

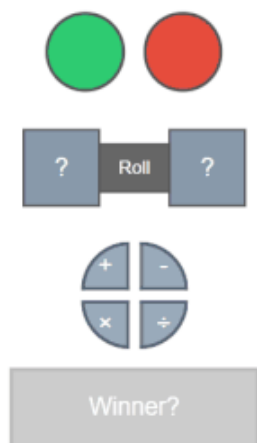
پرهیز از عدد اول - JS

در این سوال قصد داریم به طراحی بازی **پرهیز از عدد اول** بپردازیم. پرهیز از عدد اول، یک بازی دو نفره است که در آن هر بازیکن با یکی از رنگ‌های سبز، یا قرمز شناخته می‌شود. اجزای بازی عبارتند از: یک جدول 10×10 ، دو تاس با اعداد 0 تا 6، دکمه‌ی پرتاب تاس، چهار عملگر اصلی ریاضی، دو مهره‌ی سبز و قرمز، دو جایگاه شروع برای بازیکنان، و نمایشگر پیام.

در ابتدای بازی، مهره‌ها در جایگاه شروع قرار دارند. هر بازیکن می‌تواند در نوبت خود یکبار تاس‌ها را بیندازد، و سپس یکی از چهار عملگر اصلی را انتخاب کند. پس از آن، بازیکن به اندازه‌ی حاصل عملیات انجام‌شده از موقعیت فعلی خود در جدول یا حالت شروع جابه‌جا می‌شود؛ اما در صورتی که خانه‌ی مقصد یک خانه‌ی **غیرمجاز** باشد، مهره‌ی بازیکن به جایگاه شروع بازگردانده می‌شود. خانه‌ی غیرمجاز، خانه‌ای با شماره‌ی بزرگتر از صد، کوچکتر از یک، عدد اول، یا عدد تعریف‌نشده است. همچنین در صورتی که بازیکن وارد خانه‌ای شود که حریف

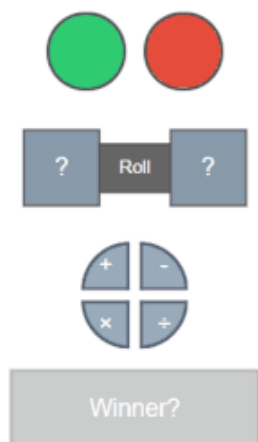
در آن قرار دارد، حریف به خانه‌ی شروع بازگردانده می‌شود و بازیکن وارد شده جای او را می‌گیرد.

در زیر یک دور کامل بازی به تصویر کشیده شده است. ابتدا نوبت بازیکن سبز است و با هر انجام نوبت، رنگ پس‌زمینه‌ی بازی، به رنگ بازیکن اتمام‌کننده‌ی نوبت در می‌آید (تعویض رنگ پس‌زمینه جزو پیاده‌سازی‌های بازی نیست و تنها برای توضیح مثال‌ها آورده شده است).



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

پیاده‌سازی بازی تا حدی در فایل‌های اولیه انجام شده است. در زیر محتوای قابل دریافت با اجرای فایل `avoid_primes.html` را مشاهده می‌کنید. در پیاده‌سازی اولیه هیچ کدام از دکمه‌ها کار نمی‌کنند، و بازیکن‌ها در حالت شروع هستند. شما باید منطق بازی را با تکمیل فایل `avoid_primes.js` به پایان برسانید.



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

شرح دقیق‌تر قوانین بازی

توجه کنید که

- در این بازی، تقسیم به صورت صحیح در نظر گرفته می‌شود؛ یعنی برای مثال $3/2 = 1$.

- منظور از عملیاتی با حاصل تعریف نشده، تقسیم بر صفر است؛ برای مثال $0/0$ ، و $1/0$ هر دو تعریف نشده هستند.

- تاس اول که همان تاس سمت چپ است، همواره عملوند اول را نشان می‌دهد.

- اعداد اول بین یک و صد عبارتند از:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59,

61, 67, 71, 73, 79, 83, 89, 97

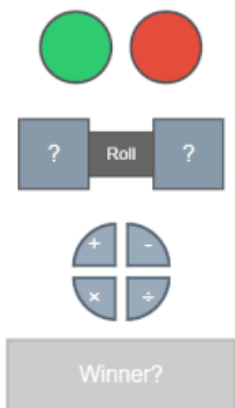
در این قسمت گیف‌های نمونه برای توضیح قوانین بازی را مشاهده می‌کنید. در ابتدای تمام مثال‌ها نوبت با بازیکن سبز است.

