

در این تمرین هدف اصلی افزایش سرعت یادگیری و قدرت پیاده سازی شما با توجه به چیز هایی که یاد گرفتید، می باشد.

تاکنون شما به ساختار قالب بندی صفحات وب و برنامه نویسی به زبان جاوا اسکریپت آشنایی پیدا کردید. حال از شما خواسته می شود در مدتی مشخص یک ابزار گرافیکی برای پیاده سازی یک شبیه سازی انتخاب کرده و فقط بخش هایی از آن را فرا بگیرید که به آن نیاز دارید، نه کل توانایی ها و ماهیت آن ابزار.

نکته بسیار مهم در این تمرین است آن است که شما کاملاً در انتخاب ابزار و روش های پیاده سازی و سرویس هایی که می خواهید از آن استفاده کنید آزاد هستید. گرچه در انتهای توضیحات روش های پیشنهادی به اندازه مناسب توضیح داده می شوند.

مرتب سازی حبابی

این نوع مرتب سازی یکی از ساده ترین ها است که می توان خیلی سریع آن را پیاده کرد. به همین دلیل برای اینکه شما درگیر مباحث الگوریتمی نشوید این مرتب سازی انتخاب شده است. هدف اصلی شما از این تمرین شبیه سازی گام های مرتب سازی حبابی است. نکته قابل توجه آن است که قسمت انیمیشن در این تمرین جز بخش های اصلی نبوده و فقط نمره اضافی دارد.

دریافت اطلاعات

در تمرین قبل با نحوه فرستاد درخواست برای یک وب سرویس (API) آشنا شدید. این بار برای مرتب سازی خود از اعداد یک وب سرویس مرتبط با اقتصاد استفاده کنید.

نحوه شبیه سازی

شبیه سازی شما باید شامل یک سری مستطیل باشد. با توجه به اعدادی که دریافت می کنید مستطیل ها به وجود بیایند و برای شبیه سازی آماده شوند. سپس با شروع عمل شبیه سازی مستطیل ها با توجه به الگوریتم مرتب سازی ای که پیاده می کنید جا به جا شوند.

گام های مرتب سازی

برای به شبیه سازی گام به گام به صورت کلی راه حل های زیادی وجود دارد. در اینجا دو راه را پیشنهاد می دهیم. گرچه شما آزادید هر روشی که صحت عملکرد داشته باشد را انتخاب و از آن استفاده کنید.

روش پیش پردازش عمل ها: در این روش شما الگوریتم را به صورت کامل اجرا می کنید و گام هایی که اتفاق افتاد را ذخیره می کنید. به طور دقیق تر برای این الگوریتم شما نیاز به یک پشته دارید و هر بار که دو عنصر را جا به جا می کنید مقدار اندیس های آن را در پشته ذخیره می کنید. سپس با توجه به ابزار گرافیکی در هر بار اجرا (render یا draw) کردن روی اعضای پشته پیمایش کنید.

پیاده سازی در جاوااسکریپت:

```
let swaps = []
let arr = [254, 123, 301, 992, 510, 111, 209, 173, 170, 200]

for(let i = 0; i < arr.length - 1; i++) {
  for(let j = 0; j < arr.length - i - 1; j++) {
    if (arr[j] > arr[j+1]) {
      let temp = arr[j]
      arr[j] = arr[j+1]
      arr[j+1] = temp

      swaps.push([j, j+1])
    }
  }
}
```

روش بازدهی لحظه ای (اسمش من در آوردم): تکنولوژی های برتر مثل Python و JavaScript چند وقتی هست که از مفهوم بازده استفاده می کنند. کلیدواژه آن در هر دو زبان برنامه نویسی yield می باشد ولی از نظر دستور (Syntax) باهم برابر نیستند. ولی ما تنها نیاز به استفاده از آن در جاوا اسکریپت داریم. اگر تا به حال با ابزار اشکال یابی یک زبان برنامه نویسی متکی بر مفسر ها (Debugging) کار کرده باشید اگر برای مثال در یک حلقه Break Point مشخص کنید. در هر بار از اجرای آن حلقه ایست می کند و منتظر دستور شما می شود. مفهوم yield دقیقاً مثل Break Point می باشد ولی در تابع ها تعریف می شود و مقداری را بر می گرداند و جایی که برگرداندن را انجام داد ایست می کند تا برای بار دوم تابع را صدا بزنیم.

