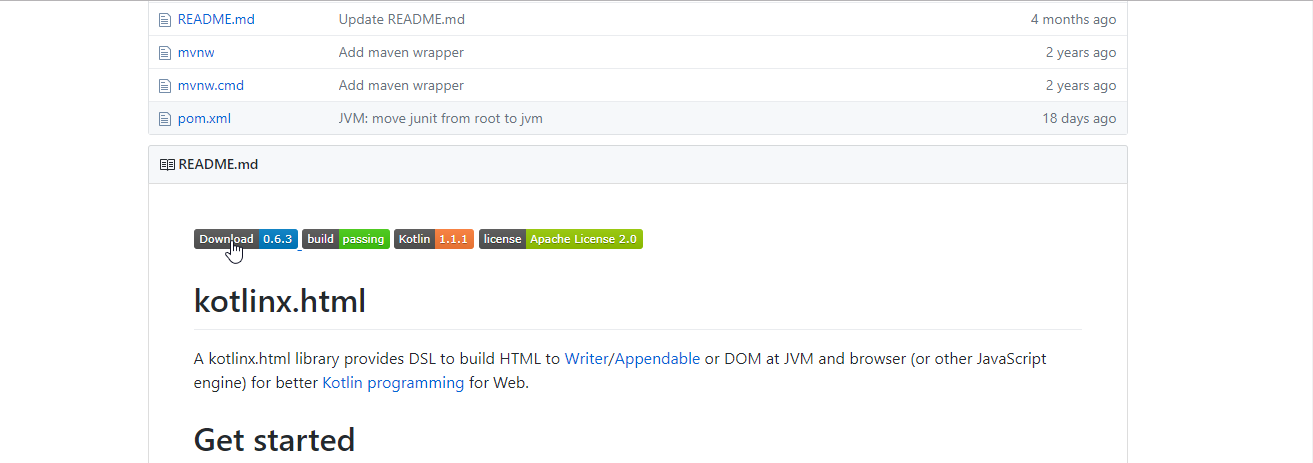
توی این قسمت میخوایم از روش دیگه­ای برای ارتباط با DOM استفاده کنیم که استفاده ازش میتونه راحت­تر باشه. برای این کار میخوایم از یک کتابخونه که شرکت JetBrains درست کرده استفاده کنیم. اسم اون کتابخونه هم kotlinx.html­هه.

لینک github این کتابخونه : [Kotlinx.html](https://github.com/Kotlin/kotlinx.html)

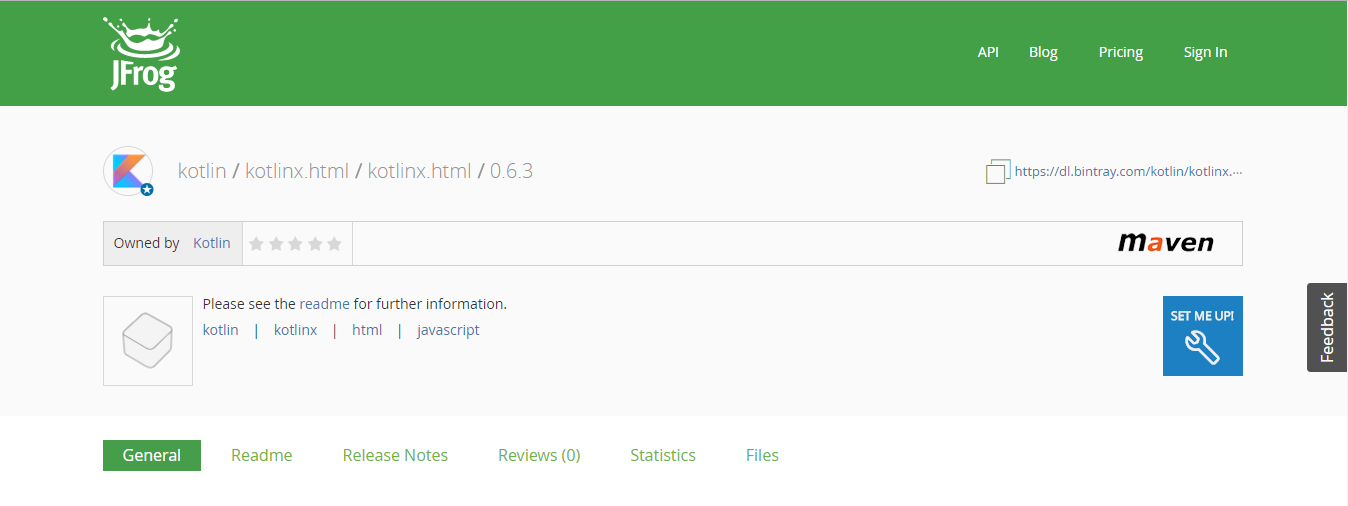
خب ما الان از هیچ سیستم مدیریت بیلدی استفاده نمیکنیم، و مستقیما با استفاده از خود IDE بیلد میکنیم. پس نیاز داریم که فایل \*.jar کتابخونه رو دانلود کنیم و توی پروژمون قرار بدیم.(روش ارائه شده ممکن درطول زمان تغییر کنه، اگه با مشکلی مواجه شدین مطلعمون کنین)