- در هر نوبت، بازیکن اجازه دارد یکبار تاس‌ها را بیندازد، و پس از آن عملگر مورد نظر خود را انتخاب کند. در این صورت نوبت او به اتمام می‌رسد و نوبت بازیکن بعدی می‌شود.

- پس از انتخاب عملگر و انجام نوبت، نوشته‌ی تاس‌ها به ؟ تغییر می‌کند.

- در صورتی که بازیکن پیش از انتخاب عملگر، مجدداً دکمه‌ی پرتاب تاس‌ها را فشار دهد هیچ اتفاقی نمی‌افتد.

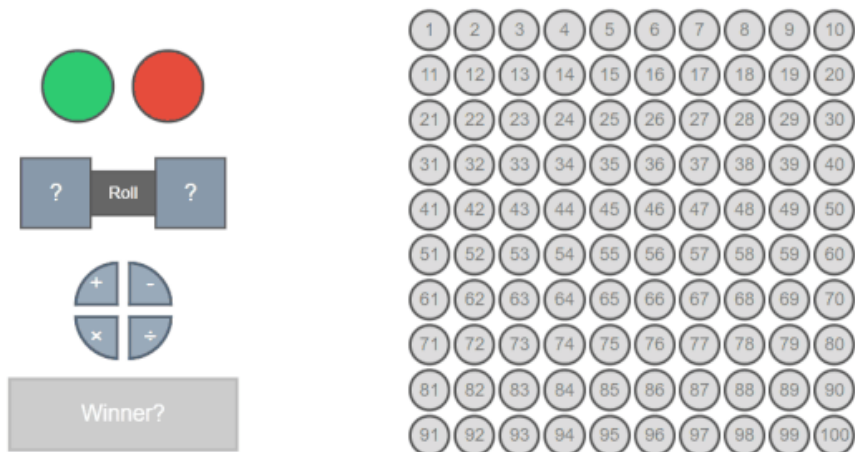
- در صورتی که بازیکن پیش از انداختن تاس‌ها در نوبت خود عملگری را انتخاب کند، هیچ اتفاقی نمی‌افتد.



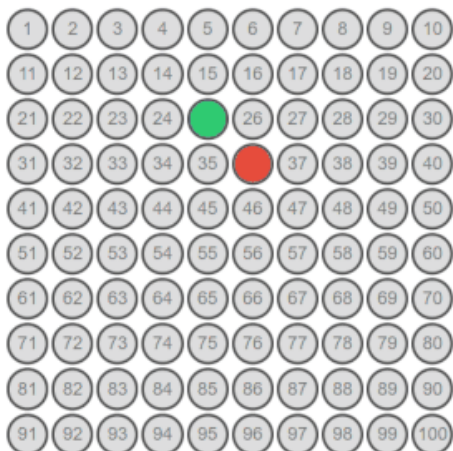
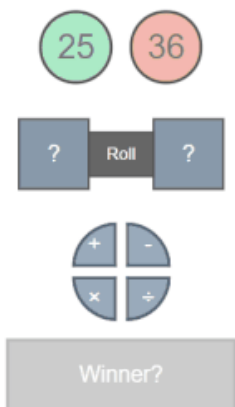
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

- در این بازی انتقال می‌تواند مثبت، منفی، یا صفر باشد؛ یعنی بازیکن می‌تواند علاوه بر حرکت به جلو، با گرفتن خروجی صفر از عملیات، در خانه‌ی فعلی خود بماند، و با دریافت خروجی منفی از عملیات، به عقب برگردد.
- در صورتی که مهره‌ی بازیکن در جایگاه شروع باشد، برای ورود به جدول باید عددی را از عملیات ریاضی روی اعداد دو تاس خروجی بگیرد که غیر اول، *بزرگتر از صفر، و

