1. В html код JS должен быть межжду тегами: <script> </script>. Он может размещаться в <head> или в <body>. Скрипт размещенный в <head> будет выполнен до отображения <body>.
2. В JS можно использовать html теги для форматирования текста.
3. Скрипты могут быть размещены во внешний файлах. Можно размещать в <head> или в <body>

<head>

<script crs=”demo.js”></script>

</head>

1. //однострочный комментарий

/\* многострочный комментарий\*/

# ПЕРЕМЕННЫЕ

1. Переменные описываются с помощью let (или var – устаревшее)
2. Если нужно написать апостроф в строке используем обратный слэш \

let sayHello = ‘Hello! I\’m a boy’;

1. В булевом типе два значения: true или false
2. Деление по модулю возвращает остаток от деления:

25 % 5 = 0

26 % 6 = 2

1. Инкремент ++ увеличивает на 1

Декремент -- уменьшает на 1

|  |  |  |  |
| --- | --- | --- | --- |
| var++ | Постинкремент | var a=0, b=10;  var a=b++; | a=10  b=11 |
| ++var | Преинкремент | var a=0, b=10;  var a=++b; | a=11  b=11 |
| var-- | Постдекремент | var a=0, b=10;  var a=b--; | a=10  b=9 |
| --var | Предекремент | var a=0, b=10;  var a=--b; | a=9  b=9 |

1. Математические опериции имеют сокращенный вид:

x -= y += 9

|  |  |
| --- | --- |
| x = y | x = y |
| x += y | x = x + y |
| x -= y | x = x - y |
| x \*= y | x = x \* y |
| x /= y | x = x / y |
| x %= y | x = x % y |

7) Операторы сравнения возвращают true или false

|  |  |  |
| --- | --- | --- |
| == | равно | 5 == 10 ложь |
| === | идентично (равно и одинакового типа) | 5 === 10 ложь |
| != | не равно | 5 != 10 истина |
| !== | не идентично | 10 !== 10 ложь |
| > | больше чем | 5 > 10 ложь |
| >= | больше или равно |  |
| < |  |  |
| <= |  |  |

1. Логические (булевые) операторы:

AND (И) && возвращает true если обе операнды истинны

OR (ИЛИ) || возвращает true если одна из операнд истинна

NOT(НЕ) ! возвращает true если операнд ложный; возвращает false если

операнд истинный

(4>2)&&(10<15) – true

1. Условный (тернарный) оператор

присваевает значение при выполнении условия

variable = (condition) ? value1 : value2

let N = (age > 18) ? ‘Too young’ : ‘Too old’;

1. Конкатерация: построение строк, путем соединения других строк

let n = ‘Nadd’;

let m = ‘hedd’;

document.write(n + m);

# УСЛОВИЯ И ЦИКЛЫ

## if else

1. if (condition) {

statements

}

1. let n=7, m=8;

if n>m {

alert (‘it is ok’)

}

else{

alert (‘it is wrong’)

}

1. let c=1;

if (c == 1) {

alert (‘1’);

} else if (c == 2){

alert (‘2’);

} else {

alert (‘wrong’);

}

## switch

let day = 2;

switch (day) {

case 1:

document.write (“monday”);

break;

case 2:

document.write (“tuesday”);

break;

default:

document.write (“another day”);

}

## Цикл for:

for (i=1; i<=5; i++) {

do document.write (i + “<br/>”);

}

где i=1 – устанавливает переменную до запуска цикла (let i=1)

i<=5 – определяет условие запуска цикла

i++ -- увеличивает значение счетчика после каждого выполнения кода в цикле

Можно так:

for (i=1, text=” “; i<=5; i++){

text = I;

do document.write (i + “<br/>”);

}

## while

let i=0;

while (I <=10) {

do document.write (i + “<br/>”);

i++;

}

## do while

let I = 20;

do {

do document.write (i + “<br/>”);

i++;

}

while (i <= 25);

## break

выпрыгивает из цикла

for (i = 0; i <= 10; i++){

if (I == 5){

break;

}

do document.write (i + “<br/>”);

}

## continue

останавливает одну итерацию и продолжает со следующей

for (i = 0; i <= 10; i++){

if (I == 5){

continue;

}

do document.write (i + “<br/>”);

}

# ФУНКЦИИ

//чтобы выполнить:

function MyFunction (){

alert ('alert box');

}

//чтобы вызвать:

MyFunction ();

MyFunction ();

в скобках параметры функций. Их может быть несколько:

function sayHello (name){

alert ('Hi,' + name);

}

sayHello ('David');

sayHello ('Sarah');

sayHello ('Masha');

## return

возвращение значения функции

function MyFunc (a, b) {

return a\*b;

}

let x = MyFunc(5,6);

alert – окно предупреждения («Добро пожаловать»)

prompt – введите ваше имя

confirm – окно подтверждения («Вы действительно хотите покинуть страницу?»)

# ОБЬЕКТЫ

Объекты могут хранить много значений.

let person = {

name: 'John',

age: 32,

favColor: green,

height: 183

};

//Доступ к свойствам объекта:

let x = person.age;

let y = person['age'];

function person (name, age, color) {

this.name = name;

this.age = age;

this.favColor = color;

}

эта функция является конструктором обьекта, принимающим параметры и присваивающим их к свойствам обьекта