برای دانلود فایل \*.jar وارد صفحه github این کتابخونه میشیم و روی گزینه دانلود کلیک میکنیم.

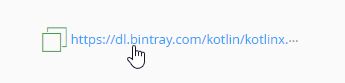
روی گزینه دانلود کلیک میکنیم و به یک صفحه دیگه منتقل میشیم.



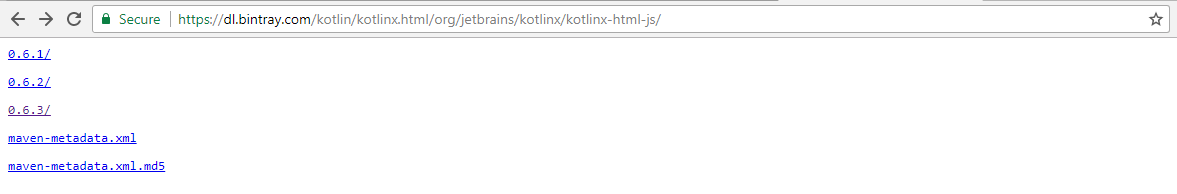
اگه بالا سمت راست دقت کنین متوجه یک لینک میشین، لینکی که در اصل با اون به صفحه بعد منتقل بشین.



بعد از این کافیه به مقصدی که توی شکل زیر میبینین برسین:

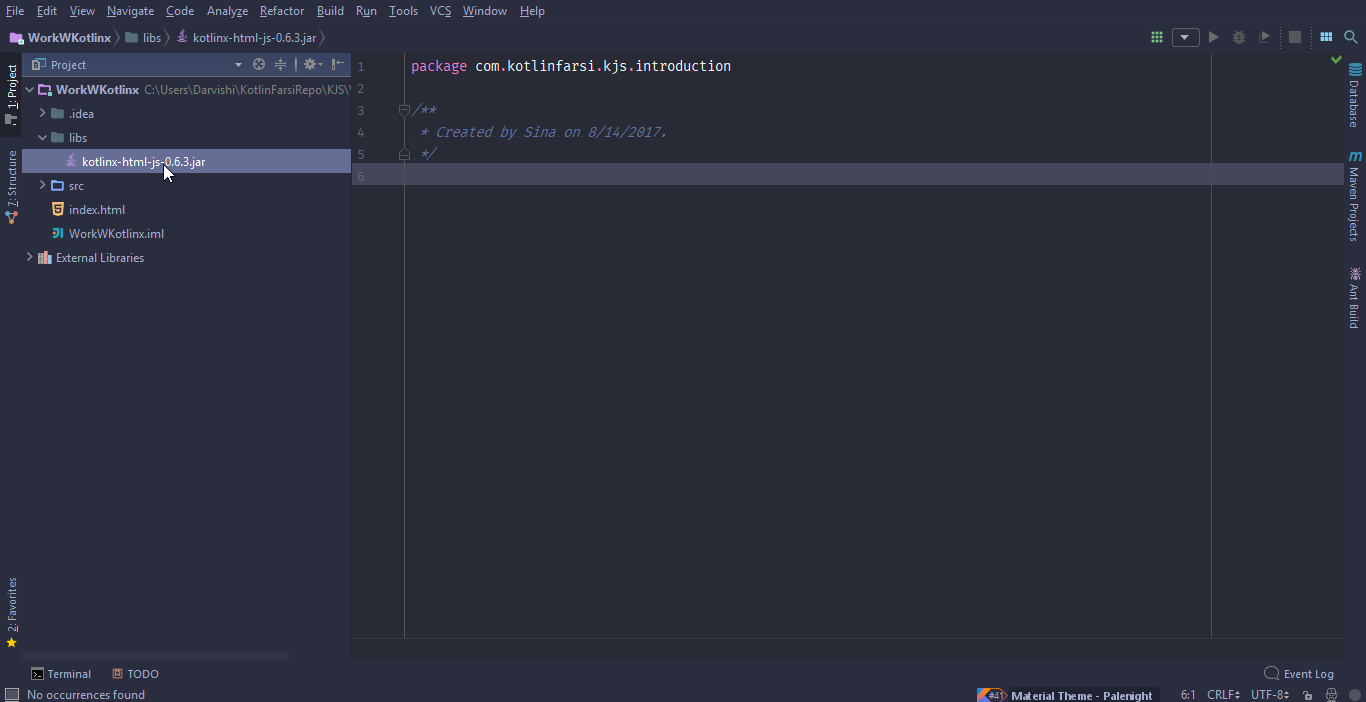


الان کافیه اخرین نسخه­ی موجود رو انتخاب کنین و بعدش توی صفحه­ای که ظاهر میشه فایل kotlinx-html-js-<latest-version>.jar رو انتخاب کنین، به عنوان مثال الان درحال حاظر که دارم این مقاله رو مینویسم اخرین نسخه 0.6.3 هه پس فایل kotlinx-html-js-0.6.3.jar رو انتخاب و دانلود می­کنم.

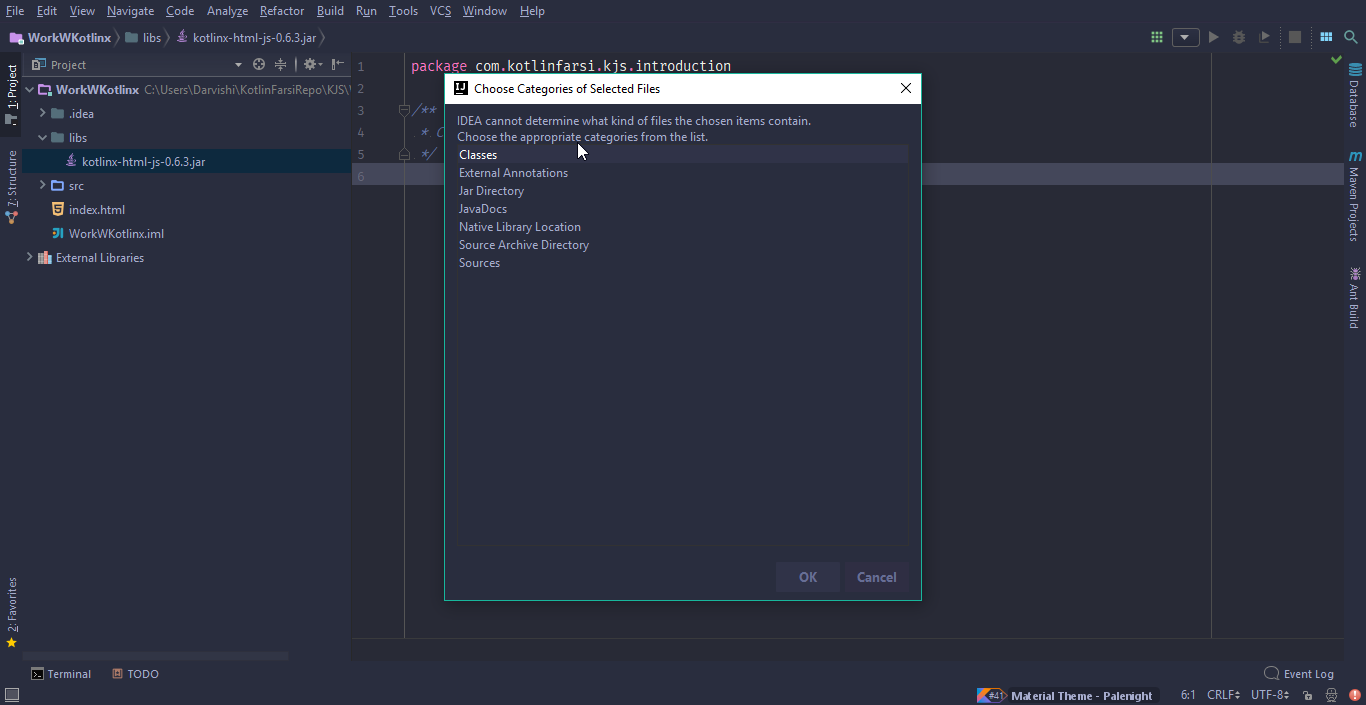
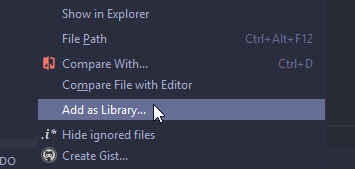


بعد از این که فایل رو دانلود کردیم. باید این فایل رو به عنوان یک کتابخونه به IDE معرفی کنیم. برای این کار باید یک فولدر توی مسیر root پروژه درست کنیم و فایل .jar رو داخلش بریزیم.

