلوک بازی‌ای که بیشترین تعداد افراد آنلاین را دارد.
    - :max-onlines-border-style نوع کادر بلوک بازی‌ای که بیشترین تعداد افراد آنلاین را دارد.
    - :game این برچسب در بردارنده‌ی اطلاعات مربوط به هر بازی است. (div.game-block)
      - :active بیان کننده‌ی فعال / غیر فعال بودن بازی است.
      - :name نام بازی
      - :image آدرس تصویر بازی
      - :url آدرس فایل XML حاوی اطلاعات تفصیلی بازی
      - :onlines تعداد افراد آنلاین در بازی
      - :text متن مربوط به بازی
        - :color رنگ متن مربوط به بازی در حالت عادی
        - :hover رنگ متن مربوط به بازی در حالت اشاره شده

نیازمندی‌ها و نکات پیاده‌سازی صفحه‌ی خانه

- با باز شدن صفحه‌ی خانه یک درخواست Ajax به آدرس ذکر شده فرستاده و XML اطلاعات صفحه را دریافت کنید. شما باید XML دریافتی را **تنها با JavaScript پردازش کرده** و با استفاده از اطلاعات استخراجی از آن، موارد زیر را با استفاده از JavaScript انجام دهید.
- مکانی که در آن قرار داریم همواره در span#pwd نشان داده شود (home/sudoku/chess/mario/...)
- رنگ پس‌زمینه‌ی header، رنگ span#pwd و آیکون بازی در حالت عادی و اشاره شده، از XML استخراج و در صفحه اعمال شوند.
- بازی‌های فعال به لیست بازی‌های فعال موجود در header (ul#games) افزوده شوند و رنگ آن‌ها در حالت عادی و اشاره شده از XML استخراج و بر روی آن‌ها اعمال شود.
- با کلیک بر روی آیکون لیست بازی‌های فعال لیست باز شود و با کلیک مجدد لیست بسته شود. در ابتدا لیست بسته است.
- آیکون خانه (#home-icon) تنها در صفحات بازی نشان داده شود و در صفحه‌ی خانه مخفی باشد. **با کلیک بر روی این آیکون از سایر صفحات می‌توان به صفحه‌ی خانه بازگشت و بازی دیگری را انتخاب نمود. (دقت کنید که فقط محتوای div#main-container با محتوای جدید جایگزین می‌شود.)**
- با کلیک بر روی هر بازی در لیست بازی‌های فعال و یا بلوک بازی‌ها، محتوای مناسب را تولید کرده و نمایش دهید.
- رنگ پس‌زمینه و کادر بلوک بازی‌ای که بیشترین تعداد افراد آنلاین را دارد باید با استفاده از اطلاعات XML تغییر کند.
- از ساختار زیر برای هر بلوک بازی استفاده کنید و بلوک‌ها را به div#main-container اضافه کنید:

```
<div class="game-block" id="chess-block" data-onlines="3">
      <div class="game-image-container">
            <img src="./chess-match.jpg" alt="">
      </div>
      <p>Play Chess!</p>
</div>
```

(فایل home_example.html موجود در پوشه‌ی examples)

بلوک بازی در حالت عادی

بلوک بازی در حالت اشاره شده

بازی سودوکو

با کلیک بر روی بازی سودوکو در صفحه‌ی خانه (#sudoku-block) یک درخواست Ajax به آدرس زیر فرستاده و XML اطلاعات بازی را دریافت کنید.