پیاده سازی در جاوا اسکریپت:

```
function* count() {  
  for(let i = 0; i < 3; i++) {  
    yield i  
  }  
}  
  
let c = count()  
console.log(c.next().value) // 0  
console.log(c.next().value) // 1  
console.log(c.next().value) // 2  
console.log(c.next().value) // undefined
```

برای استفاده از این مفهوم در مرتب سازی می توانید از کد زیر استفاده کنید.

```
let arr = [254, 123, 301, 992, 510, 111, 209, 173, 170, 200]  
  
function* bubbleSort(arr) {  
  for(let i = 0; i < arr.length - 1; i++) {  
    for(let j = 0; j < arr.length - i - 1; j++) {  
      if (arr[j] > arr[j+1]) {  
        let temp = arr[j]  
        arr[j] = arr[j+1]  
        arr[j+1] = temp  
  
        yield [j, j+1]  
      }  
    }  
  }  
}
```

برای استفاده از آن هم می توانید اینگونه عمل کنید:

```
let algo = bubbleSort()  
algo.next().value // the value is [j, j+1]
```

نمایش گرافیکی

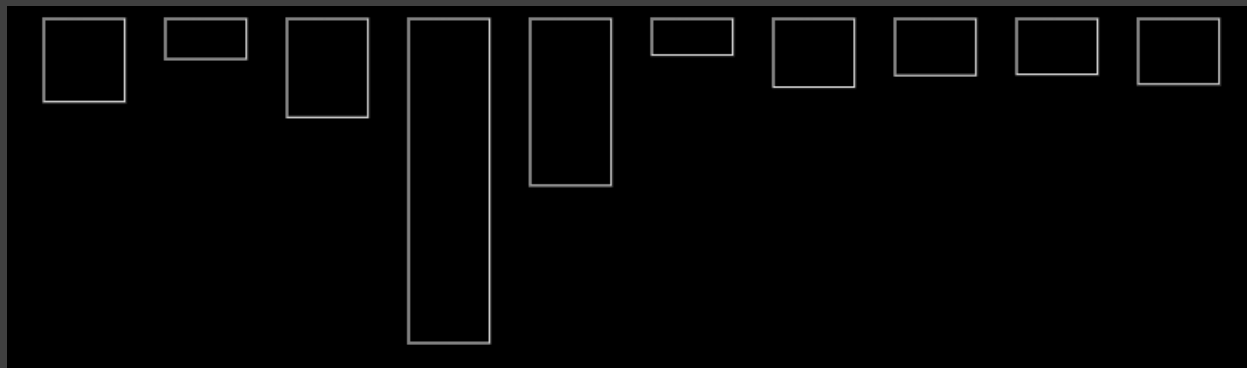
برای پیاده سازی بستری برای نمایش یک سری مستطیل آزادید از هر روشی که می خواهید استفاده کنید. برای مثال می توانید از تگ های HTML استفاده کنید و در فایل CSS کلاسی برای مستطیل ها با border مشخص کنید. ولی پیشنهاد ما استفاده از ابزار canvas در HTML5 می باشد که می توان آن را در JavaScript مدیریت کرد. این روش پیچیدگی ها را به زبان JavaScript خلاصه می کند و شما مجبور به درگیری با کم و زیاد کردن عناصر در صفحه وب نمی شوید.

برای مشاهده صفحه رسمی مربوط به ابزار گرافیکی معروف به این [لینک](#) مراجعه کنید.

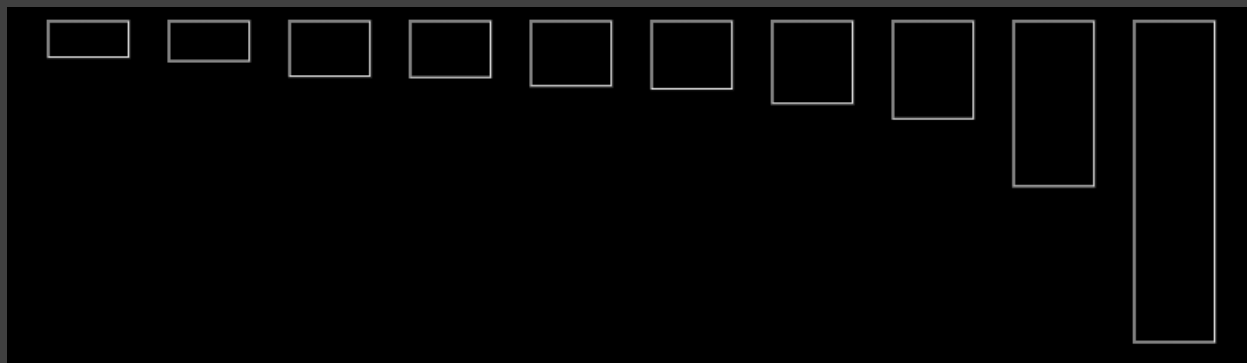
به طور خاص پیشنهاد من استفاده از [p5](#) می باشد. زیرا بسیار کامل و ساده است.

