



НИ Е ВЪРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Page Load Ime



Зареждане и обработка

- За да покаже една страница в крайният ѝ вид, браузъра минава през няколко основни етапа:
 - теглене на HTML документа (download)
 - прочитане на **head** частта и изтегляне на всички линкове в head-a (пак download)
 - парсване (обработка, идва от думата **parse**) на всичко в head-a: CSS файлове, фонтове, генериране на CSSOM-a, изпълняване на скриптовете, генериране на BOM (**window** обекта)
 - прочитане на **body**
 - парсване на html-a и генериране на DOM-a (**document** обекта), теглене на допълнителни скриптове и картинки, генериране на CSSOM-a
 - визуализация (чак след като всичко по-горе е изпълнено)

Good to know..

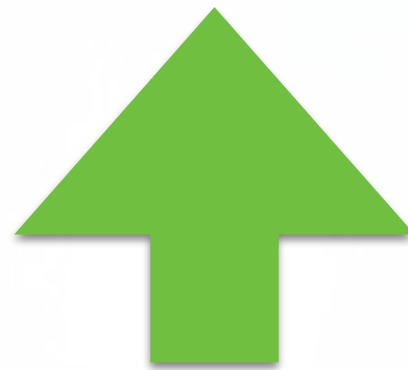
- Скоростта на зареждане е критична за потребителя
 - ако страницата се забави над 10секунди, много от потребителите се отказват да я чакат)
- Броят и видът на ресурсите, които страницата зарежда допълнително (фонтове, css, скриптове, картинки) са критични за времето за зареждане на страницата
- Начинът и последователността на включване на ресурсите може много да подобри performance-a на страницата
 - no `<script>` in the head!
 - or: **async/defer** the scripts (not fully supported so avoid)
 - generate dynamically all extra content after DOM ready

First things - first!

- Когато искаме страницата ни да се зареди бързо, трябва много внимателно да си организираме линковете и скриптовете
- Прието е всичко, което е необходимо за генерирането на визията на страницата, да се зарежда най-напред
- Това са CSS, fonts, background картинки и евентуално някои скриптове, от които зависи показването на съдържание
- Зареждането на всичко останало се отлага, за да може потребителя да не усети забавяне в зареждането на страницата

Златното правило:
style - горе ; script - долу

CSS



JS

Заклучение

- Запомнете, че браузърът винаги зарежда линковете в head-а преди всичко останало
- Запомнете също, че скриптовете, които са там, ги тегли и ги изпълнява още преди да е парснат html кода !!
(т.е. преди да е заредил DOM-а в document обекта)
- Запомнете, че визуализацията на страницата се случва едва след като е зареден DOM-а
- И не забравяйте, че скоростта на зареждане и показване (визуализация) на страницата е критична!

За четене

- Performance by Google: <https://developers.google.com/web/fundamentals/performance/?hl=en>
- Optimizing the Critical Rendering Path: <http://www.sitepoint.com/optimizing-critical-rendering-path/>
- Малко по-авансeд: <http://devnet.kentico.com/articles/optimizing-page-load-time> (статията е от 2011!)
- Fast enough: <http://www.vinta.com.br/blog/2015/web-page-optimization.html>
- async и defer: <http://peter.sh/experiments/asynchronous-and-deferred-javascript-execution-explained/>

Въпроси?

MAKE ALL EXAMPLES



Задача

<http://zenlabs.pro/sa-fe/lessons/lesson16/task/>

Полезни връзки

- Статията за `<noscript>`:
<https://webdesign.tutsplus.com/tutorials/quick-tip-dont-forget-the-noscript-element--cms-25498>
- Решенията на домашно 15:
<http://codepen.io/jenie/pen/yOWmNz?editors=0012>
<http://codepen.io/jenie/pen/zqQgBJ>
- Решенията на задачата от контролното:
<https://repl.it/CSam>



**KEEP
CALM
AND
LEARN
JAVASCRIPT**

Примери

<http://zenlabs.pro/courses/lessons/lesson16/examples.zip>

Домашно

<https://github.com/zzeni/swift-academy-homeworks/blob/fe-03/tasks/L16>