داکیومنت اموزش PWA

* 1. درس 7 Manifest.json

مانیفست به همراه SW کار تنظیمات ابتدای برنامه ما را انجام میدهد. در واقع در زمان ابتدای شروع برنامه SW تنظیمات اولیه از مانیفست می گیرد.



* 1. در 8 Manifest.json



* 1. درس 9 نصب شبیه ساز گوشی همراه
  2. درس 10 تنظیمات IOS



* 1. درس11 تنظیمات برای IE وEdge



* 1. درس 12 معرفی SW.js

در این جلسه کار با callback.js را کار کردیم. اینکه چطور به return داخل یک تابع دسترسی داشته باشیم. در شکل زیر ما به return داخل تابع دسترسی نداریم. یعنی خط consol اجرا نمیشه:



برای حل این مشکل از callback به صورت زیر استفاده می کنیم:



* 1. درس 13 مدیریت خطا ها در callback

برای کنترل خطا در callback از دو ورودی به ان استفاده کردیم. اگر بزرگتر از 10 بود خطا برگردد و در غیر این صورت ادامه دهد و در زمان فراخوانی با کنترل آرگومان اول این خطا را کنترل کریم:



* 1. درس 14 آشنایی با Promise

در این درس با Promise اشنا می شویم. در promise ابتدا یک کانستراکتور مشابه نام فایل js می سازیم که با دو ورودی کار می کند. شرط داخل تابع هم که مشخص ست. زمان فراخوانی تابع add اگر شرط درست باشده then و اگر اشتباه باشد catch اجرا می شود. با این روش از جهنم callback راحت می شویم و کنترل خطا رو راحتر انجام میدهیم.



* 1. درس 15 Promise.all

در مورد promise اول اینکه به صورت زیر می توان مدیریت خطا کرد و از شر جهنم callback راحت شد.



مورد بعدی هم promise all هست که در این مورد می توان یک تابع نوشت که دو ورودی داشته باشد به صورت ارایه بگیرد. دو تابع add,minus به عنوان ورودی به promiseall داده شدند. حالا then ,catch خروجی دو تابع را نشان می دهند(البته اگر هر دو resolve باشند. اگر یکی از توابع reject باشه، کلا catch نمایش داده میشه.). این مورد زمانی که بخواهیم خروجی 2 تا api را مقایسه کنیم به درد می خورد.



* 1. درس 16 بررسی Promis Race

در این دستور هر کدام از دوتابع که زودتر خروجی بدهند، خروجی ان نمایش داده می شود. در این شکل Add چون سریعتر انجام می شود خروجی ان نمایش داده می شود:



* 1. درس 17 XML Http Request

معرفی این سایت برای دریافت داده از api ها: <https://reqres.in/>

در فایل fetch.js به صورت زیر api را فراخوانی میکنیم. ابتدا یک کانکشن میسازیم. بعد کانکشن را باز می کنیم. نوع پاسخرا مشخص می کنیم. Onload برای این است که زمانی که کار گرفتن دیتا تمام شد. Onerroe برای زمان خطا و send دستور را اجرا می کند.



* 1. درس 18 معرفی و استفاده از Fetch

در این قسمت روش استفاده از fetch بیان شد. Fetch نوعی promise هست پس then,catch دارد. دلیل then صورتی رنگ اینه که پاسخ response.json خودش یک promise هست.



* 1. درس 19 نمایش اطلاعات API در PWA

در این قسمت اطلاعات رو از API گرفتیم و نمایش دادیم:





* 1. درس 20 لود کردن فایل با Fetch

در این قسمت سعی کردیم با fetch یک فایل html را بگیرم و نمایش بدهیم. این کار می توان برای نمایش یک المان مثل نوتیفیکیشن و متن و ... که می خواهیم در تمامی صفحات تکرار شود، استفاده کرد.