خروجی کار شما باید مثل تصویر های زیر باشد و تک تک گام های مرتب سازی را اجرا کند. برای قابل تشخیص شدن می توانید یا یک دکمه برای اجرا کردن گام به گام مشخص کنید یا با یک فاصله زمانی ثابت (برای مثال نیم ثانیه) گام به گام اجرا کنید.

حالت اولیه



حالت نهایی



در این [پیوست](#) آموزش ابتدایی p5 تدارک دیده شده.

داده های مورد نیاز

شما برای این شبیه سازی نیاز به یک سری عدد دارید تا آن ها را مرتب و شبیه سازی کنید. این اعداد باید توسط یک وب سرویس دریافت و سپس در یک آرایه ذخیره شوند. سپس می توانید بنا بر استراتژی خودتان نسبت به شبیه سازی و دیگر کار ها اقدام کنید. صرفاً از یک وب سرویس ۱۰ عدد دریافت کنید و آن ها را به صورت گفته شده در بخش گرافیکی به نمایش در بیاورید. بهتر است این ده عدد مرتبط با هم باشند. برای مثال می توانید قیمت سکه طلای تمام بهار آزادی را ده روز گذشته در نظر گرفته و بیشترین قیمت در هر روز را استفاده کنید.

از معروف ترین وب سرویس های ارز دیجیتال می توان به [Messari](#) اشاره کرد.

پیشنهاد ما [سرویس داخلی نوسان](#) می باشد که هم کار با آن ساده بوده و نیاز به پرداخت هزینه ندارد. کافی است به ربات تلگرام با آی دی `navasan_contact_bot` پیام دهید و گزینه تست وب سرویس را بزنید. سپس کلید انحصاری شما برای شما فرستاده می شود و می توانید از آن استفاده کنید. راهنمای کامل استفاده از این سرویس در این [لینک](#) موجود می باشد. گرچه برای دریافت قیمت سکه تمام بهار آزادی در روز گذشته شما می توانید از الگوی زیر برای فرستادن درخواست استفاده کنید:

http://api.navasan.tech/ohlSearch/?api_key=key&item=bahar&start=1400-01-01&end=1400-01-02

که در این درخواست برای دریافت آرچو قیمت ها به کار می رود و پارامترهایی در آن مشخص شده اند:

- item: نام ارز مورد نظر ([لینک تمامی مقدار ممکن](#))
- start: تاریخ شروع (به شمسی یا معیار رسمی linux)
- end: تاریخ پایان (به شمسی یا معیار رسمی linux)

```
[
  {
    "timestamp": 1616272200,
    "date": "1400-01-01",
    "open": 10770000,
    "high": 10770000,
    "low": 10760000,
    "close": 10760000
  },
  {
    "timestamp": 1616358600,
    "date": "1400-01-02",
    "open": 10760000,
    "high": 10760000,
    "low": 10620000,
    "close": 10620000
  },
  ...
]
```

در انتها هم یک مثال علمی در این [لینک](#) مشاهده می کنید.

نحوه نمره دهی (از ۱۰۰ نمره)

- صحت خروجی کار ۳۰ نمره
- صحت الگوریتم ۱۰ نمره
- صحت به روز رسانی مستطیل ها ۳۰ نمره
- استفاده درست از وب سرویس ۳۰ نمره
- انیمشین ۳۰ نمره اضافی
- موارد اضافی قابل توجه نیز نمره اضافی دارند. (برای مثال استفاده از شی گرایی یا React)

موفق و پیروز باشید.

شایان بالایی

علیرضا جعفرتاش