http://ie.ce-it.ir/hw3/xml/sudoku.xml

این فایل XML، ساختار زیر را دارد:

```
<sudoku selectedNumberColor="white" selectedNumberBackColor="red"
hover="yellow">

      <row> <!-- سطر صفر -->
            <cell>1</cell>    <!-- خانه‌ی (۰و۰) -->
            <cell></cell>     <!-- خانه‌ی (۰و۱) -->
            <cell>3</cell>    <!-- خانه‌ی (۰و۲) -->
            <cell></cell>     <!-- خانه‌ی (۰و۳) -->
            <cell>5</cell>    <!-- خانه‌ی (۰و۴) -->
            <cell></cell>     <!-- خانه‌ی (۰و۵) -->
            <cell>7</cell>    <!-- خانه‌ی (۰و۶) -->
            <cell></cell>     <!-- خانه‌ی (۰و۷) -->
            <cell></cell>     <!-- خانه‌ی (۰و۸) -->
      </row>
         .
         .
         .
      <row> <!-- سطر هشتم -->
            <cell>4</cell>    <!-- خانه‌ی (۸و۰) -->
            <cell></cell>     <!-- خانه‌ی (۸و۱) -->
            <cell>9</cell>    <!-- خانه‌ی (۸و۲) -->
            <cell></cell>     <!-- خانه‌ی (۸و۳) -->
            <cell>1</cell>    <!-- خانه‌ی (۸و۴) -->
            <cell></cell>     <!-- خانه‌ی (۸و۵) -->
            <cell>8</cell>    <!-- خانه‌ی (۸و۶) -->
            <cell></cell>     <!-- خانه‌ی (۸و۷) -->
            <cell>3</cell>    <!-- خانه‌ی (۸و۸) -->
      </row>

</sudoku>
```

توضیحات XML سودوکو

- در این ساختار برچسب sudoku در ریشه قرار دارد و دارای سه صفت (attribute) است:
- selectedNumberColor: بیان کننده رنگ اعداد انتخابی می باشد. به این صورت که اگر یک عدد در جدول انتخاب شد تمامی اعداد برابر با آن عدد در جدول نیز به این رنگ در می‌آیند.
- selectedNumberBackColor: بیان کننده رنگ پس‌زمینه‌ی اعداد انتخابی می باشد. به این صورت که اگر یک عدد در جدول انتخاب شد پس‌زمینه‌ی تمامی خانه‌های حاوی آن عدد به این رنگ در می‌آید. لازم به ذکر است که در این مورد و مورد قبل، چنانچه خانه از حالت انتخاب خارج شد، رنگ اعداد و رنگ پس‌زمینه‌ی خانه‌هایی که تغییر کرده است به رنگ سابق خود برخواهد گشت.
- hover: با قرار گیری نشانه گر بر روی یک خانه، رنگ پس‌زمینه‌ی آن خانه به این مقدار تغییر می کند. بدیهی است که پس از خروج نشانه‌گر رنگ خانه باید به رنگ سابق برگردد.
- هر سطر از جدول در یک row و هر خانه از جدول در یک cell قرار گرفته‌اند. چنانچه خانه‌ای مقدار نداشته باشد در جدول نیز این خانه باید خالی باشد.

نیازمندی‌ها و نکات پیاده‌سازی بازی سودوکو

- با کلیک بر روی بازی سودوکو در صفحه‌ی خانه (#sudoku-block) یک درخواست Ajax به آدرس ذکر شده فرستاده و XML اطلاعات بازی را دریافت کنید. شما باید XML دریافتی را با استفاده از XSLT Processor پردازش کرده و جدول سودوکو را ساخته و آن را جایگزین محتوای div#main-container کنید.
- با انتخاب هر خانه از جدول تمامی خانه‌های دارای اعداد یکسان با خانه‌ی انتخابی مشخص شوند. (selectedNumberColor و selectedNumberBackColor)
- در هر خانه تنها **اعداد یک تا نه (۹ .. ۱)** می‌توانند وارد شوند و کد شما **نباید** اجازه‌ی وارد کردن کاراکتری خارج از این مجموعه را به بازیکن بدهد.
- با زدن دکمه «Check it out!» درستی مقادیر خانه ها بررسی شده و پیغام مناسبی نمایش داده شود و بازی ادامه پیدا کند. **در صورتی که خانه‌ای با توجه به قوانین جدول سودوکو اشتباه پر شده باشد، خانه مورد نظر و قانون نقض شده مشخص می شوند (رنگ سطر، ستون یا بلوک ناقض قوانین بازی تغییر می‌کند).**
- با زدن دکمه‌ی «Submit» چنانچه جدول به صورت کامل پر شده باشد، یک XML مطابق با ساختاری که در فایل sudoku_solution.xsd توصیف شده است، تولید کنید و آن را با نام solution_xml در قالب یک درخواست Ajax از نوع POST به آدرس زیر بفرستید. سرور با اعتبارسنجی و بررسی صحت جدول حل شده، پیغامی به شما برمی‌گرداند. پیغام دریافتی را به کاربر نشان دهید.
    - http://ie.ce-it.ir/hw3/sudoku_validator.php
- بعد از زدن دکمه‌ی «Submit» چنانچه جدول کامل پر نشده باشد، تنها یک alert ظاهر می‌شود و بازی ادامه می‌یابد.

