|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | doctype |  |  |  |  |  |  |  |  |
|  | Html5, семантическая верстка - HTML-разметка и названия CSS-классов должны отражать не оформление, а смысл. Red vs warning |  |  |  |  |  |  |  |  |
|  | Респонсив верста, meta for mobiles  <meta name="viewport" content="width=device-width, initial-scale=1"> |  |  |  |  |  |  |  |  |
|  | Display none vs visibility hidden |  |  |  |  |  |  |  |  |
|  | Form, Input-label, required, disabled,Типы инпутов |  |  |  |  |  |  |  |  |
|  | Text overflow ellipsis, white-space no-rap |  |  |  |  |  |  |  |  |
|  | Схлопывание, box-model, border-radius, флоаты |  |  |  |  |  |  |  |  |
|  | Центрирование блока  flex-direction column;  justify-content: center  align-items: center  display: flex;  margin: auto;  top: 50%  transform: translate(-50%) |  |  |  |  |  |  |  |  |
|  | 2 колоночный макет |  |  |  |  |  |  |  |  |
|  | 3 колонки одинаковой высоты align-items stretch |  |  |  |  |  |  |  |  |
|  | Table-cell внутри большой img |  |  |  |  |  |  |  |  |
|  | Сверстать меню, позиционирование, Li:hover |  |  |  |  |  |  |  |  |
|  | <canvas> позволяет отображать графические изображения с помощью JavaScript. Также может быть использован для создания различного рода анимации  var canvas = document.getElementById( "c" );  var drawing\_context = canvas.getContext( "2d" );  drawing\_context.fillStyle = "blue";  drawing\_context.fillRect( 50, 50, 100, 100 ); |  |  |  |  |  |  |  |  |
|  | Что такое local storage in HTML5?  это способ для веб-страниц хранить пары ключ/значение локально с помощью браузера. Как и кукисы, эти данные сохраняется даже после ухода с сайта, закрытия вкладки браузера, с выходом из браузера или чего-нибудь еще. В отличие от кукисов эти данные никогда не передаются на удаленный веб-сервер (если только вы не пожелаете отправлять их самостоятельно).  HTML5-хранилище базируется на именах пар ключ/значение. Вы сохраняете информацию, основываясь на имени ключа, а затем можете получить эти данные с тем же ключом. Имя ключа это строка. Данные могут быть любого типа, который поддерживает JavaScript, включая строки, логические, целые числа или числа с плавающей запятой. Однако в действительности данные хранятся в виде строки.  5 мегабайт |  |  |  |  |  |  |  |  |
|  | Что такое session storage in HTML5?  HTML5 introduces the sessionStorage attribute which would be used by the sites to add data to the session storage, and it will be accessible to any page from the same site opened in that window i.e. session and as soon as you close the window, session would be lost.purpose to remember all the data in your session, and forget it as soon you close the tab (or window) you are working in. |  |  |  |  |  |  |  |  |
|  | В чем разница между cookies, sessionStorage, and localStorage.  Cookies are small text files that websites place in a browser for tracking or login purposes. Meanwhile, [localStorage and sessionStorage](https://www.inkling.com/read/javascript-definitive-guide-david-flanagan-6th/chapter-20/localstorage-and-sessionstorage) are new objects, both of which are storage specifications but vary in scope and duration. Of the two, localStorage is permanent and website-specific whereas sessionStorage only lasts as long as the duration of the longest open tab.  кукисы включаются в каждый HTTP-запрос, замедляя тем самым ваше веб-приложение на напрасную передачу одних и тех же данных снова и снова;  кукисы включаются в каждый HTTP-запрос при передаче данных через Интернет в незашифрованном виде (даже если все веб-приложение передается через SSL);  кукисы ограничены объемом данных примерно 4 Кб — достаточно, чтобы замедлить ваше приложение (см. выше), но не достаточно, чтобы быть полезным.  sessionStorage у каждой вкладки свой  т.е. записать туда что-то в одной вкладке, а прочитать во второй не получится даже если в этих вкладках будет открыта одна и та же страница |  |  |  |  |  |  |  |  |
|  | сборка |  |  |  |  |  |  |  |  |
|  | Css — препроцессоры, написать |  |  |  |  |  |  |  |  |
|  | Системы контроля версий |  |  |  |  |  |  |  |  |
|  | Подключение js, css |  |  |  |  |  |  |  |  |
|  | Repaint reflow |  |  |  |  |  |  |  |  |
|  | Анимация @keyframes { from {} to {} } |  |  |  |  |  |  |  |  |
|  | типы данных  Есть 5 «примитивных» типов: number, string, boolean, null, undefined и 6-й тип – объекты object. |  |  |  |  |  |  |  |  |
|  | 'use strict'; |  |  |  |  |  |  |  |  |
|  | == vs === |  |  |  |  |  |  |  |  |
|  | Принципы ООП |  |  |  |  |  |  |  |  |
|  | function doNothing() { /\* пусто \*/ }  alert( doNothing() ); // undefined |  |  |  |  |  |  |  |  |
|  | Type of, instance of /typeof null;  typeof undefined // "undefined"  typeof 0 // "number"  typeof true // "boolean"  typeof "foo" // "string"  typeof {} // "object"  typeof null // "object" (1)  typeof function(){} // "function" (2) |  |  |  |  |  |  |  |  |
|  | element.addEventListener(event, handler[, phase]); добавление на динамически добавленные ‘элементы  Детали произошедшего браузер записывает в «объект события», который передаётся первым аргументом в обработчик. |  |  |  |  |  |  |  |  |
|  | Прекращение всплытия  почему срабатывает на верхнем элементе  Многие события автоматически влекут за собой действие браузера.  Делегированеи |  |  |  |  |  |  |  |  |