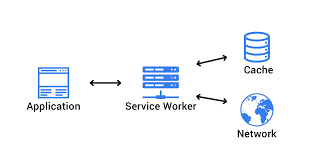
* 1. درس 21 Post و Put با Fetch

در این جلسه بقیه موارد API مثل Post رو بررسی کردیم:



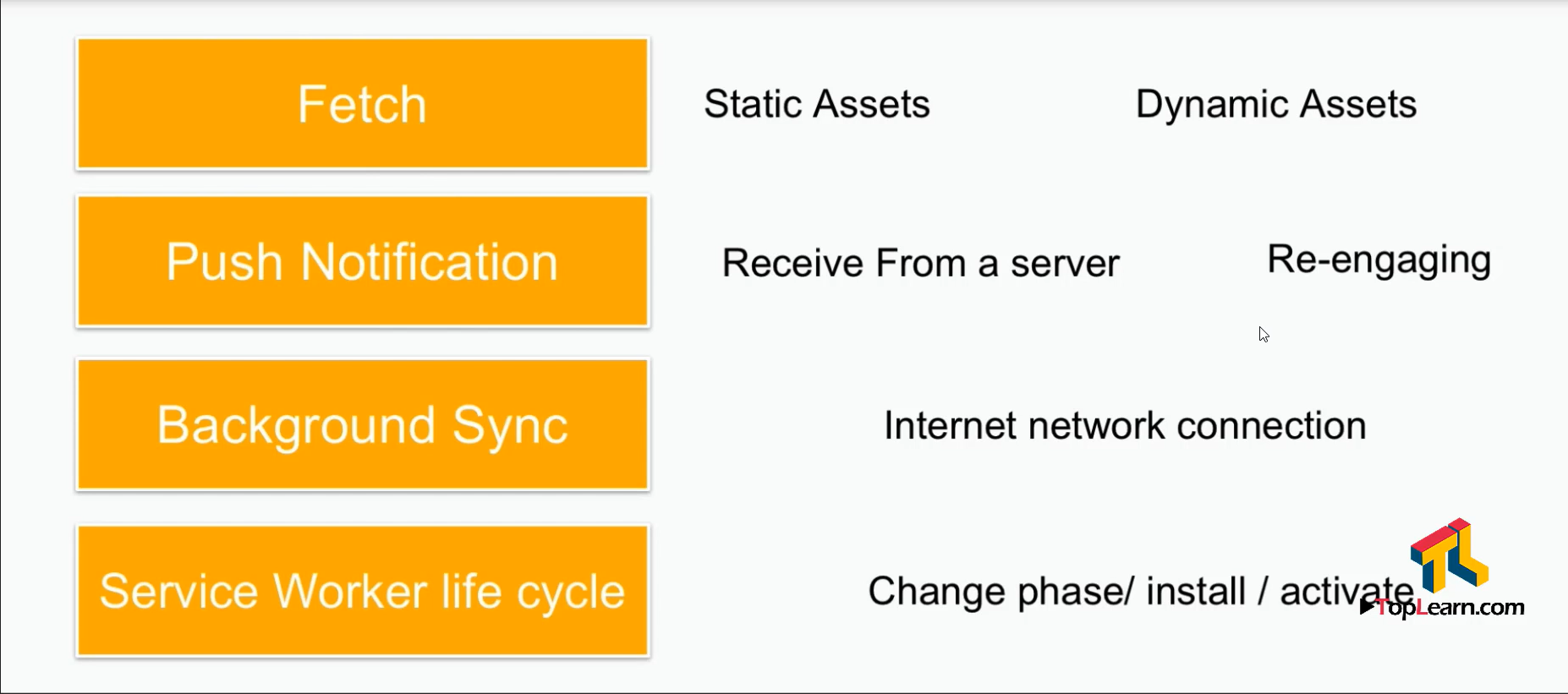
* 1. درس 22 Service Worker چیست؟

این مفهوم رابط بین application و وب و catch هست. این صفحه Script های صفحات را کنترل و ران می کند و مثل عکس به عنوان واسطه عمل می کند.



