Неделя 14

1. Какие есть способы объявления функций?

- Function Declaration
- Function Expression
- Стрелочные функции (arrow functions)

2. Приведите примеры вызова одной и той же функции всеми известными вам способами.

```
function X(a, b) {
return a * b;}
alert (3*3);

let Y = function (a, b) {
return a * b;
};
alert (3*3);

let Z = (a, b) => a * b;
alert (3*3);
```

То есть вызов функции всегда будет один

3. В чем разница между тестированием и отладкой (дебаггингом)? А что такое логирование?

Тестирования – «испытание» кода, цель которого заключается в выявлении имеющихся в нем ошибок. Цель отладки состоит в выявлении и *устранении причин* ошибок. То есть это процесс не только поиска, но и выявления причин и исправления ошибок в скрипте. Можно сказать, что тестирование – это предшественник отладки.

Логирование – это процесс записи каких-либо событий, которые происходят в коде. То есть программа на Java-языке записывает сведения о своем исполнении в некий файл или базу данных. Логирование дает возможность отслеживать ход исполнения программы и конкретно кода. По логам проще разбирать баги и устранять их.

4. В чем разница между Function Expression и Function Declaration?

Они различаются только в самом способе объявления функций, результат выполнения и порядок вызова функций не меняется.

При способе Function Expression - мы сами объявляем переменную с именем функции и присваиваем ей описание функции. А при Function Declaration сначала создается функция, и её значение помещается в переменную с названием такой функции (интерпретатор объявляет такую переменную). Соответственно, при FD

код объявления функции записывается отдельной конструкцией, function (…) {}, FE функция создается внутри другого выражения (let … = function (…) {…};).

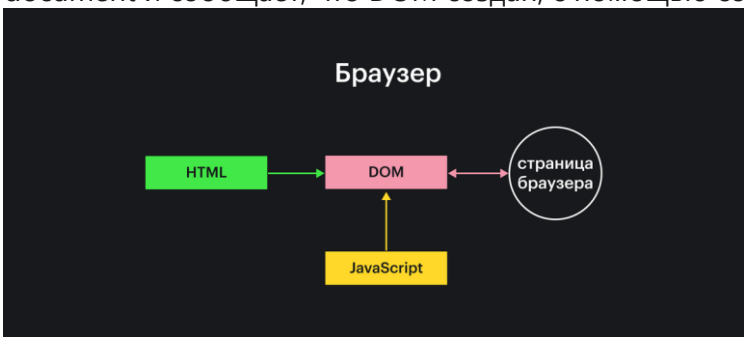
Кроме этого, есть различие во времени вызова функции JS. Функция, созданная как FD создается как бы заранее, когда скрипт готовился к выполнению, соответственно, они могут быть вызваны до их объявления. А вот функция, созданная как FE создаётся, когда выполнение доходит того места в коде, где она была прописана, то есть заранее вызвать функцию нельзя.

5. Что делает функция console.log()?

Выводит информацию в веб-консоль, в виде дерева, если это объект, которое удобно раскрывается и видны все параметры. То есть при написании скриптов можно увидеть какой-либо промежуточный результат прямо в консоли браузера, содержимое DOM.

6. Что такое BOM и DOM?

DOM (Document Object Model) - это специальная древовидная структура, которая позволяет управлять HTML-разметкой из JavaScript-кода. Фактически это все содержимое страницы в виде объектов (где каждый HTML-тег на странице является объектом). Основной объект - document.
Соответственно, весь HTML-документ можно разложить в виде дерева тегов, где теги - это элементы, которые образуют всю структуру дерева. От корневого узла-элемента (тег html), ко всем дочерним (head, title). Элемент на самом нижнем уровне - текстовый узел, это тег только с текстом, у которого нет потомков.
Браузер создаёт DOM при загрузке страницы, складывает его в переменную document и сообщает, что DOM создан, с помощью события DOMContentLoaded.



BOM (Browser Object Model) – объектная модель браузера - окружение, среда, в которой запускается браузер. Состоит из дополнительных объектов, предоставляемых браузером, чтобы работать со всем, кроме документа. Она (BOM) предоставляет доступ к navigator, location, fetch и другим объектам.

7. Есть вот такая страница: … Как найти в ней?...

1. Таблицу с `id="age-table"`

```
let age-table = document.getElementById('age-table');
console.log(age-table);
```

2. Все элементы `label` внутри этой таблицы (их три)

```
let labels = age-table.getElementsByTagName('label');
console.log(labels);
```

3. Форму `form` с именем `name="search"`

```
let form = document.getElementsByName(search);
console.log(form);
```

8. Как сделать переход на другую страницу при клике на кнопку (без <a href=...>, только средствами JavaScript)?

С помощью объекта BOM: location. Например, есть кнопка #mybotton:

```
document.getElementById('mybutton').onclick = function() {

window.location.href = 'redirect-url';

//redirect-url - адрес страницы, которая будет открываться по клику на кнопку

};
```

9. Как можно обнулить (очистить) значение внутри input?

Например, есть кнопка с id= clearButton, при нажатии на которую произойдет очистка поля ввода input с id= textInput:
<input id="textInput" type="text" placeholder="Input any string value" />
<button id="clearButton">Clear</button>
Для очистки нужно найти указанные элементы и прописать функцию, которая будет запускаться при клике на кнопку:

```
document.getElementById("clearButton").onclick = function(e) {
document.getElementById("textInput").value = "";
}
```

10. Как будет выглядеть ваша функция приветствия из прошлого домашнего задания, если ее переписать в стрелочном формате?

В прошлом задании функция была записана как Function Declaration:

```
function showHello() {
let Name = prompt('Как тебя зовут?', []);
alert(`Привет, ${Name}!`);
}
```

В стрелочном формате будет выглядеть так:

```
let showHello = () => {
let Name = prompt('Как тебя зовут?', []);
alert(`Привет, ${Name}!`);
}
```