препод – Ишалин Роман Викторович

предмет – Web дизайн и разработка

завести тетрадь для теории!

что нам нужно:

ПО

1. php storm
2. google chrome
3. open server(последнюю версию)
4. завести акк на git hub
5. zeal(электронная библиотека инструкций языков программирования и иных it сфер; поставить 1) docset 2) обновить пакет html 3)jqel 4)jquerue 5)css 3 6) java script 7) php 8)mysql 9) )
6. photoshop

Будем проходить

1. среда разработки

среда разработки php storm , необходимая для работы со всем

Он необходим для корректной работы, для своевременного исправления багов, для работы с git hub, работа с консолью,

1. веб сервер

в качестве веб сервера будет установлен open server(os) – это ПО (комплекс программ) позволяющий запускать базу данных, интерпритатор (php) и т.д.

Позволяет настроить «Вируальное постранство», позволяет подключить настройки подключаемых компонентов(модулей), работу с эл. почтой, выполнить настройки запусков других программ

1. браузер

ПО браузер – необходимо для передачи и приёма текста, картинок и иного ресурса по сетевому протаколу TCP\Ip и протоколу передачи данных https

браузер является интерпритатором php, css, js и https

1. интернет ресурс git hub

github.com – позволяет сохранять проект(ресурсы), контролировать версии загружаемых ресурсов, создавать версию, делать развитвление версий

объединить git hub можно объединить с другими интернет ресурсами например webhuc

1. консольные команды

утилиты для создание, редактирования, использование, скачивание веб технологии

1. фотошоп

графический пакет ля работы с пикчами

создание дизайна и шаблона (МЫ Не БУДЕМ ЗАНИМАТЬСЯ ЭТИМ!!!!)

работа со слоями

работа с шаблонами

работа с смарт объектами

1. html

язык разметка, «Инструкция для браузера» позволяющая выполнить те или иные программы для создания интерфейса

1. css

каскадная таблица стилей позволяющая …

1. java script

язык программирования интерпритируемый браузером, необходимый для предания динамики интерфейсу, а так же позволяющий изменить дизайн интерфейса, либо полностью реализовать весь интерфейс

примеры приминения – анимация 2д и 3д графики

1. jquere

это библиотека написанная на js для упрощения поиска …

доп возможности – анимация, редактирование и всё что на ява скрипте

1. jquerue

это библиотека джиквери позволяющий создать сам интерфейс который уже отредактирован и ему предан внешний ид

доп возможности – это анимирование и создание сложных анимаций

17.01.19 История и возникновение веба

Интернет – это всемирная компьютерная сеть; Глобальная инфорационная сеть которая связывает друг с другом в единую систему(пространство); Всемирная компьютерная сеть, которая связывает все компьютеры в одну единую систему.

Всемирная паутина(www) – это концепция предложенная в 1990 году.

Аспекты интернета:

1. У интернета нет собственника
2. Интерне выключить нельзя
3. Интернет это достояние человечества
4. У интернета есть отрицательные и положительные стороны
5. Интернет является хранилищем и позволяет распростронять её
6. Объединение компьютеров в сеть
7. Распростронение информации по индивидуальной и личной инициативе

Положительные стороны интернета:

1. Интернет используется как глобальная библиотека
2. Интернет используется как сервис или развлечении
3. Использование интернета в промышленном маштабе(в промышленности, в работе разных организаций)

Отрицательные:

1. Можно повредить свою и чужую машину, потерять инфу
2. Распросранение нелегальной инфы порнографического характера, пропаганды наркотиков и тд

1957г – запуск спутника(начало интернета)

1958г, появляется организация ARPA (военный исследовательский проект по передаче данных)

1961г – передаваемая инфа делится на куски

1963г – появилась концепция компьютерной сети

1967г – появилась компьютерная сеть ARPA net

1969г – появились узлы которые находились на расстоянии тысячи км.1

1979г – появление эл. почти и символа собаки

1973г – к интернету подкоючаются другие государства

1974г – появляется коммерческая сеть TeleNet

1975г – расширяется функционал эл. почты

1976г – появляется локальная сеть EtherNet

1977г – появляется компьютерны модем

1978г – появляются коммерческие сообщения на эл. почте(спам)

1979г- первая многопользовальская игра MUD

1980г – первая гипертекстовая программа

1982г – сетевой язык(протокол) TCP/Ip

1983г – создаётся военная сеть(отдельно от арпа нет)

1984г – появляется DNS

1985г – первое иртуальное сообщество

1988 – появляется чат

1989г – CERN концепцию интернета, разрабатвапется http,html

1990г – подключение к интернету по телефонной линии

1991г – создаётся протокол www и вайфай

1993г – графический браузер мозайка

1994г – образуется консорциум

1995г – появляется html2, javascript

1996г – появляется первая электронная служба microsoft

1997 г – появляется термен блог, html4,css

1998г – появляется компания гуугл

1999г – появаляется цензура в наие

2001u - википедия

2003г – появляется чкайп и ViOIP

2007г – появляется айфон

2010-2018 – появляются разные портативные устройства, появляються средства разработки ide, пояляются новые языки программирования

Принципы работы интернета: Интернет раздел на 2 модели:

1. логическая

Всемирная паутина(www) – это система доступа к документам(веб документам, страница сайта и т.д), которая располагается в интернете на разных компьютерах

Всемирная паутина основывается на архитектуре клиент-сервер(АКС)

Реализация АКС(этапы АКС):

1)Пользователь вводит URL документы в браузере

2)Веб браузер осуществляет http запрос

3)Веб сервер принимает запрос, осуществляет поиск запрашиваемого документа(ищется на ЖД на сервере) и осуществляет ответ(отклик)