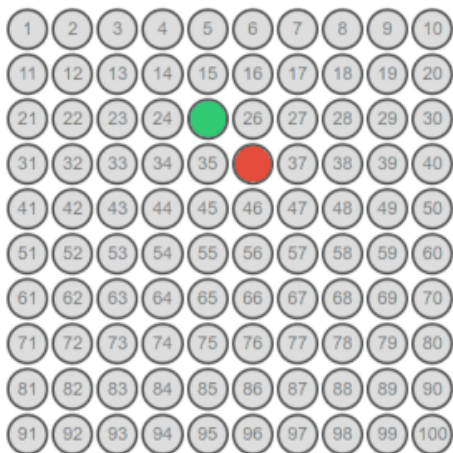
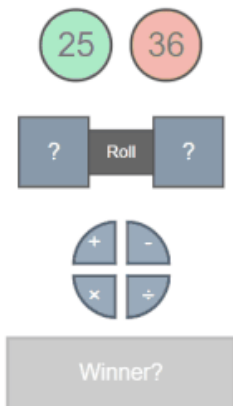
*تعریف شده باشد. در مثال زیر، ابتدا مهره‌ها در جایگاه شروع قرار دارند.



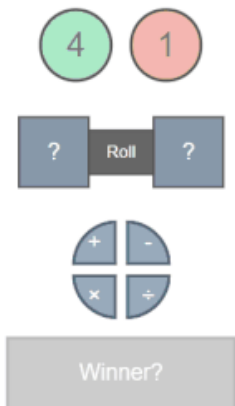
- در صورتی که بازیکن در نوبت خود عملگر را به نحوی انتخاب کند که او را به خانه‌ای با شماره‌ی اول ببرد، مهره‌اش به جایگاه شروع بازگردانده می‌شود. در مثال زیر، ابتدا مهره‌ی سبز در خانه‌ی 25 ، و مهره‌ی قرمز در خانه‌ی 36 قرار دارد.



- در صورتی که بازیکن در نوبت خود عملگر را به نحوی انتخاب کند که حاصل عملیات تعریف نشده باشد، مهره‌اش به جایگاه شروع بازگردانده می‌شود. در مثال زیر نیز ابتدا مهره‌ی سبز در خانه‌ی 25 ، و مهره‌ی قرمز در خانه‌ی 36 قرار دارد.

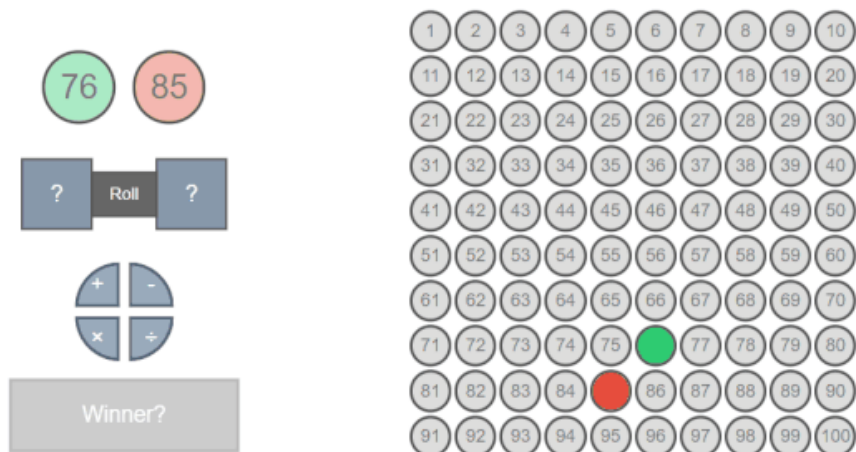


- در صورتی که بازیکن در نوبت خود عملگر را به نحوی انتخاب کند که او را به خانه‌ای با شماره‌ی کوچکتر از یک یا بزرگتر از صد ببرد، مهره‌اش به جایگاه شروع بازگردانده می‌شود. در مثال زیر، ابتدا مهره‌ی سبز در خانه‌ی 4 ، و مهره‌ی قرمز در خانه‌ی 1 قرار دارد.

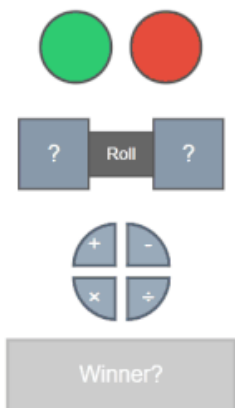


<div></div>	2	3	<div></div>	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

در مثال زیر، ابتدا مهره‌ی سبز در خانه‌ی 76 ، و مهره‌ی قرمز در خانه‌ی 85 قرار دارد.