پیغام دریافتی	توضیح
Error	مشکل در پارامترهای دریافتی
xml validation faild!	عدم تطابق XML تولیدی با ساختار XSD
Congratulation!	درست بودن پاسخ سودوکو
Oops! Wrong sulotion!	اشتباه بودن پاسخ سودوکو

جدول سودوکوی خود را باید مطابق با ساختار زیر و با استفاده از XSLT processor تولید کنید و جایگزین محتوای div#main- container نمایید:

```
<div id="sudoku">
      <table>
            <tr>
                  <td contenteditable="true"></td>
                  <td contenteditable="false">2</td>
                  <td contenteditable="false">3</td>
                  <td contenteditable="true"></td>
                  <td contenteditable="false">5</td>
                  <td contenteditable="true"></td>
                  <td contenteditable="false">7</td>
                  <td contenteditable="true"></td>
                  <td contenteditable="false">9</td>
            </tr> <!-- Other 8 Rows -->
      </table>

</div>

<div id="check-sudoku">Check it out!</div>

<div id="submit-sudoku">Submit</div>
```

	۲	۳		۵		۷		۹
۴	۵	۶	۷	۸	۹			
۷		۹		۲	۳			
۲	۳	۴		۶		۸		۱
				۹			۳	
۸	۹	۱		۳		۵		۷
۳					۸	۹		
	۷			۱			۴	
		۲	۳					۸

Check it out! Submit

با به کارگیری ساختار فوق (**فایل sudoku_example.html موجود در پوشه‌ی examples**)، جدول سودوکوی تولیدی شما به شکل بالا خواهد بود. **برای رسیدن به کارایی مناسب و پیاده‌سازی نیازمندی‌های مطرح شده، می‌توانید ساختار فوق را گسترش دهید.**

بازی شطرنج

با کلیک بر روی بازی سودوکو در صفحه‌ی خانه (#chess-block) یک درخواست Ajax به آدرس زیر فرستاده و XML اطلاعات بازی را دریافت کنید.

http://ie.ce-it.ir/hw3/xml/chess.xml

این فایل XML، ساختار زیر را دارد:

```
<chess turn="black">
	<board white-cells="#D59330" black-cells="#A15C21">
		<white field="top">
			<pawn row="6" col="0" />
			...
			<bishop type="black" row="7" col="2" />
			<bishop type="white" row="7" col="5" />
			...
		</white>
		<black field="bottom"> ... </black>
	</board>
	<score>
		<white>5</white>
		<black>6</black>
	</score>
	<chessmans>
		<pawn unicode="&#9823;">
			<white>pawns-white.png</white>
			<black>pawns-black.png</black>
		</pawn>
		...
	</chessmans>
</chess>
```