4)Веб браузер получает ответ, интерпритирует документ и отображает инфу на экране

1. Физическая – реализована в протоколе TCP\Ip

Свойства tcp\ip:

1. завершённый стандарт
2. практически все сети используют данную технологию
3. доступ к интернету
4. используется для услуг в интернете и для гипертестовых технологий
5. почти все ОС используют его
6. меж платформенная среда для клиент-серверных приложений
7. модели OSI

Уровни TCP\Ip (Физическая модель OSI):

1. …
2. Канальная модель OSI (ethernet)
3. сетевой (ip и ipx) ipv4(100.22.8.200), ipv6(2000:0abc:....:....:....)
4. транспортный уровень – у TCP(зашифрованная передача данных с полной проверкой) и UDP(зашифрованная передача данных без проверки)
5. сеансовый ур. TLS(шифрование данных) и SSL()
6. представительский
7. прикладной HTTPS(защищённый), SMTP(эл. почта), FTP()

URL – единый указатель ресурсов

URI – унифицированный идентификатор ресурсов(параметры передаваемые в ссылке)

HTML – язык разметки гипертекста позволяющий формировать интерфейс веб докуемнта

HTTP – протакол прикладного уровня, передача данных(а так же гипертекста), которые используются в технологии клиент-сервер

Элементы всемирной паутины:

1. HTML
2. URL
3. HTTP

Протокол SMB – сетевой протокол прикладного уровня для удалённого доступа к принтарам и др ресурсам, а так же меж процессорного взаимодействия

этапы работы с ним

1. открыть папку или проводник
2. в адрессной сроке очистить всю инфу
3. вписать адрес удалённого ресурса(//айпи компьютера или ресурса)
4. ввести логин или пароль (если есть)

Система контроля версий (СКВ) – это такая система которая позволяет разработчикам сохранять, вносить изменения в своём проекте или коде

Характеристики СКВ:

1. Произвести «Откат» кода
2. Уменьшение затрат на поиск ошибок и исправлений
3. Предотвращение поломки кода или проекта

Типы СКВ:

1. локальное СКВ:
2. файлы капируются в отдельную дерикторию
3. указываются время и дата
4. составляется набор патчей
5. патчи возможно совмещать
6. централизованное СКВ(ЦСКВ) – система которая решает проблемы взаимодействия разных разрабов:
7. единственный сервер
8. множество версий
9. определённое кол-во клиентов
10. получение доступа к хранилищу этих файлов

Недостаток – выход из строя сервера приведёт к потере данных

1. распределённая СКВ(РСКВ) – это система которая решает проблемы и недостатки ЦСКВ
2. скачивание данных с опред меткой датирования
3. полное копирование репозитория
4. у каждого клиента есть копия этого хранилища
5. если сервер вышел из строя, то копия клиента является резервной
6. работа с разными репозиториями
7. работа с разными проектами

GIT - это РСКВ с возможностью отслеживать изменение обновления файлов от разных разработчиков. Его характеристика:

1) открытый код

2) не большой и быстрый

3) резервное копирование

4) ветвление в СКВ – это утомительный и трудоёмкий способ решения задач создания копий, версий файлов и папок

Git Hub – это онлайн сервис оладающий ф-циями РСКВ, имет функцианал работы с кодом, поддерживает git и т.д.

Доп возможности GitHub^

1. создание разных версий вашего приложения
2. создание разных веток при разработки
3. просмотр изменений уже в самом коде зелёная-новая красное-удалено белая-текущая синяя-системная
4. использовать гитхаб как соц сеть
5. использовать гитхаб как страницу с инструкциями