|  | var lastItem = goods[goods.length - 1]; // получить последний элемент  Используем length для укорачивания массива |  |  |  |  |  |  |  |  |
|  | Если вообще не указать аргументов – скопируется весь массив:  var fullCopy = arr.slice(); |  |  |  |  |  |  |  |  |
|  | Склеить два concat |  |  |  |  |  |  |  |  |
|  | Splice в любое место  С какими методами добавления элементов в массив вы знакомы  Push - Добавляет элемент в конец массива (быстрее)  Unshift - Добавляет элемент в начало массива:  Метод «arr.filter(callback[, thisArg])» используется для фильтрации массива через функцию.  Он создаёт новый массив, в который войдут только те элементы arr, для которых вызов callback(item, i, arr) возвратит true.  Метод «arr.map(callback[, thisArg])» используется для трансформации массива.  Он создаёт новый массив, который будет состоять из результатов вызова callback(item, i, arr) для каждого элемента arr.  Эти методы используются для проверки массива.  Метод «arr.every(callback[, thisArg])» возвращает true, если вызов callback вернёт true для каждого элемента arr.  Метод «arr.some(callback[, thisArg])» возвращает true, если вызов callback вернёт true для какого-нибудь элемента arr. |  |  |  |  |  |  |  |  |
|  | Функция которая делаетпервую букву заглавной  function ucFirst(str) {  // только пустая строка в логическом контексте даст false  if (!str) return str;  return str[0].toUpperCase() + str.slice(1);  }  alert( ucFirst("вася") ); |  |  |  |  |  |  |  |  |
|  | [ {index: ‘26px’} , {index: ‘35px’} ] |  |  |  |  |  |  |  |  |
|  | Call apply |  |  |  |  |  |  |  |  |
|  | Что такое замыкание  sum(5)(-1) = 4 |  |  |  |  |  |  |  |  |
|  | Установить таймер/удалить таймер  var timerId = setTimeout(function() { alert(1) }, 1000); |  |  |  |  |  |  |  |  |
|  | unction isPalindrome (str) {  if (typeof str === 'string') {  var original = str.toLowerCase().replace(/ /g, ''),  reversed = original.split('').reverse().join('');  return original === reversed;  }  return false;  }  function isPalindrome2 (str) {  if (typeof str === 'string') {  var original = str.toLowerCase(),  length = original.length,  endIndex = length - 1,  result = true;  for (var i = 0; i < length; i++) {  var symbol = original.charAt(i);  if (symbol !== ' ') {  var endSymbol = original.charAt(endIndex);  while (endSymbol === ' ' && endIndex >= 0) {  endIndex--;  endSymbol = original.charAt(endIndex);  }  result = result && symbol === endSymbol;  endIndex--;  if (!result) {  return false;  }  }  }  return result;  }  return false;  } |  |  |  |  |  |  |  |  |
|  | // в каком порядке числа 1, 2 и 3 появятся в консоли  console.log(1);  setTimeout(function () {  console.log(2);  }, 200);  foo(); // выполняется 5 секунд  console.log(3);  Ответ: 1 3 2 |  |  |  |  |  |  |  |  |
|  | // Как найти максимальное/минимальное число в массиве, не используя циклы  Math.max.apply(null, []);  Math.min.apply(null, []); |  |  |  |  |  |  |  |  |
|  | // Записать пример эмуляции класса User с одним свойством и одним методом  // Создать класс-наследник Admin, родителем которого будет User. У админа есть своё новое свойство и свой метод.  // --------- Класс-Родитель ------------  // Конструктор родителя пишет свойства конкретного объекта  function Animal(name) {  this.name = name;  this.speed = 0;  }  // Методы хранятся в прототипе  Animal.prototype.run = function() {  alert(this.name + " бежит!")  }  // --------- Класс-потомок -----------  // Конструктор потомка  function Rabbit(name) {  Animal.apply(this, arguments);  }  // Унаследовать  Rabbit.prototype = Object.create(Animal.prototype);  // Желательно и constructor сохранить  Rabbit.prototype.constructor = Rabbit;  // Методы потомка  Rabbit.prototype.run = function() {  // Вызов метода родителя внутри своего  Animal.prototype.run.apply(this);  alert( this.name + " подпрыгивает!" );  };  // Готово, можно создавать объекты  var rabbit = new Rabbit('Кроль');  rabbit.run(); |  |  |  |  |  |  |  |  |
|  | Es 2015 |  |  |  |  |  |  |  |  |
|  | ajax |  |  |  |  |  |  |  |  |
|  | Frame works |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Где хранятся встроенные методы массивов array.prototype |  |  |  |  |  |  |  |  |
|  | Что такое сериализация  Процесс перевода структуры данных в последовательность битов/восстановление начального состояния структуры |  |  |  |  |  |  |  |  |
|  | В чем заключается разница между get post (get url , post тело http)  get предает данные через url (ограничение 1024 символа, виден всегда)  post через тело http, большие объемы данных |  |  |  |  |  |  |  |  |
|  | Что такое json  Текстовый формат обмена данными основанный на js |  |  |  |  |  |  |  |  |
|  | Что такое XMLhttpRequest  Объект XMLhttpRequest дает воЗможность браузеру делать http запросы к серверу без перезагрузки страницы |  |  |  |  |  |  |  |  |
|  | По какому протоколу осуществяется передача данных в сети интернет |  |  |  |  |  |  |  |  |
|  | В каком порядке будут располагаться элементы массива после сортировки:  [5, 6, 55, 1, 39, 45, 10].sort();  Это произошло потому, что по умолчанию sort сортирует, преобразуя элементы к строке. |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Test

Test