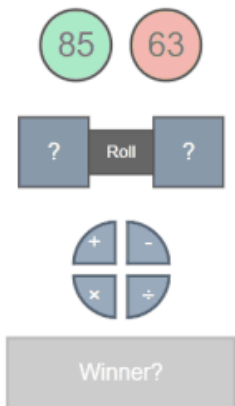


- در صورتی که بازیکن وارد خانه‌ای شود که حریف در آن قرار دارد، حریف را از آن خانه بیرون می‌کند، و به جایگاه شروع بازمی‌گرداند. در مثال زیر، ابتدا هر دو مهره در جایگاه شروع قرار دارند.



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

- در صورتی که یکی از مهره‌ها به خانه‌ی شماره‌ی 100 برسد بازی تمام می‌شود؛ نمایشگر پیام به رنگ بازیکن برنده در می‌آید، و پیام تبریک در آن نشان داده می‌شود؛ و دکمه‌ها دیگر کار نمی‌کنند. در مثال زیر، ابتدا بازیکن سبز در خانه‌ی 85 ، و بازیکن قرمز در خانه‌ی 63 قرار دارد.



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

فایل‌های اولیه

فایل‌های اولیه را از [اینجا](#) دانلود کنید. ساختار این فایل‌ها به شرح زیر است:

```
initial.zip
├─ avoid_primes.html
├─ avoid_primes.js
├─ jquery.min.js
├─ roll_dice.js
└─ style.css
```

- برای درست دیده شدن علامت‌های ریاضی، بهتر است از مرورگر کروم برای دیدن محتوای `avoid_primes.html` استفاده کنید.

جزئیات

با توجه به محتوای فایل‌های اولیه، به توضیح وظایفی که باید پیاده‌سازی شوند می‌پردازیم. در هر قسمت، موارد مهمی که شما باید پیاده‌سازی کنید **مشخص** شده‌اند.

- فایل `avoid_primes.html` : عناصر موجود در بدنه‌ی این فایل به شرح زیر هستند.

```
1 <body>
2   <div id="panel">
3     <div id="start_player_1" class="filled"><span>1</span></div>
4     <div id="start_player_2" class="filled"><span>2</span></div>
5     <div id="first_die" class="dice"><span>?</span></div>
6     <div id="second_die" class="dice"><span>?</span></div>
7     <button id="roll">Roll</button>
8     <button id="sum" class="operator">+</button>
9     <button id="subtract" class="operator">-</button>
10    <button id="multiply" class="operator">x</button>
11    <button id="divide" class="operator">÷</button>
12    <div id="msg" class="no_winner"><span>Winner</span></div>
13  </div>
14  <div id="board"></div>
15 </body>
```


- عنصر `#panel` : این عنصر صرفاً به عنوان دربردارنده‌ی اجزای بازی، به جز جدول، استفاده شده است.

- عنصر `#start_player_1` : این عنصر جایگاه شروع برای بازیکن سبز است؛ در صورتی که بازیکن سبز در حالت شروع باشد، دارای کلاس `.filled` خواهد بود و هیچ شماره‌ای داخل آن‌ها نشان داده نمی‌شود؛ در صورتی که مهره‌ی سبز داخل جدول باشد، دارای کلاس `.empty` خواهد بود و شماره‌ی خانه‌ی فعلی بازیکن در آن نشان داده می‌شود. برای نمایش شماره، عدد مورد نظر در داخل `` ای که فرزند جایگاه شروع است قرار می‌گیرد. اگر بازیکن در حالت شروع باشد، نوشته‌ی درون `span` برابر با `' '` خواهد بود.

- عنصر `#start_player_2` : این عنصر جایگاه شروع برای بازیکن قرمز است. توضیحات مربوط به `#start_player_1` به طور مشابه برای این عنصر نیز صادق است.

