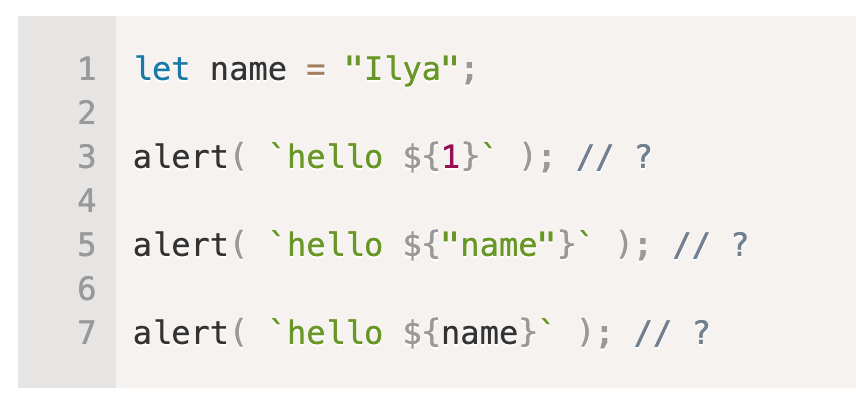
WEEK 13

1. Какие бывают варианты объявление переменных?

* let - объявление переменной с возможной последующей перезаписью
* const - объявление и (обязательно) присваивание переменной значения, в дальнейшем не изменяется
* var - устаревшее объявление переменной, может быть перезаписано (как и let), но в отличии от предыдущих имеет глобальную область видимости (в отличии от let/const - у них видимость только внутри блока кода)

1. Что выведет этот скрипт?



Сначала у нас выйдет уведомление “Hello, 1 - т.к. ранее мы не объявляли и не задавали значение переменной 1, оно отобразилось просто как число.

Во втором случае у нас выйдет Hello name, т.к. мы взяли переменную name в кавычки, что сделало её не переменной, а просто текстом.

Потом Hello Ilya, т.к. ранее мы объявили и задали переменную name.

1. Какое значение будет в z

var x = 5;

var y = 2;

var z = x + y;

alert(z);

Ответ: z = x + y = 5 + 2 = 7

1. Какими тремя способами можно подключить JavaScript код?

* Внутреннее подключение - прописав внутри тела html тег script (в head, но лучше в body), без использования атрибутов.
* Внешнее подключение - также с помощью тега script, но в этом варианте мы указываем путь к файлу .js
* Прописав внутри тега сам код, где должно происходить событие, по наступлению которого код будет выполняться. Устаревший вариант, лучше им не пользоваться, но пока учимся можно 😊

1. Есть ли какая-то разница между записями typeof str и typeof(str)?

Круглые скобки, которые можно не писать. Нам все равно возвращается строка с тем, что мы напишем после typeof.

1. Что вернет код?

function showX(x)

{

return x;

}

alert(showX(28));

Всплывающее окно с цифрой 28

1. Какой результат будет у выражений ниже?

"" + 1 + 0 10 - конкатенация (склейка)строк из-за бинарного оператора “+”

"" - 1 + 0 -1 - также конкатенация

true + false 1 - т.к true = 1, а false = 0, при сложении получается 1, правило численного преобразования

6 / "3" 2 - результат деления, в отличии от ‘+’ другие арифм. операторы работают как обычно

"2" \* "3" 6 - результат умножения

4 + 5 + "px" 9px - первый оператор + сложил два числа, а второй объединил результат со строкой

"$" + 4 + 5 $45 – в данном случае произошло объединение строк из-за первого строчного операнда

"4" - 2 2 - результат вычитания

"4px" - 2 NaN - некорректное значение, потому что мы пытаемся вычитать из не числовой строки

7 / 0 Infinity - результат деления на 0 всегда бесконечность

" -9 " + 5 -9 5 - конкатенация строк

" -9 " - 5 -14 - тут снова результат вычитания

null + 1 1 – null преобразуется в 0, поэтому 1

undefined + 1 NaN - некорретное значение, сложение не числовой строки с числовой

" \t \n" - 2 -2 - \t - табуляция, \n - перенос строки, получилась числовая строка

1. Ниже приведён код, который запрашивает у пользователя два числа и показывает их сумму. Он работает неправильно. Код в примере выводит 12 (для значения полей по умолчанию). В чём ошибка? Исправьте её. Результат должен быть 3.

let a = prompt("Первое число?", 1);

let b = prompt("Второе число?", 2);

alert(a + b); // возвращает 12

Ответ:

let a = prompt("Первое число?", 1);

let b = prompt("Второе число?", 2);

alert(Number(a) + Number(b)); // возвращает 3

1. Что будет в х?

x = 5; - объявили и задали переменную

++x; - прибавили к переменной 1

x -= 2; - выражение x = х - 2 = 6 - 2

alert(x); - вывели решение

Ответ: 4

1. Что выведет этот код?

function sum(x, y=0) {

return x+y;

}

alert(sum(5));

Ответ: 5