همینطور که میبینین یک فولدر باز کردیم به اسم libs و داخلش فایل کتابخونمون رو قرار دادیم.حالا کافیه اون فایلر رو به عنوان یک کتابخونه معرفی کنیم.برای این کار روی اون فایل کلیک راست کرده و گزینه add as library رو انتخاب میکنیم.

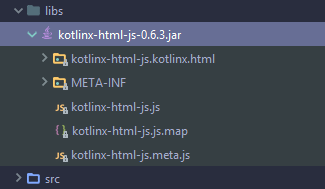


بعد از این که این گزینه رو انتخاب میکنیم، صفحه­ای ظاهر میشه به شکل زیر:



که باید بگیم این فایل jar حاوی چه چیزهاییه و ما با انتخاب classes میگیم که این فایل حاوی یک سری کلاسه و بعدش هم ok و دوباره ok و تمام.

و حالا اگه دقت کنیم توی قسمت Project شما میتونین فایل \*.jar رو به صورت زیر ببینین.



خب تا اینجا پروژه رو اضافه کردیم، حالا نوبت نوشتن کد توی فایل Main.kt فقط قبل از این که ادامه بدیم، بذارین فایل index.html رو نشون بدم تا مطمئن شیم همه چی درسته.

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Working With Kotlinx</title>  
</head>  
<body>

<div id="root"></div>  
<script src="out/production/WorkWKotlinx/lib/kotlin.js"></script>  
<script src="out/production/WorkWKotlinx/lib/kotlinx-html-js.js"></script>  
<script src="out/production/WorkWKotlinx/WorkWKotlinx.js"></script>  
</body>  
</html>

خط اول و دوم و سوم داخل body که تکراریه، اولی یک divهه که بهش ای دی root رو دادیم، دومی ادرس فایل \*.js تولیدی از Main­امونه و سومی هم فایل کتابخونه استاندارد کاتلینه. ولی خط چهارم چیه؟ درواقع خط چهارم هم فایل بیلد شده از کتابخونه­ایه که به پروژه اضافه کردیم و یادتون باشه که به این فایل برای اجرای برنامه­مون احتیاج داریم.

بریم سراغ main و شروع کنیم به کدزنی. میخوایم یک صفحه درست کنیم که توش یک نوشته داشته باشه و یک دکمه که وقتی روش کلیک میشه عملی رو انجام بده.کد زیر رو نگاه کنین:

fun main(args: Array<String>) {  
 val root = *document*.getElementById("root")  
  
 val div = *document*.*create*.*div***{** *h1* **{** + "Hello KotlinFarsi"  
 **}** *button* **{** + "Click Me"  
 *onClickFunction* = **{** *println*("Clicked Again")  
 **}  
 }  
 }** root?.appendChild(div)  
}

خط دوم خط آشناییه. در واقع داریم دنبال عنصری به id ی root میگردیم.خط چهارم داریم یک عنصر div میسازیم که داخلش حاوی یک h1 و یک buttonهه.اگه دوره مقدماتی کاتلین رو گذرونده باشین متوجه هستین که این عنصرها تنها یکسری توابع هستن. و هرچی داخلش مینویسم تنها توابع لاندایی هستن که به اون تابع پاس داده میشن یعنی h1 و button تنها یکسری توابع لاندایی هستن که به div پاس داده میشه. زمانی که داریم یک تکست رو به یک تابع پاس میده از تابع plus استفاده میکنیم که اینجا میتونین به صورت “+” ببینینش. و در انتها هم یک تابع به نام onClickFunction داریم که بهش یک تابع چاپ توی کنسول میدیم. این روشیه که ما رو قادر میکنه با DOM کارکنیم.

درواقع این چیزی که اینجا استفاده کردیم DSL صدا زده میشه که مخفف Domain Spcefic Languageهه.و در واقع این DSL ها داخل یک زبون استفاده میشنو انگار که داریم از یک زبون متفاوتی استفاده میکنیم. ولی واقعیت اینه که ما تنها یکسری تابع رو صدا میزنیم و یک ساختارهای متفاوت ندارن.