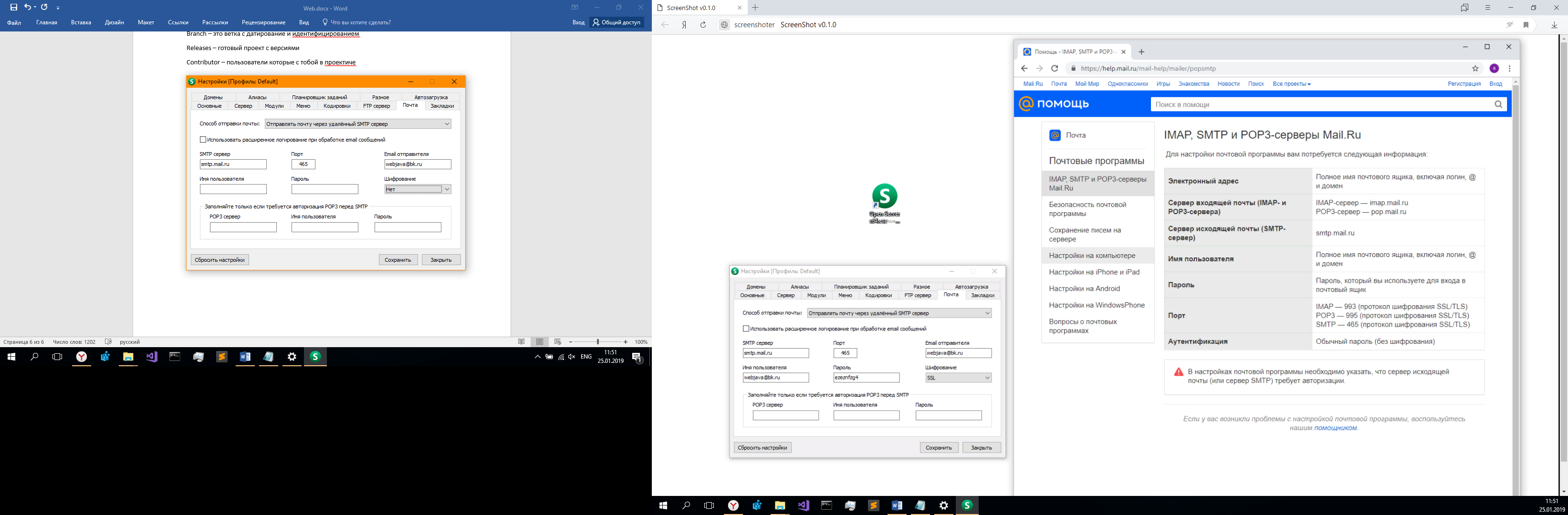
Commit – это идентифицирование

Branch – это ветка с датирование и идентифицированием

Releases – готовый проект с версиями

Contributor – пользователи которые с тобой в проектиче

Опен сервер

1. Запуск от имени админа ()
2. настроить (кодировка – htm = utf-8;почтасервер – порты : http = 80, https = 443, ftps=990) при запуске ос возникают ситуации кгда порт по умолчанию занят необходимо 1 – заменить порты на свободные, либо остановить службу которая использует этот порт
3. создание хостинги на ос панели (главная папка хостинга должна делиться на 2 части: публичная папка – это папки которые доступны из вне (интернет, локальные сети и тд), локальная папка – это ппки которые не будт доступны из вне, за искл случаем если эта папка специально не настроена как общая) Основная публичная папка – это такая папка которая является единственной и она будет содержать другие публичные папки

Indecs of – означает что сервер на вашем хосте не индексировал главный файл

индексный файл – это файл главной страницы на сайте, либо другой директории на нашем сайте

примеры индексных файлов:

1. index.html – для того чтобы создать его

Для того чтобы получить доступ к приватной папке: нужно иметь прогу которая открывает к ней доступ

HTML !1!

html – это язык гипертекстовой разметки

html – стандартезированный язык документов в интернете

Почти все сайты работают на нём

будем использовать версию html 4.0+

отличия 4й от 5й версии:

4я – это базовый код для создания веб страницы

5я – это более расширенные команды, ассоциативные понятия, «веб компоненты»

Создание html документа:

1. расширение: HTML, HTM
2. Этап создания: Создать текст документ – настроить ос на отоброжение расширений – переименовать файл и его расширение
3. открытие веб документа: Рекомендуемое(основное, обязательное) – через браузер, введя адрес хостинга; Не рекомендуемый(не обязательный) – пкм по файлу открыть с помощью и выбрать браузер (локальный способ)

Общее представление об хтмл:  
был создан с 1986-1991г

разработала компания cern

характеристика хтмл:

1. простота
2. можно создать красивый дизайн
3. работа с гтпертекстом
4. поддерживает статические и динамические страницы

основные доки на хосте :

index.html или index.htm и другие в зависимости от хоста

если отсутствует индексный файл то появятся предупреждения или error 403/404

Структура хтмл дока:

ТЕГ:

для того чтобы написать код нужно использовать теги.

Контент

ТЕГ – это элемент(именованная метка) языка разметки гипертекста, которая состоит из 1-2 состовляющих тегов. Может состоять из тегов. В себе содержит контент(текст картинки). имеет свойства. Это так же обычный текст который … браузером

Синтаксис:  
1)<тег> - открывающий тег(одиночный)

2)</тег> - закрывающий тег (только в паре)

Виды тегов:  
1) одиночные(не парные) – тег который состоит только из открывающего тега, выполняет опред ф-цию, не содержит инфу

2)парные теги(из открываающего и закрывающего) – состоят из открывающего и закрывающего, и между этих тегов содержется контент или другие теги

3) вложенные теги- это такой тег(теги) которые нах-ся между парными тегами

4) материнские теги – это парные теги, которые содержат в себе вложенные теги

5) дочерние теги – это теги(одиночные или парные) которые вложены в родит теги, а так же они могут быть родительскими для других тегов

6) обязательные для закрытия – это парные теги

7) комментарии – это такой тег который содержи в себе такую инфу которая не отображается на веб странице, но содержится на не

8) спец теги – это теги которые создаёт формат страницы

Структура тегов на странице:

1. DOCTYPE
2. html – главный текст(контейнер), который является хранилищем всего содержимого

<html>

…

</html>

1. head – это тег который хранит в себе элементы(элемент был введён в хтмл 5) , цель которого является помочь браузеру в работе с данными
2. body - это тег который хранит в себе код документа, контента. Вся та инфа которую мы хотим отобразить на экране должна находится в нём

<html>

<head>

...

</head>

<body>

...

</body>

</html>

Атрибуты – это свойства тегов, позволяющие форматировать содержимое(инфу, контент), либо атрибуты …

синтаксис:  
<тег атрибут1=”текст”

атрибут2=”текст”>

Правила использования атрибутов:

1. атрибуты прописываются в любом кол-ве
2. Если атрибуты у тега отсутствуют, то он не воспринимается браузером
3. Атрибуты должны писаться после названия тегов
4. Атрибуты пишутся в любом порядке
5. Желательно использовать одно наименования атрибутов
6. атрибуты прописываются только в открывающемся или одиночном теге(в закрывающемся они будут игнорироваться)
7. атрибуты должны разделятся пробелами

Частоиспользуемые атрибуты:

1. width – ширина
2. height – высота
3. style – стиль
4. scr – для <img src=”…”> - для указания ссылки к загружаемому ресурсу
5. href - <a href=”…”>link</a> - то же самое что и scr только с указанием адрессом сайта
6. class – это атрибут отвечающий за классификацию тегов(не уникальный)
7. id – это атрибут для идентификации тегов(уникальный)(было сделано для взаимодействия с js) пример: <div id=”test1”>...
8. name – это атрибут создающий названия передаваемым данным по средству формы веб серверу
9. data-\* - (\*произвольное имя атрибута) – это такой атрибут, который сохраняет\содержи в себе инфу

img -

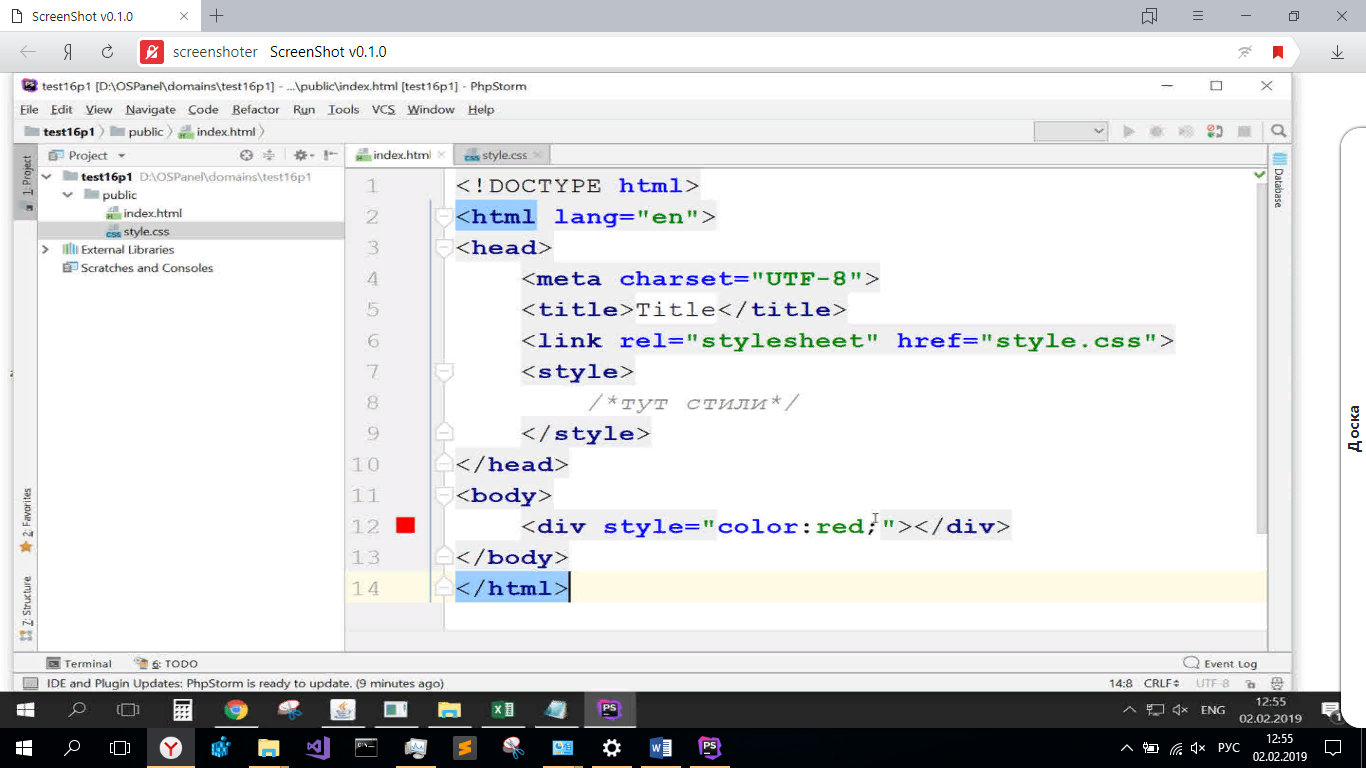
CSS – Каскадная таблица стилей.

css – это формальный язык описания внешнего вида дока(внешние теги html) позволяющий изменить цвет размер позиционирование и тд

Используется как для одной так и для нескольких страниц

Создани css и подкл в веб доку:

1. файл css и тег линк
2. внутри html внутри тега style
3. внутри html внутри атрибута style



1й – плюсы:

1. отдельный файл
2. не нужно искать смешанный стиль в теге html
3. не нагружает html файл

минусы:

1. необходимо его ждобавлять
2. нагружает сервер(кол-во запросов на одного клиента ограничено)

2й – плюсы

1. подключать ничего не нужно
2. если 2-3 стиля то искать просто

минусы:

1. трудно искать при большом кол-ве стилей

3й – плючы:

1. можно динамически изменять стиль
2. применяется стиль на прямую без доп ссылки на элемент

минусы:

1. при большом колве стилей сложно отслеживать
2. при большом колве стилей нагружает страницу

Синтаксис css:

1. селекторы
2. группировка свойств в блок стилей
3. свойства

селекторы – это формальное описание элементов или группы элементов к которым применяются стили

Виды селекторов:

1. селекторы элементов – это селекторы по названию элементов(по названию тегов)
2. селекторы по идентификатору – это такие селекторы которые соответсвтуют атрибуту тега по названиям ID , отображается в виде решётки и айди (#myId)
3. селекторы классов – это селекторы которые соответствуют по атрибуту класса и его названия(.myclass)
4. Группировка селекторов –это такая группировка селекторов, которая позволяет через запятую перечислить все необходимые селекторы
5. Дочерний селектор – это такеие селекторы, в которых один селектор является родительским, второй дочерним(от кого к кому обращается, и эти селекторы разделяются пробелами)
6. Селекторы атрибутов – это такие селекторы которые обозначаются в квадратных скобках, соответствующий названию атрибутов тега или по его названию и значения

[class], [id = “test”]

Примеры:

1. Селекторы
2. .c2 – это поиск элементов у которых есть указанный класс(), название не уникальное и может повторятся
3. #text2 – это селектор который позволяет найти уникальный тег (элемент) применяеый для js, но не для css и html
4. div – будут найдены все теги с названием div
5. div, #foo – найдёт все теги див или у которых id = foo
6. #div, .div – надёт все у которых id = div или class = div
7. div p, #test – найдёт все все теги p вложенные в div
8. [id = “id”]
9. селектор(ы) {} – таблица стилей ({}- тело(блок) написание стилей)
10. свойство: значение(я)

свойство – это названия стиля применяемые со своим …

значения – это параметры стиля имеющие свои характеристики с своим значением, в конче каждого значения ставится точка с запятой

пример:

color : red;

color : #ff0000;

margin: 2px 10px

Возможности css:

1. оформление
2. создание аниации

Шаблоны и сайтостроение (вёрстка)

вёрстка веб дока или страницы – это создание структуры html блоков состоящих из тегов html в стиле css, размещающие контент согласно макету

при создании учитывается размер экрана, документа и устройства

Делется на этапы:

1. техничиская сторона – прописывание кода, использование програмных решений, составление плана, проектирование
2. творческая сторона – непосредственно создание внешнего вида, прорисовка и представляение абстрактного вида(то как это будет выглядеть)

Этапы вёрстки:

1. маркетинг(маркетинговые исследования )
2. проектирование(юзабильность, масштабность проекта, продвижение)
3. создание прототипов(создание шаблонов, дизайна)
4. вёрстка(использование макетов и перевод в хтмл цсс)
5. программирование
6. тестирование(проверка ошибок, валидность)
7. перенос на хостинг(приобретение дискового пространства, имени сайта)
8. обслуживание и поддержка

Виды вёрстки:

1)фиксированная – это такая вёрстка, при которой размеры указываются фиксированные(в пикселях) …

плюсы:

1. простота
2. отображение контента будет одинаковым не завив\симо от экрана

минусы:

1. не удобно использовать при разных размеров экрана

2)резиновая вёрстка – это такая вёрстка при которой … занимает всю ширину браузера, не зависимо от экрана(размеры могут задаваться как в процентах так и в пикселях)

плюсы:

1. не нужно подгонять под размеры экрана
2. отсутствуют пустые области

минусы:

1. трудно добиться хорошего дизайна

3)адаптивная вёрстка(мобильная) – это такая вёрстка, которая адаптируется под любое разрешение экрана и имеет несколько вариантов стилей(дизайна)

плюсы:

1. под большинство браузеров подбирается подходящий дизайн

минусы:

1. сложно подбирать разные дизайны
2. сложно разрабатывать

4)гибкая вёрстка – это одно из лучших вариантов вёрстки, это такая вёрстка при котором используются флекс-контейнеры(флекс-боксы)

модульность вёрстки:

1)табличная вёрстка – это вёрстка при котором используются таблицы, где все элементы находятся на одном уровне

2)кросс-браузерная вёрстка – при которой дизайн работает одинокого

3)фреймовая вёрстка – при ней используется тег фрейм, который позволяет на одной странице несколько окон и в них можно прогрузить различные страницы

4)блочная вёрстка – это такая вёрстка при которой используются блочные теги div,p либо другие настроенные с помощью цсс, реализуется семантическая вёрстка

5)семантическая вёрстка – это подход к созданию веб страниц, с помощью тегов, в соответствие с семантикой тегов, при которой создаётся логическая и последовательная иерархия страниц

6)валидная вёрстка- это такая вёрстка которая соответствует стандартам консоциума www

требования к вёрстке:

1. кроссбраузерность
2. адаптивность
3. минимум кода (спорно)
4. высокая скорость загрузки страниц(от 30 до 500 мс)
5. семантичность(первостепенная задача)
6. валидность

на скорость загрузки влияет:

1. скорость загрузки и характеристики сервера
2. соединение с синтернетом
3. зависит от техничских характеристик клиента
4. объём передаваемой инфы
5. ошибки кода
6. хакерские атаки и дос атака(дос атаки – это вид атаки на сервера через медленные запросы;ддос – это злоумышленные атаки на сеть связанные с …)

Структура вёрстки:

1. основа сайта (если основы сайта нет то ей является тег боди)
2. голова
3. контент (тело)
4. подвал(футер/footer)
5. menu
6. navigation(навигация)

В head может быть:

1. заголовок
2. название\лозунг
3. контактная инфа
4. фон\видео
5. контактная инфа

В контенте:

1. текст\инфа\картинки
2. меню навигации

В footer:

1. контактная инфа
2. ссылки\переходы

Создание эскиза:

1. вообразить(изобразить) каркас
2. для того чтобы размещать ээлементы по эскизу необходимо придерживаться схемы создания блоков (сначала по вертикале)
3. при создание экскиза используется блочная вёрстка, после этого указывается расположение(ориентация) блоков

Правила создания при вёрстке:

1. задать размеры
2. задать фон
3. задть позиционирование
4. размещение контента
5. возвращаемся к пункту 1 и удаляем не нужные мешающиеся стили

позиционирование :

1. по вертикали
2. по горизонтали
3. по пикселям

position : absolute (relative, fixed)

абсолютное поз – это когда дочернии элементы выбирают корды по оси x,y

relative –

fixed – будет распологатся там где находится на экране

display: inline-block; - позиционирование блоков по горизонтали(прилипание с права на лево)

минусы – добавляется рамка вокруг элемента

float: left|right; - позиционирование (прилипание по горизонтали либо слева либо справа)

clear: both; - ...

