1. 介绍：浏览器暴露给js的一个接口，可以通过这个接口查看用户访问网站的连接建立时间，dns的时间，使用该api需要在页面完全加载完成之后才能使用。最简单的方法是在window.onload事件中读取各种数据。
2. memory:浏览器内存情况

jsHeadSizeLimit:

totalJSHeapSize:表示当前js堆栈内存的总大小

usedJSHeapSize: 表示所有被使用的js堆栈内存

如果usedJSHeapSize> totalJSHeapSize表示可能出现了内存泄漏

1. navigation：网页导航相关：

redirectCount:重定向属性--只读。表示当前页面是几次重定向过来的。有同源策略限制。

type:返回值是0,1,2中的一个；分别对应：TYPE\_NAVIGATE(用户通过常规导航方式访问界面，比如点击一个连接，或者一般的get方式)

TYPE\_RELOAD:用户通过刷新，包括js调用刷新接口等方式访问界面。

TYPE\_BACK\_FORWORD:用户通过后退按钮访问本页面。

1. Timing组合值的意义--里面的时间指的是时间戳
2. connectEnd: 1648955804742 //
3. connectStart: 1648955804742
4. domComplete: 1648955826675
5. domContentLoadedEventEnd: 1648955805480
6. domContentLoadedEventStart: 1648955805480
7. domInteractive: 1648955805243
8. domLoading: 1648955804838
9. domainLookupEnd: 1648955804742 //dns查询耗时
10. domainLookupStart: 1648955804742 //dns查询耗时
11. fetchStart: 1648955804742
12. loadEventEnd: 1648955826676
13. loadEventStart: 1648955826676
14. navigationStart: 1648955804733
15. redirectEnd: 0
16. redirectStart: 0
17. requestStart: 1648955804742
18. responseEnd: 1648955804799 //request请求耗时
19. responseStart: 1648955804742 //request请求耗时
20. secureConnectionStart: 0
21. unloadEventEnd: 0
22. unloadEventStart: 0

request请求耗时：responseEnd- responseStart

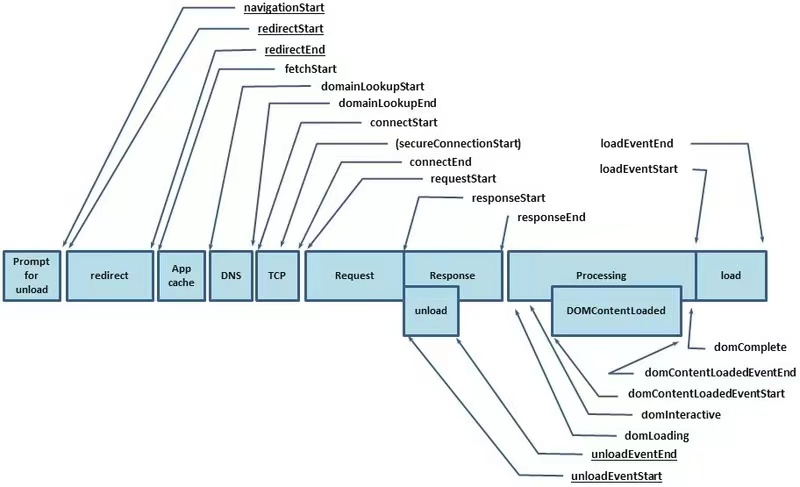
responseStart--navigationStart：白屏时间。

onload时间：loadEventEnd-navigationStart:也是onload回调函数执行的时间。

domready时间：domContentLoadedEventEnd-navagationStart

tcp耗时：connectEnd-connectStart

解析dom树耗时：domComplete-domInteractive



1. now()返回一个当前页面执行时间的时间戳

用来计算程序的执行时间

performance.now()

doSomething()

performance.now()

前后的结果相减就是doSomething执行的时间

1. mark()自定义添加标记的时间。

performance.mark("hello-start") //执行前做个标记

console.log("oooooo")

performance.mark("hello-end") //执行后做个标记

var name = "myMeasure"

performance.measure(name, "hello-start", "hello-end") //储存程序执行的时间

performance.getEntriesByName(name) //获取某个标记的程序的执行时间