توضیحات XML شطرنج

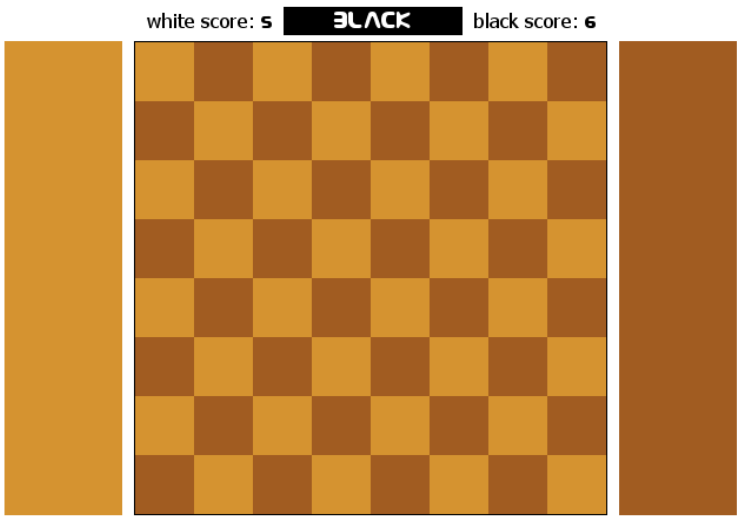
- `chess:` به عنوان ریشه‌ی XML است و همه‌ی اطلاعات مربوط به بازی شطرنج در آن وجود دارد.
  - `turn:` بیان کننده‌ی این است که «**نوبت** حرکت چه بازیکنی است؟». دو مقدار `black` و `white` می‌گیرد .
- `board:` در بر دارنده‌ی ویژگی‌های صفحه‌ی شطرنج و وضعیت قرارگیری مهره‌های هر بازیکن در زمین شطرنج است.
  - `white-cells:` رنگ خانه‌های سفید شطرنج
  - `black-cells:` رنگ خانه‌های سیاه شطرنج
  - `white:` در بردارنده‌ی مهره‌های بازیکن سفید است.
    - `field:` موقعیت زمین بازیکن سفید را مشخص می‌کند و دو مقدار `top` و `bottom` را به خود می‌گیرد. (این صفت در پیشروی سربازهای بازیکن سفید مؤثر است.)
    - `pawn، rook، knight، queen، king:` به ترتیب موقعیت مهره‌های پیاده، رخ (قلعه)، اسب، وزیر و شاه را مشخص می‌کنند.
      - `row:` بیان کننده‌ی این است که مهره در کدام سطر از صفحه‌ی شطرنج قرار دارد. مقادیر مجاز برای این صفت اعداد صفر تا ۷ هستند.
      - `col:` بیان کننده‌ی این است که مهره در کدام ستون از صفحه‌ی شطرنج قرار دارد. مقادیر مجاز برای این صفت اعداد صفر تا ۷ هستند.
    - `bishop:` علاوه بر موقعیت مهره‌ی فیل (`row` و `col`) نوع آن را نیز مشخص می‌کند.
      - `type:` دو مقدار `black` و `white` به خود می‌گیرد و به ترتیب نشان دهنده‌ی فیل سیاه و فیل سفید است.
  - `black:` در بردارنده‌ی مهره‌های بازیکن مشکی است.
    - `field:` موقعیت زمین بازیکن مشکی را مشخص می‌کند و دو مقدار `top` و `bottom` را به خود می‌گیرد. (این صفت در پیشروی سربازهای بازیکن مشکی مؤثر است.)
    - موقعیت مهره‌ها و نوع آن‌ها دقیقاً مانند موارد بیان شده در `white` است.
- `score:` در بردارنده‌ی امتیازات بازیکن‌های سفید و مشکی است.
  - `white:` امتیاز بازیکن سفید
  - `black:` امتیاز بازیکن مشکی
- `chessmans:` در بردارنده‌ی تصاویر و یونیکد مهره‌های شطرنج است.
  - `pawn، rook، knight، bishop، queen، king:` به ترتیب حاوی یونیکد و تصاویر مربوط به مهره‌های پیاده، رخ (قلعه)، اسب، فیل، وزیر و شاه هستند.
    - `unicode:` یونیکد مهره
    - `white:` تصویر مهره برای بازیکن سفید
    - `black:` تصویر مهره برای بازیکن مشکی