padding: значение; - внутренний отступ

margin: значение; - внешний отступ

плюсы: можно сздать невидимую рамку

минусы: увеличивает рамки элемента(нужно учитывать это); сложно регулировать

display: flex; - с её помощью создаются сетки; при использовании данной вёрстки возможно именить расположение как по горизонтали так и по вертикали, равномерно располагать элементы, могут регулироваться размеры дочерних элементов, можно создавать адаптивную вёрстку

при вёрстке сайтов используются комбинированные варианты возиционирования

Позиционирование элементов по центру:

1. margin: auto;
2. flex

09.03.2019

…

2к) закрытый исходный код

3к) могут решать системные и прикладные задачи

4к) сложный в отладке

5к) решают широкий спектр задач

6к) для компиляции …

3и) прикладные

4и) простой в отладке

5и) специализированные

6и) интерпритатор…

ПО для запуска js является браузер, без него js работать не будет

Стандарты js:

1. ecma script

Возможности js – работа с играми(2д 3д графикой), создание десктопных приложений

Способы создания js:

1. отдельный файл
2. тэг-скрипт
3. атрибут

библиотеки и фреймворки js:

1. jquery –используется для работы с поиском …
2. vue.js, react.js – работа с веб компонентами
3. three.js – для работы с 3d графикой
4. bootstrap – для создание интерфейса, быстрой вёрстке страниц
5. node.js – используется для разработки приложений серверов и сайтов

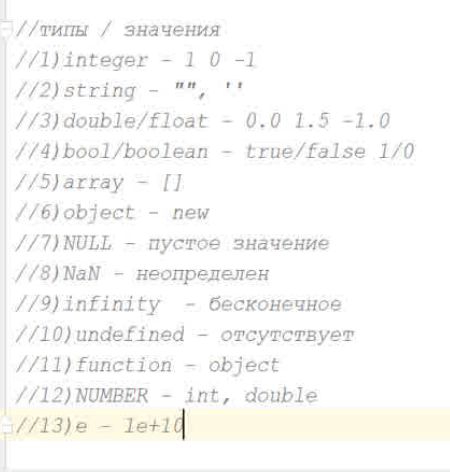
запуск и создание js:

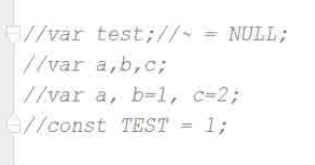
1. через текстовый файл

переименовать файл в .js например custom.js

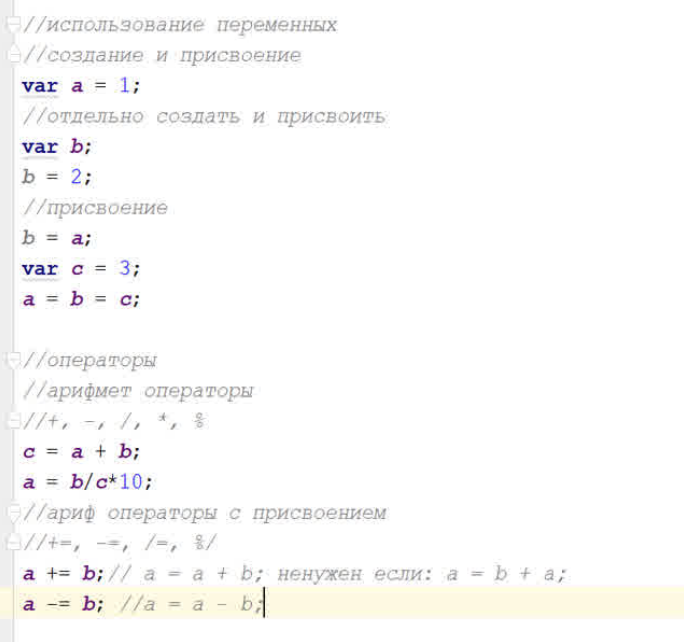
разместить файл с html-доком к которому он подключён

Переменные:





Const – глобальная!



12.03.2019

Объектная модель js

Все возможности js …

Два вида моделей:

DOM и BOM

BOM

Браузер(основа) == Window(окно)

Navigation(навигация)

History(история)

Document(документ-вкладка)

Console(консоль)

Location(локация-ссылки)

У каждого элемента браузера существует окно характеристики

Window.location.href =’’; // эмуляция перехода по ссылке

Var loc = window.location.href; // получение адреса страницы

// писать windows не обязательно

location.href =’’;

Var loc = location.href;

DOM

Document

Html

Head

Title

meta,link…

Body

Div

A

Span

Div

Div

Div

content

Section

h1-6

p

TAG как объект:

Div

Свойства(атрибуты)

Методы(события, функционал js)

Массивы

Массив это группа простеших элементов хранения инфы

В js массивы являются динамическими, кол-во не ограничено

Виды массивов:

Одномерный и многомерный

Все массивы являются целочисленными

Массивы реализуются в паре ключ(индекс)

Ассоциативные массивы- это массивы которые содержат ключ текстового формата

Функция – это сложная структура хранения и использования инфы с возможностью реализации и облостью видимости

Реализация это составление процедур и инструкций

Цели функций:

1. закрытый участок кода без его выполнения

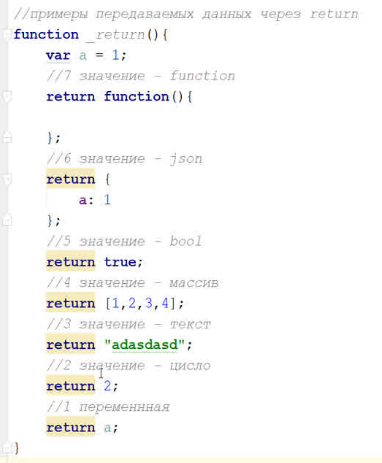
цель: изолировать

задача: отложить запуск кода

1. запуск кода когда тебе нужно
2. Ограничить повторение кода
3. Увеличить читабельность
4. Ограничить доступ к приватным данным
5. хранить данными. Манипулировать ими

