



НИ Е ВЪРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Преговор



(и подготовка за въпроси от интервю)

Разликата между HTML5 и HTML4 ?

- html5 е новият стандарт, който се поддържа от всички модерни браузери
- изисква DOCTYPE декларация в началото на документа: `<!DOCTYPE html>`
- изисква meta charset
- подобрена семантика! (section, article, section, aside, header, footer, nav)
- въвежда много нови елементи и атрибути (http://www.w3schools.com/html/html5_new_elements.asp).
- Някои стари елементи са премахнати (deprecated) или променени (`<center>`, `<big>`, `<i>` и т.н.)
- Дава възможност за валидация на уеб форми. Нови инпут типове
- audio, video
- canvas, svg и т.н.

<http://www.w3.org/TR/html5-diff>

Какво ново в CSS3 ?

- **media queries**
- box-sizing
- display: flex
- ::before , ::after
- filters
- анимации
- transformations
- transitions

Кои HTML елементи нямат затварящ таг?

- input
- img
- link
- meta
- source
- track
- br

HTML & CSS

- Каква е разликата между id, class и name атрибутите?
- Как да предовратим валидация на форма в html5
- Как да направим страницата responsive за мобилни устройства?
- Какво бяха HTML data атрибути?
- Какво е DOM ?
- Какво е BOM ?
- Какво е bootstrap ?
- Как разбираме дали дадено css пропърти или html атрибут се поддържат от всички широко използвани браузъри?

HTML & CSS

- Какво е viewport?
- Каква е разликата между width и device-width?
- Какво е мета съдържание? Дайте примери.
- Какво е отложено зареждане? Кога и как се ползва?
- Колко начина за селектиране на елементи знаете?
- Приоритетна верига - кога някой стил ще презапише друг?
- Как се позиционират елементи чрез CSS?
- px vs em vs rem
- псевдоелементи и псевдокласове!

JAVASCRIPT

- Какво е OOP?
 - ✓ *Обектно ориентирано програмиране. Представяне на нещата като обекти, които имат характеристики и поведение.*
 - ✓ *Използва 4 основни принципа: Наследяване, Полиморфизъм, Енкапсулация и Абстракция*
- Обекти в Javascript?
 - ✓ *сложна структура от данни, изградена от ключ-стойност двойки*
 - ✓ *всичко, което не е примитив (прост тип данни) е обект*
 - ✓ *имат характеристики и поведение (полета и методи)*
- Какво е JSON
 - ✓ *нотация за представяне на обектите в javascript. Ползва се най-вече в комуникационни канали като ajax, тъй като обектите могат да се представят като текст (JSON.stringify()), както и да се декодират след това отново в обекти (JSON.parse())*

JAVASCRIPT

- Какво е прототип?
 - ✓ всеки клас в JS има прототип
 - ✓ в прототипа съхраняваме всички споделени методи за инстанциите на неговия клас
 - ✓ така избягваме копирането на методите във всеки обект, но пък губим енкапсулация
- Какво е контекст? Какво е област на видимост? Какво е неймспейс?
- Как се селектират DOM елементи?
- Създаване и изтриване на елементи?
- Какво е JQuery?
 - ✓ javascript framework, js библиотека

JAVASCRIPT: събития

- Определение
- Кога възникват
- Прихващане
- Обработка
- Прекратяване
- Видове събития

JAVASCRIPT: събития

- ✓ *възникват, когато нещо се случи - например когато страницата се зареди или юзер-а кликне някъде по страницата*
- ✓ *събитието предизвиква "реакция" - такава която е по подразбиране или такава, която ние сме дефинирали*
- ✓ *тази реакция се нарича обработваща функция (handler) и представлява набор от действия, които се изпълняват след като събитието е възникнало*
- ✓ *за да зададем обработваща функция за дадено събитие, използваме DOM метода `addEventListener()`, който представлява механизъм за прихващане на събития*

JAVASCRIPT: грешки

- Кога възникват, как се обработват, за какво се използват?
 - ✓ има грешки породени от недоглеждане (неочаквани) и такива, които възникват "налично" и се използват за *flow control*
 - ✓ за втория вид ("наличните" грешки) използваме ключовата дума *throw*
 - ✓ грешките могат да се обработват (желателно е това да са само "наличните" грешки), за целта се използва *try-catch* механизма
 - ✓ Грешката предизвиква аварийен изход от текущия контекст, както и от всички съдържащи го, до първият съдържащ грешката *try-catch* блок. Ако няма такъв - грешката е необработена

AJAX, COOKIES

- Какво е AJAX
 - ✓ асинхронни заявки към сървъра
 - ✓ ползваме го, за да изпращаме или получаваме данни от сървъра, без да презареждаме страницата
 - ✓ като резултат - user experience-а се повишава многократно
- Какво е cookie
 - ✓ това е файл на компютъра на клиента, в който браузъра записва данни под формата: ключ=стойност
 - ✓ ползваме бисквитката за да пазим сесия и настройки, които да се помнят дори и след затварянето на страницата
 - ✓ бисквитките имат срок на годност

Общи

- Какво е уеб клиент?
- Какво е Git?
- Разликата между Git и Github?
- Какво е IDE?
- Какво е API?
- Разликата между front-end и back-end?
- Какво е HTTP? А HTTPS?
- Какво е SSL?



Въпроси?