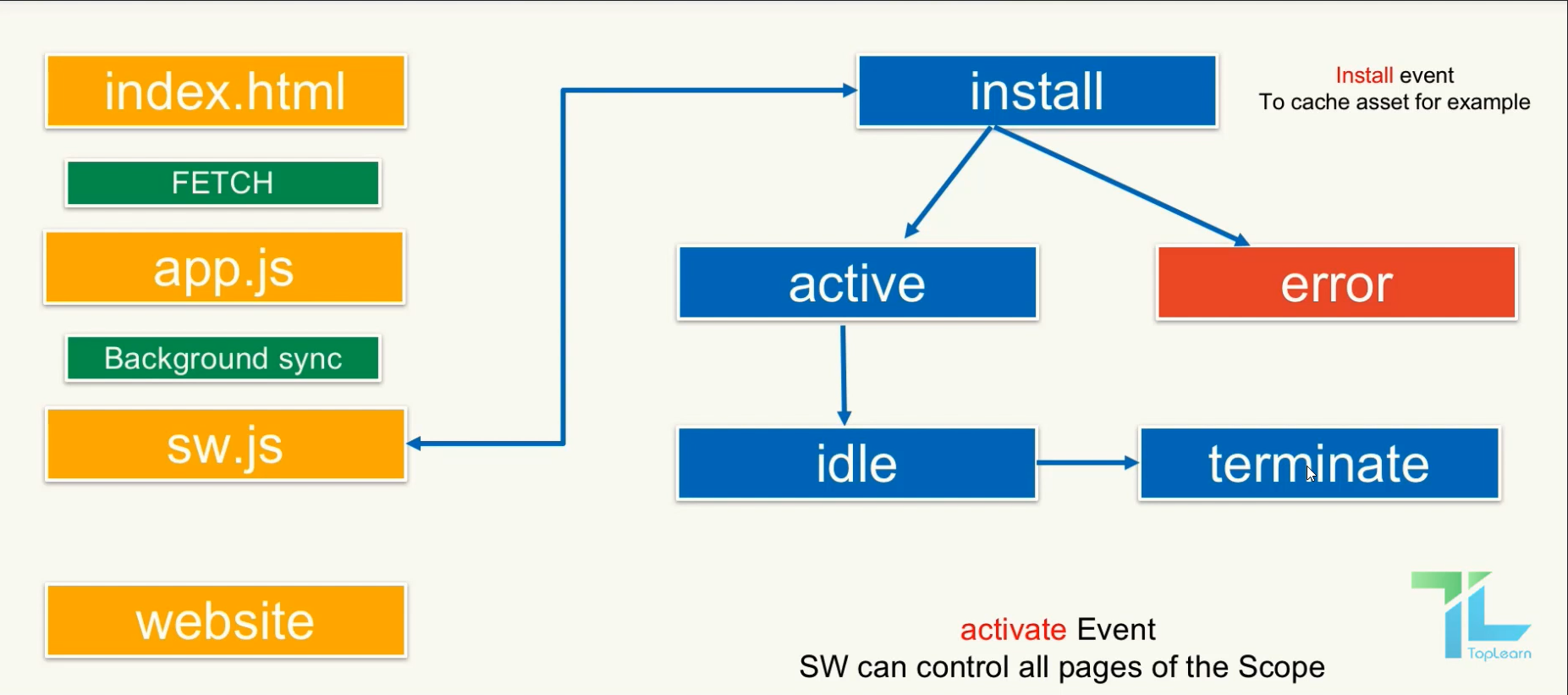
* 1. درس 23 ویداد های Servise Worker

در این جلسه رویداد های سرویس وروکر رو به صورت تئوری بررسی کردیم.



* 1. درس 24 چرخه حیات Servise Worker

در این جلسه چرخه کار سرویس ورکر به صورت تئوری بررسی شد.



* 1. درس 25 ساخت و install کردن Service Worker