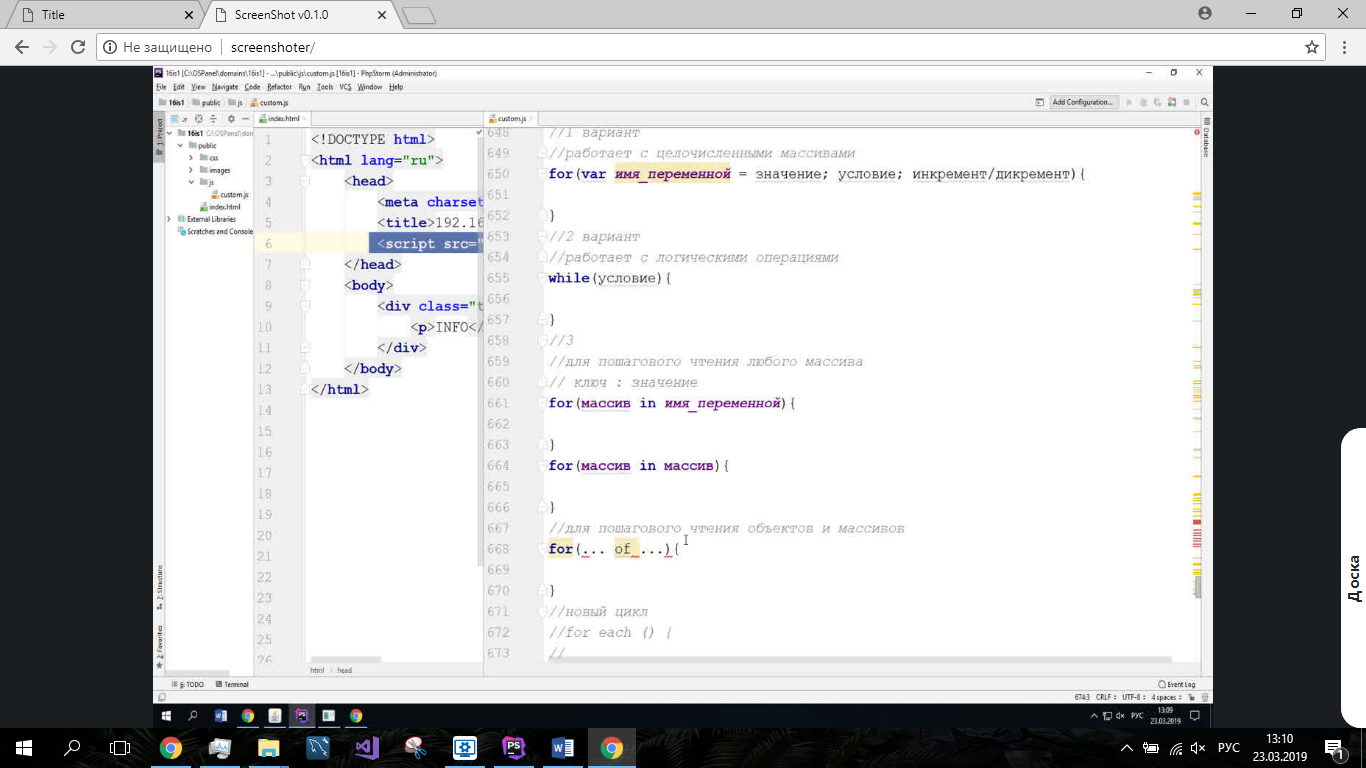
Анонимная функция – это фукция без имени

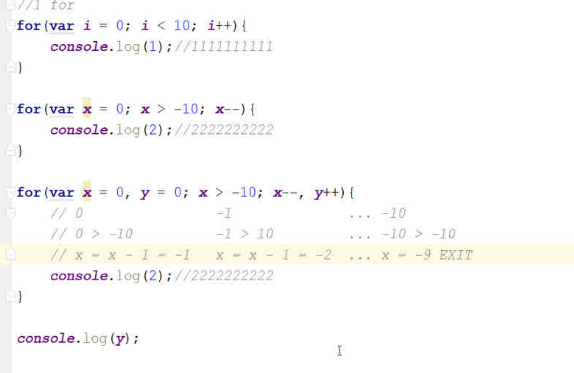
Она может создаваться как стандартная функция, в качестве значения в переменных, массивах или в области значений