- عناصر `dice` : این دو عنصر تاس‌های بازی را نشان می‌دهند. تاس اول با شناسه‌ی `#first_die` ، و تاس دوم با شناسه‌ی `#second_die` نشان داده می‌شود. عدد روی تاس، در داخل `` قرار می‌گیرد. در حالتی که عدد روی تاس معلوم نباشد، یعنی نوبت بازیکن انجام شده باشد، ؟ جای آن را می‌گیرد.
- عنصر `#roll` : با کلیک روی این دکمه، در صورتی که بازیکن در نوبت خود تاس را نینداخته باشد، محتوای جدید در داخل تاس‌ها به نمایش در می‌آید. هر بار برای انداختن تاس از تابع `roll_dice()` که در ادامه توضیح داده شده است استفاده می‌شود.
- عناصر `.operator` : در صورتی که بازیکن تاس انداخته باشد اما هنوز عملیات مورد نظر خود را انتخاب نکرده باشد، با کلیک روی هر کدام از این دکمه‌ها، عملیات مربوط به آن روی عملوندها صورت می‌گیرد. به این ترتیب نوبت بازیکن انجام می‌شود، و باید موقعیت او با توجه به خروجی عملیات به‌روزرسانی شود.
- عنصر `#msg` : این عنصر، همان نمایشگر پیام است. در صورتی که بازی هنوز در جریان باشد، کلاس `.no_winner` ، و در صورت برد هر کدام از بازیکن‌های سبز، و قرمز به

ترتیب کلاس‌های `.winner_msg_1` ، و `.winner_msg_2` را خواهد داشت. هم‌چنین پیام مربوط به بازی، در داخل `` نمایش داده می‌شود که اگر بازی هنوز در جریان باشد، پیام `Winner?` ، و اگر به اتمام رسیده باشد، پیام `Congratulations!` را خواهد داشت.

- عنصر `#board` : این عنصر، جدول بازی است. خانه‌های جدول در فایل اولیه‌ی `avoid_primes.js` به آن پیوند می‌خورند.
- فایل `avoid_primes.js` : با داشتن محتوای اولیه‌ی این فایل، خانه‌های جدول در داخل `#board` قرار می‌گیرند. خانه‌ی `i` ام جدول، در حالتی که خالی باشد، به صورت زیر خواهد بود:

```
1 <div>
2   <span>i</span>
3   <span class="pawn"></span>
4 </div>
```

اگر مهره‌ی بازیکن سبز وارد خانه‌ای از جدول شود، فرزند دوم آن خانه، یعنی `.pawn` ، کلاس `.player_1.pawn` را به خود می‌گیرد. اگر بازیکن قرمز وارد خانه‌ای از جدول شود، همان عنصر، کلاس `.player_2.pawn` را به خود می‌گیرد. هر خانه، موقع خالی شدن کلاس `.pawn` را خواهد گرفت.

- فایل `roll_dice.js` : در این فایل تابع `roll_dice()` برای قرار دادن عدد تصادفی جدید در داخل تاس‌ها پیاده‌سازی شده است. این تابع دو عدد تصادفی بین 0 و 6 تولید می‌کند و در داخل تاس‌ها می‌نویسد.

- فایل `style.css` : در این فایل ویژگی‌های مربوط به کلاس‌ها و عناصر تعیین شده‌اند.

- توجه کنید که در بازی دو مهره‌ی سبز و قرمز وجود ندارند که جابه‌جا شوند؛ بلکه شما باید با تغییر صحیح کلاس در جایگاه‌های شروع و خانه‌های جدول حرکت آن‌ها را نشان دهید.

- در نهایت باید تمام قوانین بازی با توجه به جزئیات فنی توضیح داده شده پیاده‌سازی شوند.

نکات

- در داوری، سناریوهای مختلفی از بازی اجرا می‌شوند، و موقعیت مهره‌ها، وضعیت جدول، پیام بازی، و عملکرد دکمه‌ها پس از هر سناریو مورد ارزیابی قرار می‌گیرند.
- توجه کنید که داوری خودکار بر مبنای کلاس و شناسه‌ی عناصر که در فایل‌های اولیه داده شده است، انجام می‌شود؛ بنابراین در هر مورد باید دقیقا همان اسامی گفته‌شده را به عناصر اختصاص دهید.
- محتوای اولیه‌ی فایل `avoid_primes.js` و نیز سایر فایل‌ها را به هیچ عنوان تغییر ندهید. تنها کد خود را به آن اضافه کنید.
- توجه کنید که برای انداختن تاس باید **حتما** از تابع گفته‌شده استفاده کنید؛ زیرا این تابع، در همان فایل، در داوری با اندکی تغییرات استفاده خواهد شد.

- پروژه را با ساختار زیر ارسال کنید.

```
[your-zip-file-name].zip  
└─ avoid_primes.js
```