نیازمندی‌ها و نکات پیاده‌سازی بازی شطرنج

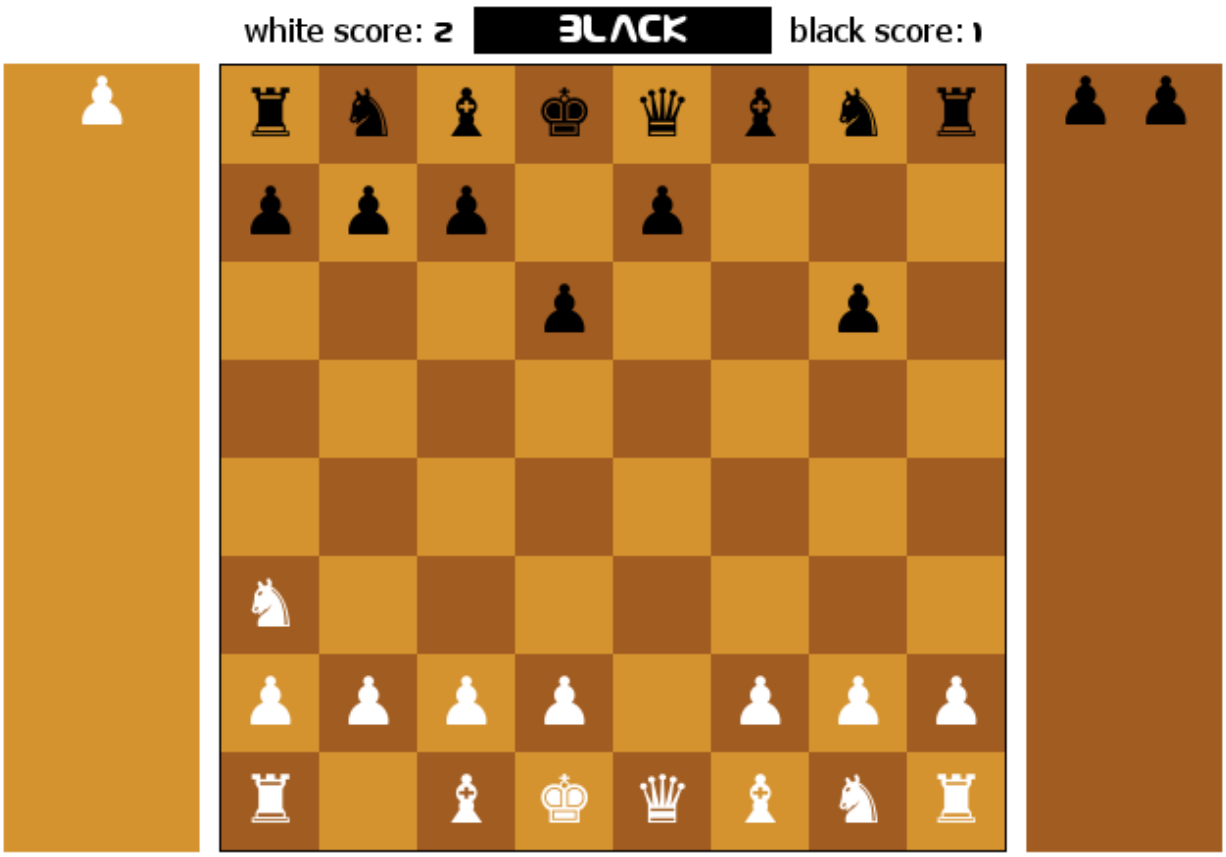
- با کلیک بر روی بازی شطرنج در صفحه‌ی خانه (`#chess-block`) یک درخواست Ajax به آدرس ذکر شده فرستاده و XML اطلاعات بازی را دریافت کنید. شما مختارید هر گونه که دوست دارید XML دریافتی را پردازش کنید و صفحه‌ی شطرنج را بسازید. همچون قبل، محتوای جدید را جایگزین محتوای قبلی `div#main-container` کنید.
- اگر تعداد مهره‌های بازیکنی در XML اطلاعات بازی ۱۶ عدد نبود به این معناست که بازیکن حریف قبلاً مهره‌هایی از او را زده است، مهره‌های زده شده باید در حاشیه‌ی صفحه و در قسمت تعبیه شده‌ی مناسب قرار بگیرند. مهره‌های مشکی در `#black-chessman-panel` و مهره‌های سفید در `#white-chessman-panel`.
- پیاده‌سازی قوانین حرکت مهره‌ها
- محاسبه و نمایش امتیاز هر بازیکن. امتیاز هر بازیکن را بر اساس ارزش مهره‌های زده شده از حریف در نظر بگیرید.
  - وزیر: ۹ امتیاز
  - رخ: ۵ امتیاز
  - فیل: ۳ امتیاز
  - اسب: ۳ امتیاز
  - پیاده: ۱ امتیاز
- نمایش دادن نوبت بازی
- چنانچه در روند بازی مهره‌ای زده شد، آن مهره بیرون از صفحه‌ی شطرنج و در حاشیه‌ی آن (در مکان مناسب) قرار بگیرد.
- چنانچه هر بازیکن در نوبت خود، بر روی مهره‌ای از خود کلیک کرد، آن مهره انتخاب شود (این امر مثلاً با تغییر رنگ خانه‌ی حاوی آن مهره **و یا تغییر رنگ خود مهره (لازم به استفاده از یونیکد به جای تصویر مهره است)** می‌تواند صورت بگیرد) و خانه‌های مجاز برای حرکت آن مهره در صفحه مشخص شوند (مثلاً با تغییر رنگ آن خانه‌ها و یا قرار دادن علامتی در آن‌ها). چنانچه مهره ای از حریف در آن خانه‌ها بود، آن خانه **(و یا آن مهره)** به رنگ قرمز در آید. با کلیک بر روی خانه‌ای جهت حرکت، همه‌ی خانه‌ها **به جزء خانه‌های مبدأ و مقصد به حالت اولیه برگردند و پس از حدود نیم ثانیه این دو خانه نیز** به رنگ معمول خود برگردند.
- تشخیص حالات «کیش» و «کیش و مات».
- پس از کیش و مات بازی تمام شود و بازیکن برنده اعلام شود.
- پیاده‌سازی حرکت «شاه-قلعه» (Castling): این حرکت در حالتی مجاز است که تمامی شرایط زیر برقرار باشد؛
  - شاه و قلعه در سطر ابتدایی خود باشند.
  - هیچ کدام از آن‌ها از آغاز بازی حرکت نکرده باشند.
  - هیچ مهره‌ای مابین شاه و رخ وجود نداشته باشد.
  - **شاه در حالت کیش نباشد و پس از این حرکت در حالت کیش قرار نگیرد و خانه‌های محل عبور شاه در حین این حرکت نیز در معرض کیش نباشند.**
- پیاده‌سازی قانون ترفیع پیاده (promotion): چنانچه پیاده‌ای در زمین حریف به خانه‌ی آخر رسید می‌تواند با وزیر، **رخ، فیل و یا اسب، به انتخاب خود بازیکن** تعویض شود. **در حالت انتخاب مهره جهت جایگزینی، کاربر با کلیک بر روی مهره‌های زده شده‌ی خود توسط حریف که در حاشیه‌ی صفحه‌ی شطرنج قرار دارند، مهره‌ها را با هم تعویض می‌کند.**
- **در نوبت هر بازیکن صفحه به سمت آن بازیکن قرار بگیرد (زمین او پایین باشد.)**
- **حرکت مهره‌ها به صورت انیمیشنی ساده پیاده‌سازی شود.**

