

## P5JS

در این راهنما قصد داریم به صورت اجمالی استفاده از فریمورک P5 را شرح دهیم.

## نحوه استفاده در پروژه

اگر از NodeJS استفاده می کنید (React، Vue و ...) لطفا این [لینک](#) مراجعه کنید و می توانید این بخش را نادیده بگیرید.

ساختار پروژه شما شامل یک فایل با پسوند html می باشد. در این فایل باید اسکریپت اصلی p5 را درج کنید. که یا می توانید از لینک cdn استفاده کنید و یا آن رو دانلود کرده و به صورت محلی رو سیستم خود آن را لینک کنید. [آدرس اسکریپت](#)

پس در نهایت یکی از دو خط زیر باید در قسمت head فایل اصلی شما وجود داشته باشد.

```
<head>
  <script type="text/javascript" src="p5.min.js"></script>
  <script type="text/javascript" src="https://cdn.jsdelivr.net/npm/p5@1.3.1/lib/p5.js"></script>
</head>
```

سپس شما باید یک فایل جاوا اسکریپت را بعد از اینکه اسکریپت اصلی p5 را اضافه کردید به صفحه اصلی خود اضافه کنید و قسمت اصلی برنامه شما در این فایل انجام می شود.

```
<body>
  <script type="text/javascript" src="script.js"></script>
</body>
```

حالا در فایل script.js شما می توانید از امکانات این کتابخانه بهره ببرید.

## ساختار کلی

معماری کتابخانه p5 از نوع [ماشین حالت](#) است، به این گونه که برای نمایش خروجی باید دو تابع اصلی را مشخص کرد. این دو تابع setup و draw نام دارند.

- setup: تابعی است که در ابتدای کار یکبار اجرا می شود.
- draw: تابعی است که هر ثانیه به ۶۰ بار اجرا می شود.

دقت کنید عدد ۶۰ همان fps (frame per second) می باشد. که تعداد باری است که canvas به اصطلاح گرافیک کامپیوتری render می شود. این مقدار به صورت پیش فرض برابر ۶۰ بوده و قابل تغییر می باشد. تمامیت عملیات ها باید در تابع draw انجام شوند. البته می توان از ساختار هایی مثل شی گرای استفاده کرد.

برای هر بخش در صورت نیاز لینکی به کدی برای مثال آن درج شده که تمامی آنها در [اینجا](#) قابل مشاهده هستند.

تابع `setup`

در این تابع معمولاً اندازه canvas مشخص می شود.

```
function setup() {
  createCanvas(400, 400)
}
```

تابع `draw`

در این تابع هدف اصلی پیاده می شود. برای مثال اگر بخواهیم یک مستطیل ایجاد کنیم:

```
function draw() {
  background(220)
  rect(x, 200, 50, 50)
  x += 1
}
```

دقت کنید که همواره باید تابع `background` را در ابتدا صدا زد، این تابع یک مستطیل هم رنگ با آرگومانش روی تمام اشکال کشیده شده در فریم قبلی می کشد و به ما اجازه می دهد فریم جدید را ایجاد کنیم. این تکنیک دقیقاً همان تکنیکی است که در سینما به کار گرفته می شد. برای نمایش دادن یک فیلم عکس هایی پشت سر هم به نمایش در می آمدند تا به نظر پیوسته باشند.

**تابع `rect`** یک مستطیل ایجاد می کند. اما کجا؟ محور canvas و در کل هر چیزی که با استاندارد گرافیک کامپیوتری کار کند دقیقاً قرینه محور  $y$  های حالت عادی است. یا به طور دیگر سمت چپ ترین و بالاترین پیکسل در canvas نقطه مرکزی با مختصات  $(0, 0)$  می باشد و هر چه در راستای محور  $x$  به راست برویم مولفه اول مختصات یا همان  $x$  افزایش پیدا کرده و اگر به چپ برویم کاهش می یابد. (دقیقاً مثل حالت عادی) اما اگر در راستی محور  $y$  به پایین برویم مقدار مولفه دوم یا همان  $y$  افزایش پیدا کرده و در صورت بالا رفتن آن کاهش پیدا می کند.

این تابع در حالت عادی ۳ یا ۴ آرگومان می پذیرد. (اطلاعات دقیق تر در [مستندات](#) موجود است).

نقطه معرف یک مستطیل در گوشه بالایی سمت چپ آن قرار دارد و از آنجا رسم می شوند. یعنی در آرگومان این تابع که به ترتیب  $x$ ،  $y$ ،  $w$  و  $h$  می باشند. دو آرگومان اول مختصات نقطه ای هستند که مستطیل از آن جا رسم می شود. اگر پارامتر چهارم داده نشود یک مربع ایجاد می شود و عرض آن برابر با  $w$  می باشد. در صورت وجود پارامتر چهارم طول مستطیل در راستای محور عمودی برابر با آن می شود.

برای تست کردن این بخش به این [لینک](#) مراجعه کنید.