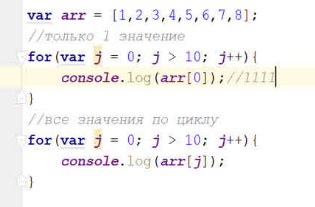


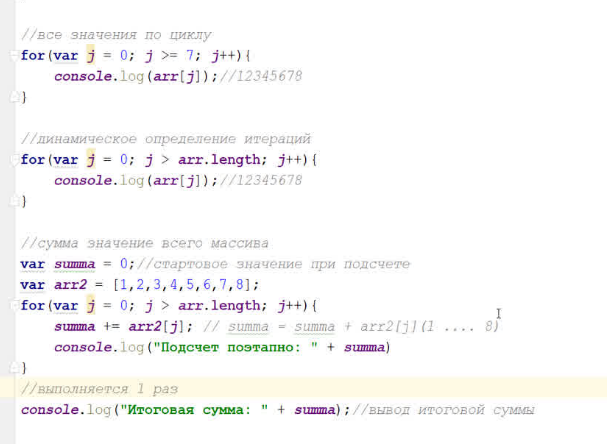


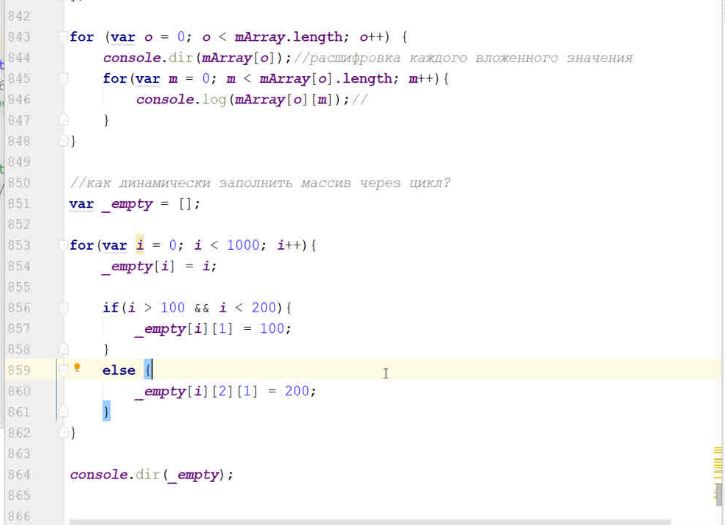
Циклы.

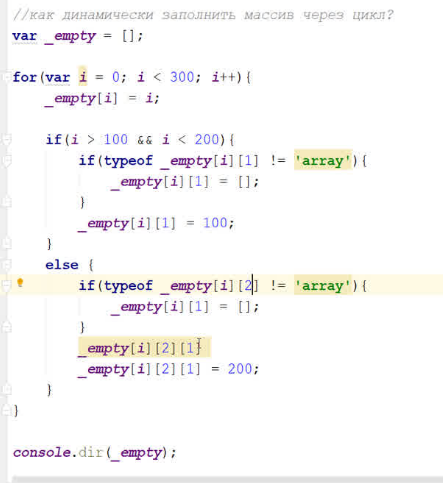
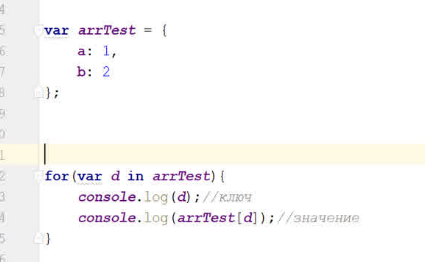
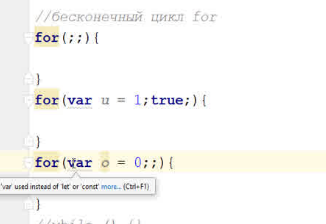
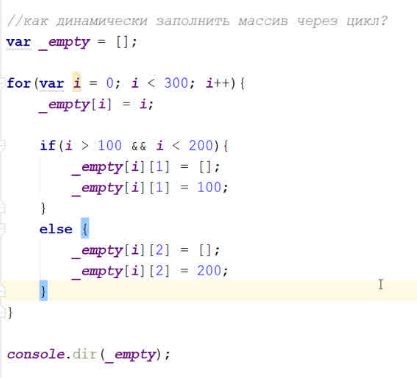










**Ссылки.**

Ссылки состоят из

url- указатель на ресурс

uri- указатель на параметр

/\*ССЫЛКИ.  
URL и URI  
URL:  
1) протокол:// - http://,https:// (правила передачи данных)  
2) домен - адресс иил название сайта  
 -поддомен - деректория сайта  
 -домен - уникальное имя сайта(строчка), ip  
 -доменная зона - географическое местоположение(.ru .com .рф и тд)  
3) папки и запрашиваемый ресурс  
 ресурс: - файлы : текстовые аудио видео .html .php . js .css и тд  
 путь/.../index.html  
  
URI:  
1) ? - разделитель между URL и URI  
2) параметры1=значения1&параметры2=значения2&параметры3=значения3  
 где параметр - название ключа(индекса)  
 где значения - передаваемые данные на сервере(/ ? : # - спецсимволы)  
  
Полный пример ссылки:  
http://api.vk.com/papka/index.php?a=1&b=2  
  
Виды ссылок:  
1) абсолютная  
 http://api.vk.com/papka/index.php?a=1&b=2  
 http://api.vk.com/papka/index.php  
 http://api.vk.com/papka/  
 http://api.vk.com/  
2) относительная  
 /papka/index.php?a=1&b=2  
 /papka/index.php/  
 /papka/  
 ../papka/index.php?a=1&b=2  
 ../papka/index.php/  
 ../papka/  
 papka/index.php?a=1&b=2  
 papka/index.php/  
 papka/  
  
 относитьельная ссылка создаёться на сервере и передаёться клинету, относительная ссылка  
 используеться , но она приобразуется в абсолютный адресс =>  
 => относительных ссылок нет  
\*/  
  
/\*  
Методы - способы передачи данных от клиента к серверу  
1) GET (обязательно знать) - передача данных ввиде текста  
 вариант: строка запроса, форма, ajax(async)  
 данные - отображаються в ссылке  
 небезапасно передавать персональные данные  
2) POST (обязательно знать) - передача даных в текстовом формате и ресурсов  
 варианты: форма, ajax(async)  
 данные - частично шифрует  
 защищает данные от просмотра  
3) DELETER  
4) PUT  
5) OPTIONS  
6) etc  
\*/  
  
  
/\*  
Таймеры.  
1) setinterval - установка счётчика времени с постоянным  
выполнением  
2) settimeout - установка времени через которое выполниться  
действие и остановиться  
  
параметры:  
1) функция выполнения по таймеру  
2) время - мс - 1000 мс = 1 с  
  
\*/