از ساختار HTML زیر جهت ساختن صفحه‌ی شطرنج استفاده کنید و آن را جایگزین محتوای `div#main-container` نمایید:

```
<div id="chess">
	<div id="chess-info">
		<div class="score">
			white score: <span id="white-score">5</span>
		</div>
		<div id="turn" class="black">black</div>
		<div class="score">
			black score: <span id="black-score">6</span>
		</div>
	</div>
	<div id="white-chessman-panel"></div>
	<table>
		<tr>
			<td></td>
			<td></td>
			<td></td>
			<td></td>
			<td></td>
			<td></td>
			<td></td>
			<td></td>
		</tr> <!-- more 7 chess rows -->
	</table>
	<div id="black-chessman-panel"></div>
</div>
```



با بهره‌گیری از ساختار فوق (**فایل `chess_example.html` موجود در پوشه‌ی `examples`**) صفحه‌ی شطرنج تولیدی شما به شکل تصویر بالا خواهد بود. برای رسیدن به کارایی مناسب و پیاده‌سازی نیازمندی‌های مطرح شده، **می‌توانید ساختار فوق را گسترش دهید.**



«تصویری از یک صفحه‌ی شطرنج حین انجام بازی»

موارد امتیازی تمرین

- **موارد امتیازی تمرین، در متن با این رنگ مشخص شده‌اند.**
- این تمرین در مجموع ۲۵ تا ۳۰ نمره امتیازی دارد.

