



НИ Е ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Преговор (подготовка за 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 character encoding
- има нови елементи, като section, article, aside, audio, video, canvas, svg и т.н. Някои стари елементи са deprecated или променени (`<center>`, `<big>`, `<i>` и т.н.)
- много променени и нови атрибути (lang, charset, novalidate, и т.н.).
- дава възможност за валидация на уеб форми.
- нови APIs - storage, geolocation
- подобрен DOM (унифициране на event-ите, много нови методи и функционалност)

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

Какво ново в CSS3 ?

- **media queries**
- box-sizing
- display: flex
- 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?
- ВАЖНО! Как се позиционират елементи чрез CSS?
- px vs em vs rem

JAVASCRIPT

- Какво е OOP?
 - Обектно ориентирано програмиране. Представяне на нещата като обекти, които имат характеристики и поведение.
- Обекти в Javascript?
 - Всички инстанции и наследници на класа Object. Хеш таблица от характеристики и методи. Всичко което не е примитив е обект.
- 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()` / , който да се предаде към сървъра, както и да се декодира след това отново в обект / `JSON.parse()` /

JAVASCRIPT

- Какво е прототип?
 - това е образаца, същността на даден клас.
 - използва се за задаване на функционалност на обектите след дефиницията на самият клас
- Какво е контекст?
- Как се селектират елементи?
- Създаване и изтриване на елементи?
- Какво е JQuery?
 - javascript framework, js библиотека

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, който 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

- Грешки - кога възникват, как се обработват, за какво се използват?
 - има грешки породени от недоглеждане (неочаквани) и такива, които възникват "нарочно" и се използват за flow control
 - за втория вид ("нарочните" грешки) използваме ключовата дума **throw**
 - грешките могат да се обработват (желателно е това да са само "нарочните" 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, LESS

- Какво е AJAX
 - асинхронни заявки към сървъра
 - ползваме го за да презареждаме само парчета от страницата, а не цялата
 - като резултат - user experience-а се повишава многократно
- Какво е cookie
 - това е файл на машината на клиента, в който браузъра записва данни под формата: ключ=стойност
 - ползваме я за да пазим сесия и настройки, които да се помнят дори и след затварянето на страницата
 - бисквитките имат срок на годност!
- Какво е прекомпилиране на css, защо и как (Less и Sass)

Общи

- Какво е MVC?
- Какво е Git?
- Разликата между Git и Github?
- Какво е IDE?
- Какво е API?
- Разликата между front-end и back-end?
- Какво е HTTP?
- Какво е SSL?

HTML animation

CSS transforms

- 2d трансформации
 - `translate(x,y)` - преместваме по някоя от координатите
 - `scale(x,y)` - промяна на размера на някое от измеренията (височина, ширина, дълбочина)
 - `rotate(angle)` - завъртане
 - `skew(x-angle,y-angle)` - накланяне
- 3d трансформации: `translate3d`, `rotate3d`, `perspective`

CSS transitions

- Пример: `p { transition: all 2s; }`
- Правят "мек" преход от едно състояние (т.е. визия) на елемента в друго
- Има няколко варианта на преход: `ease`, `ease-in-out` и т.н.
- Задаваме време за което трае прехода в секунди
 - например ако нещо трябва само да се уголеми и заложим време 0.1 секунда - то само ще премигне. Ако заложим 2секунди - ще се увеличава постепенно в рамките на 2сек

- Можем да зададем transition само за определени properties, или за всички наведнъж
- Въпросните properties трябва да имат различни стойности в двете състояния
- Състоянията се превключват при смяна на клас или събития от тип hover, focus и т.н.
- Важно! При различните браузъри има вариации в името на transition пропъртито!!
- Ако комбинираме transitions и transforms, можем да направим интересни ефекти наподобяващи анимация

CSS animation

- Това е надграждане над transitions и transforms
- Добавя нещо много съществено за анимацията: keyframes
- Това означава, че можем да зададем повече от едно преходни състояния на елемента (нещо като кадри)
- Можем да пилложим повече от една анимации едновременно върху един елемент
- Важно: Има различен синтаксис в различните браузъри!!



Въпроси?

Resources

- <http://learn.shayhowe.com/advanced-html-css/css-transforms/>
- <https://css-tricks.com/almanac/properties/a/animation/>
- <https://developer.mozilla.org/en/docs/Web/CSS/animation>

Примери

<http://zenifytheweb.com/courses/lessons/lesson22/example/index.html>