نکات پیاده سازی

- این تمرین به صورت انفرادی انجام می شود. در صورتی که تمرین تحویلی دو دانشجو مشابه هم باشد، برای هر دو نفر نمره صفر در نظر گرفته می‌شود.
- مشابه تمرین قبل، حق استفاده از هیچ گونه از کدهای آماده یا کتابخانه‌های مختلف را ندارید و تمامی کدها (JavaScript, JQuery, XSLT) باید توسط خودتان نوشته شود.
- در هنگام پیاده سازی به تعداد درخواست های Ajax خود دقت داشته باشید. برای هر فایل XML باید فقط یک بار درخواست بدهید. مثلا وقتی فایل XML را از سرور دریافت کرده اید، باید محتویاتش را در داخل یک متغیر در کد جاوا اسکریپت ریخته و در دفعات بعد، از این متغیر استفاده کنید. در صورت درخواست بیش از یک بار برای یک فایل XML از نمره شما کسر خواهد شد.
- توصیه می شود یک بار قبل از شروع کار، قالب فایل‌های XML را بررسی کنید تا درک درستی از سلسله مراتب آن‌ها داشته باشید.
- برنامه‌ای که می‌نویسید (شامل همه فایل های HTML, CSS, JS, JQuery, XSLT) به صورت local است و بر روی سرور قرار نمی‌گیرد بلکه در زمان اجرا به سرور وصل شده و اطلاعات (فایل های XML) را دریافت می‌کند.
- با توجه به اینکه مرورگر های امروزی same-origin-policy را رعایت می کنند، شما نمی توانید از یک دامین (در این تمرین، به صورت local است) به یک دامین دیگر (ce-it.ir) درخواست Ajax بدهید. برای این تمرین کافی است این سیاست محدودکننده را غیر فعال کنید. به عنوان مثال برای مرورگر کروم می توانید با استفاده از دستور زیر در خط فرمان این محدودیت را از بین ببرید:
  - `Chrome.exe --disable-web-security`

  و یا با استفاده از افزونه allow-origin-policy اقدام به این کار کنید.
- در صورتی که مشکل قبلی حل نشد، می توانید از یک php cross-domain proxy که بر روی یک وب سرور که بر روی کامپیوتر خود نصب می کنید استفاده نمایید.
- فایل های XML ای که الان در سایت هستند هنگام تحویل پروژه تغییر خواهند کرد، یعنی محتوای تمام فایل هایی که پروژه شما با آنها تصحیح خواهند شد با فایل های فعلی متفاوت خواهند بود. بنابر این در هیچ قسمت از کد برنامه خود نباید از مقادیر ثابت استفاده کنید و همه چیز باید کاملا generic و بر اساس فایل دریافتی باشد.
- خواندن فایل style.css و دقت در نحوه سبک‌دهی و به کارگیری شبه کلاس‌های استفاده شده و ترکیبی آن خالی از لطف نیست!!!

نکات تحویل تمرین

- علاوه بر فایل های HTML,CSS,JS,JQuery فایل XSLT نیز باید تحویل داده شود.
- موارد بالا را داخل یک پوشه گذاشته، آن را با zip فشرده کرده و آن را مطابق الگوی زیر نام‌گذاری کنید:
  - `HW3-StudentNumber-FirstNameLastName.zip`
- **زمان تحویل تمرین ساعت ۲۲:۰۰ روز جمعه ۷ خرداد ماه (۱۳۹۵/۳/۷) است.**
- به ازای هر ۸ ساعت تأخیر ۵ درصد (هر روز ۱۵ درصد) از نمره شما کسر خواهد شد. بنابراین در صورت تأخیر بیشتر از یک هفته هیچ نمره ای کسب نخواهید کرد.
- نمرات تمرین در صفحه‌ی درس (ceit.aut.ac.ir/~bakhshis/IE) اعلام می‌گردد.
- همچنین در صورتی که به سایت درس (ceit.aut.ac.ir/courses) دسترسی ندارید تمرین‌های خود را در زمان مقرر به ایمیل ehsanm94@gmail.com ارسال کنید.